通信情報保護學會論文誌
第 10 券 第 2 號, 2000. 6

A Method on Maintaining Consistency of Certificates in Public Key Infrastructure using DNS

Woo-jin Seok*, Man-hee Lee*, Ok-hwan Byeon*

DNS를 사용한 공개키 인증서의 일치성 보장에 관한 연구

석 우 진*, 이 만 희*, 변 옥 환*

ABSTRACT

In this paper, we propose a new algorithm which resolves the inconsistency problems occurring when DNS servers are employed as elements of PKI. The inconsistency may take place between primary name servers and secondary name servers, and between cached certificate and original certificate. The former can be removed by adapting RFC 1996 NOTIFY opcode for DNS. In order to eliminate the latter type of inconsistency, we develope a new algorithm which is implemented with two additional RR(Resource Record). The present algorithm is designed such that DNS contacts the destination DNS prior to returning public key to users. Therefore, the inconsistency problem occurring when DNS is operated as PKI can be eliminated by using the proposed adaptation and algorithm.

keyword : *PKI(Public Key Infrastructure), certificate, DNS(Domain Name System)*

I. Introduction

With increasing demand on commercial use of Internet necessarily, security has become an essential element in Internet. Considerable efforts have been given to designing a more secure mechanism. Most of them adopt a sequence of encryption and decryption of messages to prevent eavesdropping and unauthorized modification during message transfer.

There are two types of cryptographic systems: symmetric key cryptographic system and asymmetric key cryptographic system.[1] In symmetric key cryptographic systems, the same key is used in the encryption and decryption of messages and its examples include DES and IDEA. Both parties participating in a communication connection should keep this key secret.

In contrast, asymmetric key cryptographic systems use complementary pairs of keys to separate the functions of encryption and decryption. One key, the private key, is kept secret like a key in symmetric key cryptographic systems. The other key, the public key, does not need to be kept secret. This two-key approach can simplify key manage-

* 연구개발정보센터 슈퍼컴퓨팅사업단 슈퍼컴퓨팅인프라개발실

ment, by minimizing the number of keys that need to be managed and stored in a network, and can enable keys to be distributed via unprotected systems such as public directory services. Examples of asymmetric key cryptographic systems include RSA.[2]

The use of symmetric cryptographic systems has some troubles with distributing the keys for every public user since the key management is too difficult to be implemented, that is, there should be so many pair keys for every public user. It makes the asymmetric cryptographic systems, such as RSA, to be focused nowadays. For public key distribution in asymmetric cryptographic system, the PKI (public key infrastructure) should be established for distributing public key. In the PKI, a user registers his public key to a certification authority, and the certification authority signs the user's public key with his private key and publishes it as the certificate of the user.[3]

Recently some proposals have been made to extend the DNS (Domain Name System) so that users can retrieve public keys and/or certificates from the DNS securely.[4,5,6] It is efficient to adopt DNS for infrastructure of public key distribution because DNS is already established in the Internet and gives users naming service well.[7,8] However, there is a critical drawback. The distributed certificate containing the public key with signature of certificate authority may be inconsistent with the original certificate in DNS. This can occur due to an unexpected revocation within expiration time.

In this paper, we propose a method which can solve this inconsistency problem mentioned above, also can support PKI established by using DNSs to be free of inconsistency of certificates. The organization of this paper is as follows. Section 2 introduces the DNS operation and the role in PKI, respectively. Section 3 explains our proposed method to remove the inconsistency of certificates and followed by conclusion in section 4.

II. DNS in PKI

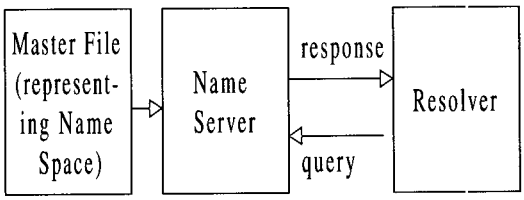
2.1 Basic Operation of DNS

DNS converts domain name into IP address on the basis of TCP/IP protocol. DNS was revised to improve the inefficient mechanism established by the old name conversion system based on host.txt file. DNS consists of three components, which are domain name space, resolver, and name server, asshown in Figure 1.[7,8]

Name server returns IP address with RR (Resource Record, unit of naming information) in response to the query of users through resolver on the basis of hierarchical domain name space. And name servers share their naming information with other name servers for backup by their information exchange, which is called zone transfer. Zone transfer should be processed at every negotiated interval because the naming information can be updated in authoritative name server, which is called primary name server. Resolver transfers the query of users to name server, and the response of name server to users. Recently a proposal has been made to extend the DNS for users to get response from name server securely.[4] In that proposal, the information stored in the DNS server and query/response is protected from unauthorized modification and so on.

2.2 PKI and X.509

Key management in PKI deals with the secure generation, distribution, and storage of keys. Users should be able to securely obtain



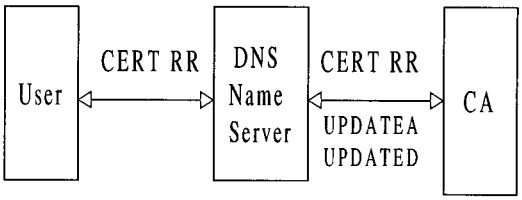
(Figure 1) Three components of DNS

a key pair suited to their efficiency and security needs, and should be able to legitimately obtain other users' public keys; otherwise, an intruder can either change public keys listed in a directory, or impersonate another user. To prevent this kind of intrusion, certificate is used. Certificate is digital documents attesting to the binding of a public key to an individual or other entity. It allows verification of the claim that a specific public key belongs to a specific individual. In its simplest form, certificates contain a public key and a name. As commonly used, a certificate also contains a expiration time, the name of the certifying authority that issued the certificate, a serial number, and perhaps other information. Most importantly, it contains the digital signature of the certificate issuer. The definition of Certificate is well expressed in X.509.[3]

A CRL is a list of certificates that have been revoked before their scheduled expiration time. There are several reasons why a certificate might need to be revoked and placed on a CRL. For instance, the key specified in the certificate might have been compromised or the user specified in the certificate may no longer have authority to use the key. When verifying a signature, one examines the relevant CRL to make sure the signer's certificate has not been revoked. A CRL is maintained by a CA, and it provides information about revoked certificates that were issued by that CA. CRL only contains current certificates, because revoked certificates should not be accepted in any case: when expiration time of a revoked certificate expires, that certificate can be removed from the CRL.[3]

2.3 Storing certificates in DNS

DNS has been developed and modified for infrastructure of naming service in Internet, thus applying DNS to PKI can be done efficiently and easily. The CERT RR (a type of RR for certificate) and the CRL RR (a type of RR for certificate revocation list) are proposed to apply DNSs as components in PKI.[5,6] As shown in Figure 2, these two kinds of RRs can be stored in DNS server after CA (certificate authority) signs them. The interfaces between DNS server and CA can be implemented by dynamic update operation proposed in RFC 2136.[6,9]



(Figure 2) Interfaces of DNS servers

The new CERT RR can be stored in DNS server from CA by using UPDATEA parameter in dynamic update operation and the CERT RR can be deleted by using UPDATED parameter[6,9]. After DNS is established as certificates repository, it distributes certificates to other DNSs which want to get its CERT RR. The secondary name server wants to get RRs including CERT RR by zone transfer.

And a DNS that is queried a CERT RR by users but has not the CERT RR wants to get the CERT RR. In this case the DNS forwards the query to parent-DNS logically associated in domain name space by DNS protocol. Finally the DNS can get the requested CERT RR. Then, it caches the CERT RR, and answers to the requesting user.[7,8] In this DNS mechanism, inconsistency problem can occur when the original certificate is revoked without updating the CERT RRs which are zone-transfered to secondary name server and cached in other DNSs. Followings are the two cases for these problems.

1) Inconsistency caused by zone transfer from primary name server to secondary name server.
2) Inconsistency between distributed certificates

and original certificates by caching operation of DNS.

Zone transfer and caching are the core functions in DNS, so the inconsistency problem in DNS based PKI is inevitable. We will propose a method to overcome this inconsistency problem in next section.

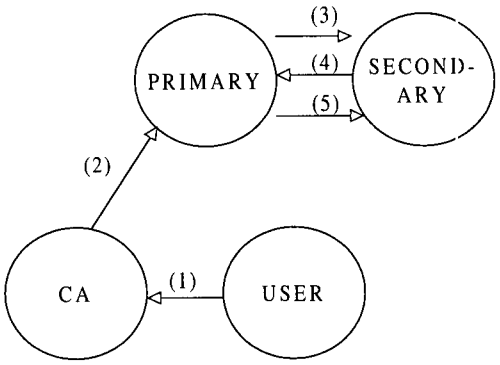
Ⅲ. A Proposed Method on Maintaining Consistency of Certificates

In this section, we describe the solutions for inconsistency problems as mentioned in the previous section. The following subsection suggests a solution for inconsistency between primary name server and secondary name server, and the next subsection proposes a solution for inconsistency caused by caching.

3.1 Consistency between primary and secondary name servers

In addition to a name server which has a naming information, another name server needs to be set up for robustness.[10] If only one name server is set up and it goes down, no one can look up names. A second name server splits the load with the primary name server or handles the whole load if the first server is down. The crucial difference of the two name servers is where the servers get their data. A primary name server reads its data from files, but a secondary name server loads its data over the network from primary name server. This process is called a zone transfer, and the zone transfer is activated by secondary name server when the refresh time is expired.

The transfer of certificate CERT RR and certificate revocation list CERT RR from primary name server to secondary name server can cause the inconsistency problem because the secondary name server cannot recognize the update of the two kinds of RRs stored in primary name server until next zone transfer. With NOTIFY operation as proposed in RFC 1996, we can solve the inconsistency problem between primary name server and secondary name server. When a primary name server updates one or more RRs in which secondary name servers may be interested, the primary name server sends the changed RR's name, class, type, and so on, to each known secondary name server using the best efforts protocol based on the NOTIFY opcode.[11] NOTIFY operation uses the DNS message format, although it uses only a subset of the available fields. The response message from the secondary name server contains no useful information, but its reception by the primary name server is an indication that the secondary name server has received the NOTIFY DNS message.



(Figure 3) Zone Transfer using NOTIFY opcode

Upon completion of a NOTIFY transaction for SOA (start of authority) query type, the secondary name server should behave just like the zone had reached its refresh interval. And then, a zone transfer should be initiated. The applying NOTIFY operation to the DNS to remove the inconsistency between primary name server and secondary name server is shown Figure 3.

1) When a certificate of a user is revoked by some reasons, it should be recognized by CA, and then CA will update the certificate revocation list CERT RR in primary name server.

2) CA appends the revoked certificate to certificate revocation list CERT RR, and updates the certificate revocation list CERT RR in primary name server by using UPDATEA and UPDATED. At this time, the inconsistency of certificate revocation list CERT RR can occur between primary name server and secondary name server, since the CERT RR information in the secondary name server does not match with that of primary name server. Next three steps, zone transfer by NOTIFY opcode, can solve inconsistency of certificate revocation list CERT RR.
3) Primary name server transfers the DNS message containing NOTIFY opcode, and makes the refresh time in secondary name server expire.
4) The secondary name server whose refresh time is expired checks the serial number of primary name server, and it demands zone transfer if the serial number of primary name server is larger than that of secondary one.
5) The primary name server transfers its zone file including certificate CERT RR and certificate revocation list CERT RR to the secondary name server through TCP connection. Therefore, inconsistency of CERT RR will not occur.

As mentioned above paragraphs, we can solve the inconsistency problem between primary and secondary name server by NOTIFY opcode when a CERT RR in the primary name server is revoked. Therefore, the DNS can distribute certificates in PKI without inconsistency between primary and secondary name server.

3.2 Consistency between cached certificates of user and original certificates of name server

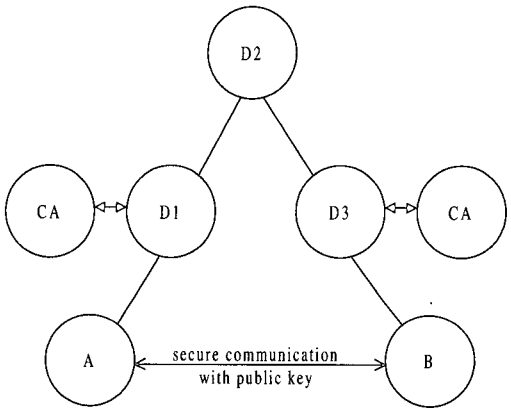
In this subsection, we show the case of inconsistency between cached certificates of user and original certificates of name server, and propose a solution for the inconsistency.

3.2.1 Inconsistency caused by caching

As shown in following Figure 4, when user A want to communicate with user B securely, user A should acquire user B's public key in user B's certificate. Then user A can communicate with user B securely by using B's public key in acquired certificate. When DNS is employed for distributing the certificate in PKI, the certificate is represented as CERT RR in database of DNS. DNS contains various types of RR, and it should support their own role. And it should also cache the RR data for next query, and the cached RR remains in cache for explicit TTL(Time-To-Live) time. During this time, DNS returns the answer by extracting the data from cache if it exists.

If a TTL time is not explicitly referred, the default TTL time is set to the expiration time in SOA RR. However, because the certificate CERT RR and certificate revocation list CERT RR have expiration time in itself, the TTL time for CERT RR will be needless any more. Therefore, the TTL time for CERT RRs should be set to the expiration time in them. However, if the certificate CERT RR is revoked in TTL time set by expiration time, the certificate CERT RR already distributed is not matched with the original one.

To avoid this, we can consider two method expressed in Exp.1 and Exp.2. Firstly, we may not cache CERT RRs. That is, we may put zero



(Figure 4) Consistency between DNS server and users

value into TTL time of CERT RRs. If it does, DNS should query and distribute a CERT RR every time when requested. This method will require much time for querying and description of CERT RR signature by using RSA or other algorithm. The time required in this method is shown in following Exp. 1.

$$(2+\mathrm{LogU})\mathrm{Qtime}+\mathrm{RSAtime} \quad \text{(Exp 1)}$$

Where Qtime is DNS query processing time and RSAtime is RSA processing time, and U is the number of nodes in domain name space.

A required query is transferred from a DNS to its parent DNS first, then to root DNS, and finally to the destination DNS. If we let the whole number of nodes in the naming space of DNS be U, the length from root to destination DNS will be LogU. Totally, 2+LogU query transfer will be required to get to destination DNS. Thus the total query processing time will be 2+LogU multiplied by each query prcessing time. When a CERT RR is returned as an answer to the query, DNS should check the signature in CERT RR. Checking the signature takes as much time as what RSA or other algorithm requires. Therefore, when DNS does not cache CERT RR, it should take (2+LogU) Qtime+RSAtime.

Secondly, it may use certificate revocation list CERT RR in DNS. When certificate revocation list CERT RR is changed, the DNS distributes the updated information in certificate revocation list CERT RR to other client DNSs, and makes them update the cached information of CERT RR. This method has some difficulties in being implemented because the DNS should manage client DNS information, and the information will be too big to be contained in a management table. The size of management table will be calculated as following Exp. 2.

$$\mathrm{Ncert}\times\mathrm{Ndns}\times\mathrm{Ninfo} \quad \text{(Exp. 2)}$$

Where Ncert is the number of revoked CERT RR, Ndns is the number of related DNS and Ninfo is the number of information of each client DNS, such as IP address and so on

The number of client DNS which has been related with a CERT RR can be too large, therefore, the size of table required in Exp. 2 can give troubles in implementing the DNS. In next subsection, we propose a new algorithm which can reduce the required time in querying and the size of table to be implemented practically.

3.2.2 Proposed algorithm

Two methods mentioned above are not practical because of much required time for query and much large size of table for notification of CRL update. In this subsection, we propose a practical algorithm which does not require much time and big size of table.

In the first query, that is, a user queries a CERT RR which has not been queried before, DNS sends a query to the destination DNS via the root DNS to get the CERT RR. And after getting the CERT RR from the destination DNS, the requesting DNS verifies the signature and returns the public key to the requester.

However, in the second and later query, the DNS returns the public key from the cached CERT RR. At this time, the DNS check if the CERT RR is still valid or not by consulting the original DNS of the CERT RR before returning the public key of CERT RR. If the CERT RR is valid, the DNS returns public key to requester, otherwise, the DNS discards the CERT RR from cache because it is not useful any more.

In this paper, we define two new types of RR, NSA type and CHK type, for our proposed algorithm. The NSA(Name Server Address) type represents the IP address of its authoritative DNS. Thus, we assume that the destination DNS should send its IP address with CERT

RR in response to the query. The CHK(CHecK) type requires the function of checking if the requested CERT RR is included in certificate revocation list CERT RR or not. The flow of the proposed algorithm is shown well in Flowchart 1 on the basis of the proposed two types.

```
if (the type of query is CERT && the query is
not authoritative here) {
    Address=Find(Destination domain name,
    NSA type)
    /* FInd returns the address of Destination
       domain name server from local zone file
       */ Response=Query(Address, Destination
       do- main name, CHK type)
    /* Query returns Yes if the Destination
       domain name is recorded in the CRL
       in Destination domain name server
       having Address */
    if(Response is Yes)
            Discard the CERT RR
    else
            Return the public key of CERT in
            cache
}
```

Flowchart 1. The proposed algorithm

In the proposed algorithm, the DNS should extract IP address of the DNS which is authoritative for the requested CERT RR when a user queries public key with CERT type. And it checks if the CERT RR is included in certificate revocation list CERT RR or not by sending query the destination DNS whose IP address is extracted from NSA type. The destination DNS returns the Response by checking the domain name in certificate revocation list CERT RR. If the domain name exists in certificate revocation list CERT RR, it means that the CERT RR has been revoked by some reasons, that is, the CERT RR is not valid any more. If the domain name does not exist in certificate revocation list CERT RR, it means that the CERT RR is valid. The time required for processing the CERT RR in the proposed algorithm is shown as following Exp. 3.

$$\begin{cases} (2+\mathrm{LogU})\mathrm{Qtime} + \mathrm{RSAtime} & \text{if first query} \\ \mathrm{Qtime}+\mathrm{CHKtime} & \text{if second or later query} \end{cases} \quad \text{(Exp. 3)}$$

Where Qtime is DNS query processing time, RSAtime is RSA processing time, and CHK time is the time for checking if the requested CERT RR exists in certificate revocation list CERT RR or not.

At the first query, the time shown in Exp. 1 is required, and after the first query, the CERT RR will be cached. At the second or later query, it requires only DNS query time and the time for checking with CHK type because the proposed algorithm can use the cached CERT RR. And the check time will be O (LogM), where M is the number of the elements of certificate revocation list CERT RR and O (f(x)) means big-O of f(x). While the stability of consistency of the proposed algorithm is the same with that of consistency gained from Exp. 1, the processing time of the proposed algorithm is much less than that required in Exp. 1. And the size of table for database required for the proposed algorithm is much less than that of Exp.2. The database required in the proposed algorithm is the just information for mapping CERT RR and related DNS IP address. The size of this database is so small that it does not have any trouble to be implemented. The Table 1 summarizes the delay and space for information of each method of Exp1, Exp2 and Exp3(proposed algorithm).

(Table 1) The summary of each method

	Delay	Space
Exp.1	takes much delay	none
Exp.2	none	takes big space
Exp.3	take not much delay	takes small space

Ⅳ. Conclusion

When a user in Internet wants to communicate with another user, the message from a user can be eavesdropped, unauthorizedly modified and so on. It makes the users encrypt their message using a certificate taken from PKI. If PKI is organized by DNS, inconsistency of certificates can occur in name servers. In this paper, we proposed a method to solve inconsistency of certificates between primary name servers and secondary name servers, and between users and name servers. To solve inconsistency between primary and secondary name servers, we adapted NOTIFY opcode introduced in RFC 1996. To solve inconsistency between users and name servers, we proposed a new algorithm which did not take much time to get certificate and did not require much space to store information. With the proposed adaptation and algorithm, DNS can eliminate the inconsistency occurring when PKI is constructed with DNS. As a result, it can be guaranteed to communicate securely in PKI by using the proposed adaptation and algorithm. We remained mathematical proof for consistency of certificates as further study.

References

[1] Bruce Schneiner, *Applied cryptography Second Edition*, John Wiley & Sons, Inc., 1996.

[2] W.Ford, *Computer Communications Security: Principles, Standard Protocols and Techniques*, PTR Prentice Hall, 1994.

[3] Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF RFC 2459, 1998.

[4] Domain Name System Security Extensions, IETF RFC 2535, 1999.

[5] Storing Certificates in the Domain Name System, IETF 2538, 1999.

[6] Chan-Soon Im, Ok-Hwan Byeon, Young-Chul Shim, "A Public Key Infrastructure based on the Secure DNS," Proceedings of 1998 International Computer Symposium, pp.57-70, December, Taiwan, 1998.

[7] Domain Name-Concepts and Facilities, IETF RFC 1034, 1987.

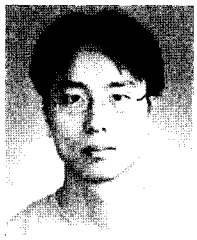
[8] Domain Name-Implementation and Specification, IETF RFC 1035, 1987.

[9] Dynamic Updates in the Domain Name System, IETF RFC 2136, 1997.

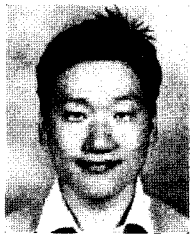
[10] Paul Albitz and Cricket Liu, *DNS and BIND Second Edition*, O'Reilly & Associates, Inc., 1997.

[11] A Mechanism for prompt Notification of Zone Changes (DNS NOTIFY), IETF RFC 1996, 1996.

〈著者紹介〉



석 우 진 (Woo-jin Seok)
1996년 2월 : 경북대학교 컴퓨터공학과 졸업
1998년 2월 : 광주과학기술원 정보통신공학과 석사
현재 : 연구개발정보센터 슈퍼컴퓨팅사업단 근무
〈관심분야〉 인터넷 QoS, ATM, 인터넷 보안



이 만 희 (Man-hee Lee)
1995년 2월 : 경북대학교 컴퓨터공학과 졸업
1997년 2월 : 경북대학교 컴퓨터공학과 석사
현재 연구개발정보센터 슈퍼컴퓨팅사업단 근무
〈관심분야〉 병렬알고리즘, 인터넷 보안, Network Management, Traffic Measurement

변 옥 환 (Ok-hwan Byeon) 종신회원
1979년 : 한국항공대학교 통신정보공학과 학사
1985년 : 인하대학교 전자공학과 석사(망보안)
1993년 : 경희대학교 전자공학과 박사(망관리)
1978년 9월-현재 : KIST 시스템공학연구소/ETRI/KORDIC 책임연구원
1983년 12월-1984년 12월 : 미국 OSM Computer Corp. 객원연구원
1997년 4월 -1998년 4월 : 미국 UIUC/NCSA 객원연구원
1995년 - 현재 한국통신정보보호학회/한국정보처리학회 이사
〈관심분야〉 컴퓨터네트워크,망관리 및 보안, 슈퍼컴퓨팅인프라응용