コンピュータセキュリティ　14−27
（2001. 7. 25）

PKI環境における異なるドメイン間での証明書失効状態通知方法の提案と設計

三宅 優[†]　Jonathan Millen[‡]　Grit Denker[‡]
田中 俊昭[†]　中尾 康二[†]

[†](株)KDDI研究所
[‡]SRI International, Inc.
E-mail: miyake@kddilabs.jp

概要

PKI環境で使用される証明書が他のドメインで使用される場合には、その証明書の失効状態に関する情報をそのドメインに配布する必要がある。他のドメインへの情報の配布は、セキュリティ上の観点から必要最小限にすべきであり、インターネットを介して相互接続する場合には、第三者からのアクセスに対しても注意しなければならない。本稿では、異なるドメインへ証明書の失効情報を安全に配布する方法について提案するとともに、現在広く利用されているWWW環境に提案方式を適用するための導入方法について述べる。

キーワード：PKI、アクセス制御、証明書失効リスト、OCSP、WWW

Notification of Certificate Revocation Status between Different Domains under PKI System

Yutaka Miyake[†]　Jonathan Millen[‡]　Grit Denker[‡]
Toshiaki Tanaka[†]　Kouji Nakao[†]

[†]KDDI R&D Laboratories, Inc.
[‡]SRI International, Inc.
E-mail: miyake@kddilabs.jp

Abstract

When public key certificates are used to control access by a client in one domain to a server in another domain, the certificate revocation status should be distributed to the server domain also. For security reasons, the distribution of information to other domains should be minimized, and external distribution points are subject to attack from third parties on the Internet. In this paper, we consider a mechanism to convey the current revocation status of certificates to other domains securely under PKI (Public Key Infrastructure) system. We also discuss a method to adapt our proposed mechanism to the WWW environment using standard browsers.

Keywords: PKI, Access Control, Certificate Revocation List, OCSP, WWW

1 Introduction

The combination of WWW (World Wide Web) browsers and servers makes it convenient for anyone to access information on servers. Some servers provide private or sensitive information and require users to log in with password or smart cards. This form of access control is suitable if users must be distinguished from one another.

Recently, many organizations have been introducing PKI (Public Key Infrastructure) and distribute public key certificates for each member. The certificate includes user information, a public key for the user, and a signature by the CA (Certificate Authority). If the CA is trustworthy and it is proved that a user has a private key that corresponds to the public key in the certificate, the user can be identified and authenticated.

It is possible to use the PKI system for access control. In case of WWW accessing, Web servers can identify the accessing clients by checking their certificates and possession of private keys. This method has several advantages compared with systems passwords or smart cards. The main difference between the PKI and password base is the procedure after the password or private key is compromised. In case of password-based access control, the client should change the password for every Web server. If the client accesses many Web servers with the same password, this procedure will be required for each Web server. Though the clients may assign different passwords for each Web server, this method necessitates troublesome password management. In the case of PKI, when the private key is revealed, the client revokes the certificate corresponding to the private key, and asks the CA to issue a new certificate. It is not necessary to inform every Web server that the private key and certificate are altered, because the Web servers obtain this revoked status information when they validate the certificate of accessing clients. Even if someone attempted to use the old certificate, when the revocation status of the certificate is checked, it is proved that the certificate is invalid.

In order to incorporate the PKI system into access control, the server has to have the capability to validate the certificates of clients. Normally, the server checks the revocation status of a certificate by retrieving a CRL (Certificate Revocation List) [1, 2] or accessing an OCSP (Online Certificate Status Protocol) [3] responder. If the client and server belong same domain and their certificates are issued by a CA in this domain, it is easy for the server to check the status of client's certificate. However if the client and server belong to different domains and their CAs are managed independently, a mechanism to inform the server in the other domain of the status of the certificate should be provided.

In this paper, we discuss the notification method for conveying the status of certificates between different domains. In this situation, both domains rely on their own root CA and there may be some restrictions to access resource servers on the other domain. Therefore we propose an appropriate procedure to exchange the status of certificates safely. Moreover, we also describe an implementation design of our approach for the WWW environment.

2 Notification of revoked certificate status between different domains

2.1 Connect of two independent PKI domains by cross certificates

In order to construct a certificate chain from a client certificate to a server CA, both the CA that issues the client certificate and the server CA

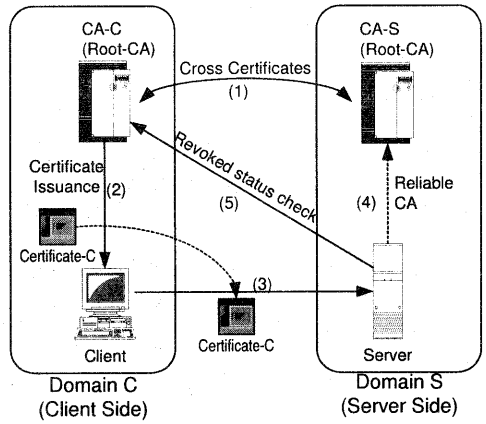

Figure 1: Certificate validation using cross certificates

would normally have the same root CA at the top of the CA hierarchy. Generally, the certificate chain is made via the root CA. However, many domains manage their own root-CA independently. In this case, another method is needed to establish a certificate chain between different domains.

Cross certificates [4] have been introduced to resolve this problem. Figure 1 shows the mechanism of cross certificates. In this figure, there are two domains, Domain C for client side and Domain S for server side. Domain C and Domain S have their own root CA, CA-C and CA-S respectively. In the cross certificates, CA-C and CA-S exchange their public key certificates and sign the received certificates. This means that the server that relies on CA-S can accept certificates signed by CA-C, because there is a certificate for CA-C signed by CA-S. In this situation, the server in Domain S can validate the certificate of client (Certificate-C) with the following procedure.

(1) The root-CAs in Domain C (CA-C) and Domain S (CA-S) exchange their public key certificates. The received certificates are signed by each root-CA.

(2) CA-C in Domain C issues a public key certificate for a client (Certificate-C). This certificate is signed by CA-C.

(3) When the client accesses a server in Domain S, it sends its public key certificate (Certificate-C) to the server.

(4) The server checks the validity period and signature in the certificate. If the signature of this certificate is signed by the CA-C, the server needs to retrieve the certificate for CA-C. Therefore the server retrieves the cross certificate for CA-C, and checks its signature. Because the cross certificate for CA-C is signed by the reliable CA (CA-S), the server can rely on it. If the certificate (Certificate-C) is signed by a public key in the certificate for CA-C, it is deemed conditionally valid.

2.2 Check of revocation status for certificates between different domains

Using cross certificates, the sever can validate certificates issued by CAs in other domains. After validating the signature in certificate, the server should check the revocation status of the client certificate ((5) in figure 1). If the client certificate is revoked before it expires, the certificate becomes unavailable.

In order to get the revocation information for certificates, the server should retrieve a CRL [1,2] that is a list of revoked certificates, or should access an online verification server, such as an OCSP responder [3]. In case of the CRL, the server requires the CRL that was issued by the same CA that issued the client certificate. The online verification server also needs the information from the CA that issued the client certificate. Therefore the server has to access another domain to retrieve the revocation information for the client certificate if the CA in another domain issued it.

When the server retrieves the CRL from another domain or accesses the online verification server in other domain, there are some problems.

- Each domain must prepare an access point that provides certificate status information outside of the firewall for other domains. Because the access point is external, maximum security protection is required for its data.
- Access to the access point should be restricted to known, registered domains. The access point should not send information about its domain to unrelated domains.
- The access point should provide only the information that is required to access other domains. Outflow of unnecessary information may reveal the organizational structure in its domain, and it may become a security hole.

These problems also increase the management cost for each domain, and increase the risk of intrusion from other domains by placing the access point outside of the firewall.

2.3 Push mechanism for notification of revoked status

Because it is not safe to make an external access point for other domains, we use a push mechanism for notification of revoked status. This means that the client domain sends the status information directly to the other domain that needs this information directly.

As a notification method of the revoked certificate status, we considered and rejected the approach in which each domain distributes CRLs to other registered domains periodically. This method has the following disadvantages.

Lack of scalability: The CRL information that should be sent to other domains increases along with the number of registered domains. If there are many registered domains that need CRLs, this method requires a high speed network.

Useless information: The CRL lists the serial number of all revoked certificates. However most of this information may not be used at the other domains. It may mean that useless information is sent to many domains periodically.

Information leak: Since the CRL includes every serial number of revoked certificates, a change in this information may suggest management policy of the PKI, organizational restructuring, etc., in that domain. Therefore each domain should keep to a minimum the information sent to other domains.

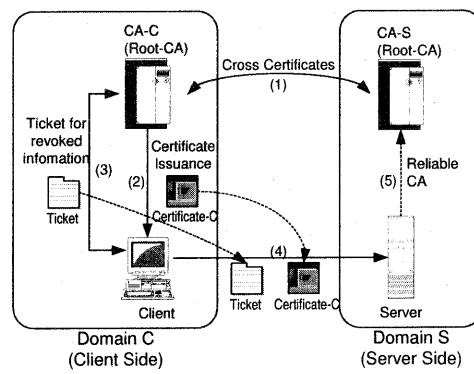

Figure 2: Transfer of revoked status information with ticket

In order to resolve these problems, we propose the mechanism shown in Figure 2. In this mechanism, the client sends a ticket that has the status of the client's certificate. The procedure for this mechanism is:

(1) Root CAs in Domain C (CA-C) and Domain S (CA-S) exchange their public key certificates. The received certificates are signed by each root CA.

(2) CA-C in Domain C issues a public key certificate for a client (Certificate-C). This certificate is signed by CA-C.

(3) Just before accessing a server in Domain S, the client requests a ticket that certifies the non-revoked status of the client's certificate by the CA-C. The validity period of this ticket is short.

(4) When the client accesses the server in Domain S, it sends both its public key certificate (Certificate-C) and the ticket to the server.

(5) The server checks the conditional validity of the certificate as before. It also checks the signature of the received ticket with same process. If the ticket is not expired and it indicates that the certificate has not been revoked, the certificate is accepted as valid.

Only the minimum amount of information is sent to the other domain in this mechanism. Since the ticket that includes the status information is sent with the certificate, the client can also know the time and domain where the status of the client certificate is sent.

3 Implementation to WWW Environment

In order to apply our proposed approach to a practical WWW environment, some modification and additional elements are required. In this section, we describe a system architecture that incorporates our approach in the WWW environment with PKI.

3.1 Assumptions

Currently, browsers and servers for WWW are used widely. Therefore, it is desirable that the current system (browsers, Web servers, CA for PKI) can be used. Accordingly, we designed the system architecture in consideration of the following points.

- There are several kinds of browsers. We would not like to modify each browser one by one. Moreover, we would not like to distribute special software to each client for this system. Therefore, we do not change the client browser.
- It is not easy to add new functions to a CA. Therefore, we decide to make a separate software module called a PGS (Privilege Granting Server). The function of this software is to issue the ticket that indicates the status information of a certificate. Since the ticket is signed with the private key of CA, we assume that the PGS is executed on same computer on which the CA is running.
- SSL (Secure Socket Layer) [5, 6] has the capability that the server can authorize the client by using the client's certificate (SSL Handshake Protocol). Because many browsers and Web servers support SSL, we decided to use this function for our system.

3.2 System Architecture

Figure 3 shows the system architecture that incorporates our method. We adopted a Web proxy design for ticket handling. When the client browser uses the Web proxy to access the Web server, the Web proxy obtains a ticket and passes it to the server. The procedure for this architecture is as follows.

(1) Root CAs in Domain C (CA-C) and Domain S (CA-S) exchange their public key certificates. The received certificates are signed by each root-CA.

(2) CA-C in Domain C issues a public key certificate for a client (Certificate-C). This certificate is signed by CA-C.

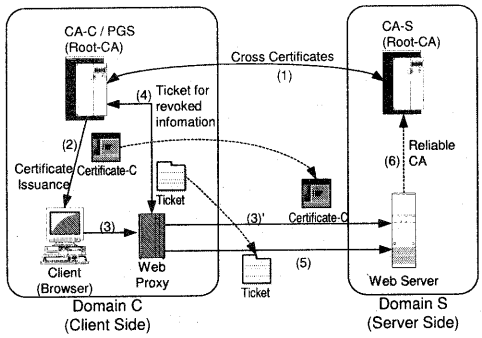

Figure 3: System architecture

(3) The client browser tries to access the Web server in Domain S via the Web proxy in Domain C. The Web proxy relays communication between the browser and Web server.

(4) During SSL handshake, the Web proxy extracts the certificate information sent by the client. The Web proxy requests the status of the certificate from the CA-C / PGS, and receives the ticket.

(5) The Web proxy sends the ticket to the Web server in Domain S. This action is independent of communication between the browser and Web server.

(6) The server checks the validity periods and signatures in the certificate and ticket as before.

A new Web proxy and some modification of the Web server is required in this architecture. However, there is no modification of the client browser because the Web proxy executes the added procedure instead of the client. The Web proxy might run on the client workstation, or it might run on a gateway so that it can be shared by all the clients on a local network.

3.3 Protocol for authorization and ticket

Figure 4 shows the protocol for client authorization. (The steps that do not relate to client authorization are omitted in this figure.) SSL handshake protocol messages are shown in italics.

(1) In the SSL handshake protocol, the Web server requests the public key certificate of the client by using SSL *CertificateRequest* message.

(2) The client returns the certificate by using the SSL *ClientCertificate* message.

(3) The Web proxy in Domain C extracts the certificate from the message *ClientCertificate*. It asks the PGS to issue the ticket for this certificate.

(4) After the Web proxy receives the ticket from the PGS, it forwards the ticket to the Web server in Domain S. The authorization module in the Web server receives this ticket. This communication is independent from the SSL handshake protocol. The authorization module is a new function added to the web server to validate and read the ticket.

(5) Both the browser and Web server send the *ChangeCipherSpec* message. After this message, all messages are encrypted.

(6) Both the browser and Web server send the *Finished* message. This message completes the SSL handshake protocol.

(7) When the Web server receives the *Finished* message from the browser, it checks the revoked status of certificate by contacting the authorization module. This check takes the place of other activities like password checking that are performed in other servers to control client access.

(8) The SSL module in the Web server receives the result (Accept or Deny) of revocation status from the authorization module.

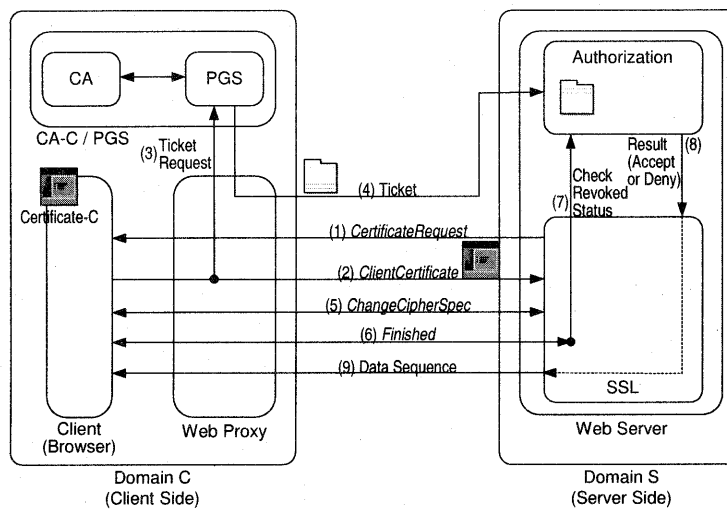

Figure 4: Protocol for client authorization

(9) If the result is Accept, data on accessing pages are transferred to the browser. In case that result is Deny, messages that indicate access denial are sent to the client.

The Web server does not alter the communication sequence between the server and client. It observes the SSL message passively to obtain the client certificate. It does not know any private key or decrypt any encrypted messages. Therefore the secret or authenticated communication between client and server is protected from the Web proxy after the SSL handshake is established.

The ticket uses the OCSP Response Message format [3]. Minimally the following information has to be included in this ticket.

- CA Name
- Serial Number of Client Certificate
- Revocation Status (Revoked, Not Revoked, Unknown)
- Validity Period of This Ticket
- Signature by CA

4 Conclusion

We have proposed a PKI mechanism to support access control of a client in one domain to a server in another. When the server and client belong to different PKI domains, the server requires retrieving a CRL or accessing to an online verification server, such as an OCSP responder, to check a revocation status of the client certificate in normal PKI system. However, from the viewpoint on security, it is not desirable for the client domain to provide an access point outside of an Internet firewall and to distribute unnecessary information. Therefore we have proposed the mechanism to attach a ticket that includes the revoked status of certificate with the sending client certificate. With this mechanism, only the minimum status information is sent, it is sent unforgeably, and only when access is requested. The external access point to distribute status information to other domains is not required.

In order to implement this architecture easily into the WWW environment, we designed a system architecture that does not require modification to Web browsers. It uses a Web proxy that ob-

serves an SSL connection passively, a PGS module to support ticket generation in conjunction with a client CA, and an authorization module in the Web server to check access using the certificate and ticket.

Access control and authorization at the server are based on the contents of the client certificate, using some form of ACL (Access Control List). While access could be controlled on the basis of the user or subject name, the client CA can also create client certificates containing a more general privilege field. Access control by privilege is more efficient when the server has many individual clients, and is willing to trust the client CA to assign server privileges to clients in its domain. This approach can be supported without change to the architecture as described.

References

[1] "ITU-T Recommendation X.509: Information Technology – Open Systems Interconnection– The Directory: Public-Key and Attribute Certificate Frameworks," ITU-T, March 2000.

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

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

[4] C. Adams, S. Farrell, "Internet X.509 Public Key Infrastructure Certificate Management Protocols," RFC-2510, March 1999.

[5] A. Freier, P. Karlton, P. Kocher, "The SSL Protocol: Version 3.0," Netscape Communications Corp., March, 1996.

[6] T. Dierks, C. Allen, "The TLS Protocol: Version 1.0," RFC-2246, January 1999.

[7] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC-1510, September 1993.