1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

A Public Key Infrastructure based on the Secure DNS

Chan-Soon Im*, Ok Hwan Byeon*, Young-Chul Shim**

* ETRI, Supercomputer Center, Taejon, Korea
** Hong-Ik University, Dept. of Computer Engineering, Seoul, Korea
Email : *{csim,ohbyeon}@garam.kreonet.re.kr
**shim@cs.hongik.ac.kr

ABSTRACT

In recent years, asymmetric key cryptographic system s are widely used for the security of e-mail, WWW, an d electronic commerce applications. For the proper use of asymmetric key cryptographic systems in these appl ications, one needs a system called a public key infrastr ucture. The public key infrastructure is a system in wh ich a user can register his public key and obtain other user's public key. In the public key infrastructure, a us er's public key certificate is generated by a certification authority and stored in a directory system. In this pap er we propose to use the DNS, which is already providi ng naming services world-wide, as the directory system of a public key infrastructure covering users all over t he world. The extensions required to the DNS to store certificates and certification revocation lists are explain ed and interfaces with which certification authorities ca n store new certificates and certification revocation lists are described. And the procedures with which a user can retrieve a certificate from the DNS are presented.

1. INTRODUCTION

In recent years as more people are connected to and use the Internet, the issue of the security in the Internet is becoming more important. Many mechanisms and t ools are used to provide required security services in th e applications of the Internet. They include firewalls, a uthentication mechanisms, cryptographic systems, securit y vulnerability checking systems, security audit systems, and intrusion detection systems. Among these mechan isms, to protect messages from eavesdropping and unaut horized modification of their contents, cryptographic sys tems are most widely used. Using cryptographic syste ms, it is possible to verify the identity of the remote pa rty participating in a communication connection, verify the identity claiming to be that of the originator of a m essage, and protect messages from being disclosed or ill egally modified/deleted. There are two types of cryptogr aphic systems: symmetric key cryptographic systems an d asymmetric key cryptographic systems. In symmetric key cryptographic systems, the same key is used in the encryption and decryption of messages and its example s include DES and IDEA. This key should be kept sec ret by both parties participating in a communication con nection. In contrast, asymmetric key cryptographic syst ems use complementary pairs of keys to separate the fu nctions of encryption and decryption. One key, the pri vate key, is kept secret like a key in symmetric key cry ptographic systems. The other key, the public key, doe s not need to be kept secret. This two-key approach ca n simplify key management, by minimizing the number of keys that need to be managed and stored in a netw ork, and can enable keys to be distributed via unprotect ed systems such as public directory services. Examples of asymmetric key cryptographic systems include RSA [1].

Asymmetric key cryptographic systems are widely us ed for the protection of many Internet applications such as electronic mail systems, WWW systems, and electro nic commerce systems[2]. In systems using asymmetric key cryptographic systems, if a user A wants to comm unicate with a user B, A needs to know B's public key. A can get B's public key from B directly or some tru sted third party. Whatever the case may be, it is impor tant to assure that B's public key does not come from an attacker who pretends to be the user B. This leads t o public keys being distributed in the form of certificat es. A certificate, generally speaking, is a data structure which is digitally signed by some party which users of the certificate will trust. A public key certificate, or j ust a certificate in our paper, is a data structure which binds the identifier of some party with a public key val ue. The certificate data structure is digitally signed by some other party known as a certification authority. Pu blic key certificates can be stored and distributed in an unprotected way, including publication in a directory w hose services are not necessarily trusted. Provided that a user knows in advance the authentic public key of th e certification authority, that user can check the validity

1998 International Computer Symposium
Workshop on Cryptology and Information Security
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

of the signature on the certificate. If this checks correctly, the user can be confident that the certificate carries a valid public key for the identified party[1].

The system in which a user can register his public key and obtain other user's public key is a public key infrastructure. In the public key infrastructure, 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. The certificate is stored in a directory system, and other users can retrieve the certificate from this directory system. Although some proposals, e.g., an Internet PEM (Privacy Enhanced Mail) public key infrastructure, have been made to build a public key infrastructure covering users world-wide[3,4], only a small scale infrastructures have actually been built and used covering a small group of users belonging to an organization or using a certain Internet application. To be able to build a world-wide public key infrastructure, we need both a world-wide standard naming scheme for users and machines and a world-wide directory systems.

The Internet Domain Name System (DNS) provides a standard method for naming machines and has a world-wide directory system from which users can get the IP address of a machine using its domain name or vice versa. Recently a proposal has been made to extend the DNS so that users can retrieve public keys and/or certificates from the DNS securely[5,6]. So this DNS with security extensions, which will be called the secure DNS in our paper, will be an excellent candidate for building a world-wide public key infrastructure. But the current proposal of the secure DNS (1) does not include user certificates, (2) does not provide a certification authority interface with which certification authority can publish certificates and certificate revocation lists(CRLs), and (3) does not provide guidelines how users can obtain user certificates from the secure DNS. In this paper we will further extend the secure DNS so that these three deficiencies can be made up for and, therefore, the secure DNS can be used as a building block in the world-wide public key infrastructure.

The organization of the paper is as follows. Section 2 and Section 3 briefly introduce the public key infrastructure and the secure DNS, respectively. Section 4 explains our approach for extending the secure DNS so that it can be used as a directory system in the public key infrastructure and followed by the conclusion in Section 5.

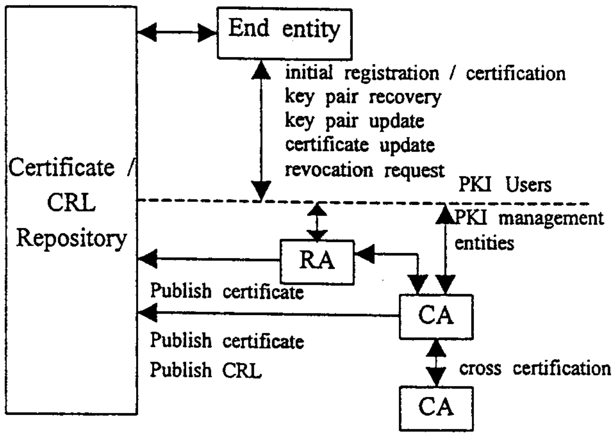

Figure 1. Components of a public key infrastructur

2. The Public Key Infrastructure(PKI)

Figure 1 shows the components of a PKI and relationships among them in X.509 standards[7,8]. Each component is defined as follows.

- End entity : User of PKI certificates and/or end user system that is the subject of a certificate
- CA : Certification authority
- RA : Registration authority, i.e., an optional system to which a CA delegates certain management functions
- Repository : A system or collection of distributed systems that store certificates and CRLs and serves as a means of distributing these certificates and CRLs to end entities

An end entity performs the following management operations with either CA or RA.

- Initial registration/certification : This is the process whereby an end entity first makes itself known to a CA or RA. The end result of this process is that a CA issues a certificate for an end entity's public key, and returns that certificate to the end entity and/or posts that certificate in a public repository.
- Key pair recovery : As an option, a user's private key may be backed up by a CA and the user can recover his private key in case that he loses it.
- Key pair update : Every key pair needs to be replaced with a new key pair, and a new certificate needs to be issued.
- Certificate update : As certificates expire they may be refreshed if nothing relevant in the environment has changed.
- Revocation request : An authorized person can advise a CA of an abnormal situation requiring certificate revocation.

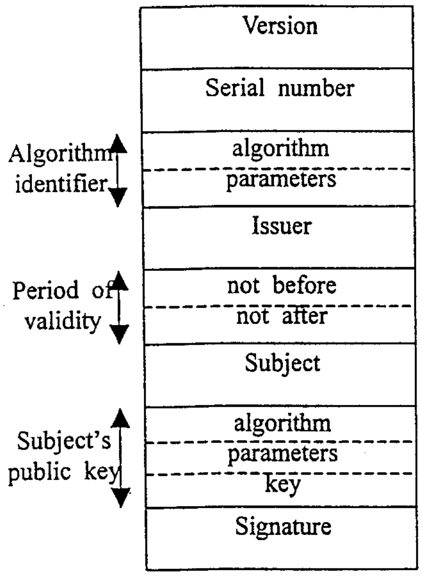

Figure 2. X.509 Certificate

The X.509 certificate format is as in Figure 2

- Version : Differentiates among successive versions of the certificate format: the default is 1988.
- Serial number : An integer value, unique within the issuing CA, which is unambiguously associated wi th this certificate.
- Algorithm identifier : The algorithm used to sign th e certificate, together with any associated parameter s.
- Issuer : The CA that created and signed this certifi cate.
- Period of validity : Consists of two dates: the first and last on which the certificate is valid.
- Subject : The user to whom this certificate refers
- Public-key information : The public key of the subj ect, plus an identifier of the algorithm for which th is key is to be used.
- Signature : Covers all of the other fields of the cert ificate, and consists of a hash code of the other fiel ds, encrypted with the CA's private key.

The standard uses the following notation to define a certificate:

CA <<A>> = CA {V, SN, AI, CA, T_A, A, A_P}

where

Y<<X>> = the certificate of the user X issued by c ertification authority Y

Y{I} = the signing of I by Y. It consists of I with an encrypted hash code appended

Each certificate includes a period of validity. Typica lly, a new certificate is issued just before the expiration of the old one. In addition, it may be desirable on oc casion to revoke a certificate before it expires for one o f the following reasons:

- The user's private key is assumed to be compromis ed.
- The user is no longer certified by this CA.
- The CA's private key is assumed to be compromis ed.

Each CA must maintain a list consisting of all revoke d but not expired certificates issued by that CA. These lists are called CRLs and should be posted on the dire ctory. A CRL contains the following information.

- Signature : Identical to the Signature field in certifi cates. This specifies the algorithm used to comput e the signature on this CRL.
- Issuer : Identical to the Issuer field in certificates.
- This update : Contains the time the CRL was issue d, specified as UTC.
- Next update : Contains the time the next CRL is e xpected to be issued, specified as UTC.

The following two fields repeat as a pair once for each revoked certificate:

- User certificate : Contains the serial number of the revoked certificates.
- Revocation date : Contains the UTC time the certif icate was revoked.

One disadvantage of the X.509 system is that a large and complex PKI should be availabe before users can register and retrieve public key certificates. PGP propo ses and uses a clever and simple method for managing public keys[9, 10]. In PGP, users can obtain other use r's public key certificate from that user, other users, or key servers using e-mail, FTP, etc. But a PGP certifica te does not include neither user names nor signatures. A PGP certificate just includes a public key, its algorith m, and its validity period. So a PGP certificate should be accompanied by one or more user ID packets and z ero or more signature packets. A user ID packet has a user ID string which is normally an e-mail address. A user can have many e-mail addresses but only one pub lic key. In this case all these e-mail addresses will be associated with this public key. A public key along wit h its user ID can be optionally signed and this signatur e can be stored in a signature packet. One reliable met hod for distributing public keys in PGP environments is using trusted key servers. Users register their public k eys to the key servers and obtain other users' public ke ys from the key servers. When a user registers a publi c key to a key server, he sends the public key and a user ID packet securely to a key server. Then the key server signs the certificate and the user ID packet and distributes them on other users' demand. One big difference between X.509 and PGP is that X.509 requires a large hierarchy of certification authorities but PGP key servers need not form a hierarchy and build trust relationship among them. Therefore PGP users can start with only one key server.

In PGP the convention for revoking a public key is for a user to issue a key revocation certificate, signed by the user. This certificate has the same form as a normal signature certificate but includes an indicator that the purpose of this certificate is to revoke the use of this public key.

3. The Secure DNS

The secure DNS provides the following three security services.

- Key distribution : Entities such as zones, hosts, and users have a pair of a private key and a public key. The public keys are stored in the DNS server and distributed on demand.
- Integrity and data authentication of the information stored in the DNS server : All the resource records that are the information stored in the DNS server are signed by the DNS server and this signature is stored as a separate resource record.
- Integrity and data authentication of DNS queries and responses : The DNS header and content of a DNS query or response is signed by the private key of the DNS client or DNS server, respectively.

More detailed explanation of these security extensions follow.

3.1. Key distribution

All the entities such as zones, hosts, and users have a pair of private key and a public key and the public key is stored as a KEY resource record in the DNS server.

foo.host.example.   IN KEY   RDATA

RDATA contains the public key of foo.host.example with the format in Figure 3.

0 15	16 23	24 31
Flags	Protocol	Algorithm
Public Key		

Figure 3 KEY RDATA format

The flag indicates whether the key belongs to a user, a host, or a zone and specifies for what purposes this key can be used. The protocol indicates what protocol including DNS can use this key. The algorithm specifies the asymmetric key algorithm of the public key.

The DNS server also stores certificates for entities and CRLs. They are stored as resource records of a CERT type as follow

foo.host.example.   CERT   RDATA

and RDATA contains following information.

- Type : Specifies whether the certificate is a X.509, SPKI, or PGP type.
- Key tag : Specifies which public key in the KEY resource records the public key in this certificate corresponds to.
- Certificate or CRL.

3.2. Integrity and data authentication of resource records

The integrity and data authentication of resource records in a DNS server is basically provided by SIG resource records. A SIG resource record contains a signature for some other resource record. The signature is made with the private key of the zone which the signed resource record belongs to. A SIG resource record is stored with the following format

foo.host.example.   IN SIG   RDATA

and the RDATA contains the following information.

- Type : Specifies whether the type of the signed resource record is an NS, A, MX, or CNAME type.
- Algorithm : Indicates what hash algorithm and asymmetric key algorithm are used for the signature.
- Signature expiration : Tells when the signature expires.
- Time signed : Specifies when the signature was made.
- Signer's name
- Signature

When a name server returns KEY, A, CNAME resource records as an answer, it also sends the corresponding SIG resource records and the resolver receiving the answer checks the integrity of the answer by verifying the signatures in the SIG resource records.

3.3. Integrity and data authentication of a query/respons

e

The content of a DNS query/response and the DNS header are signed by the private key of the DNS client or server so that the integrity and data authenticity of t he query/response can be guaranteed.

4. Building a PKI using secure DNS servers as a Repository

In this Section we explain how we can build a PKI using DNS servers as a public key repository.

4.1. Storing certificates and CRLs in a DNS server

We assume that certification authorities in an X.509 PKI form a hierarchical structure like the servers in the DNS. We also assume that PGP users registers and st ores in the repository their public keys through PGP cer tification authorities. One certification authority is asso ciated with only one DNS server and, therefore, stores t he certificates and CRLs that it publishes through this DNS server. But there can be DNS servers which do n ot have any associated certification authority. For exam ple, in a university having many departments, there can be one DNS server for the university and many child DNS servers for the departments. If there is one certifi cation authority for the whole university, this certificatio n authority is associated with the university DNS server while the department DNS servers do not have any cer tification authorities associated. If a DNS server and a certification authority are associated with each other, th e DNS server registers its public key to that certificatio n authority and has the public key certificate of that cer tification authority while the certification authority's na me is registered in that DNS server.

A user or service is identified with its e-mail address which is unique in its zone. And this unique e-mail a ddress is used as the subject name in the certificate. In the case of PGP, this does not cause any problem, bec ause PGP already uses e-mail address as the subject na me of certificates. But this can be a problem with X.5 09 certificates, because the distinguished name, which i s an ordered list of {attribute, value} pair, is used in th e subject field and it is quite different from the e-mail address in the Internet. But this problem can be solved using the optional Subject Alternative Name field whic h allows additional identities to be bound to the subject of the certificate. We store the unique e-mail address in the Subject Alternative Name field and leave the Sub ject field empty in an X.509 certificate.

Differently from the IETF draft, we use two resource record types, CERT and CRL, for certificate and CRL resource records, respectively. In the following X.509 e xample, if a university with a domain name hongik.ac.k r has a user with an e-mail address shim@hongik.ac.kr and a certification authority with a service name ca@ho ngik.ac.kr, then the university DNS server will have the following entries in its database.

```
shim.hongik.ac.kr.  CERT  RDATA-1
                    ; certificate for the user shim
ca.hongik.ac.kr. CERT RDATA-2
                    ; certificate for the certification
                    ; authority
ca.hongik.ac.kr. CRL RDATA-3
                    ; CRL published by
                    ; ca.hongik.ac.kr
```

We use different resource record type names for certific ates and CRLs because we want to easily tell apart wh ether the RDATA associated with a certification authorit y is a certificate or a CRL. The RDATAs for the CER T and CRL type resource records have two fields: type and data. The type field specifies whether the certific ate or CRL is X.509, PGP, SPKI, or other type and the data field has the actual certificate or CRL. We did n ot include the key tag field as in the IETF draft becaus e we believe that it is not necessary to store the same public key of a certificate in a separate KEY resource r ecord again.

A DNS server regularly examines all the certificates in its database to check if the validity time of certificat es has expired. If it finds such a certificate, it deletes t hat certificate from its database.

4.2. Publishing certificates and CRLs

In X.509 a certification authority stores certificates a nd CRLs in the associated DNS server. A user register s his public key by sending it to an X.509 certification authority. The certification authority generates a certific ate by signing the public key and sends the certificate t o the associated DNS server. When the DNS server re ceives the certificate from the certification authority, it verifies the validity of the certificate by checking the si gnature in the certificate with the certification authority' s public key. If the verification is successful, it retrieve s the owner name of the certificate from the Alternative Subject Name field in the certificate and stores the cer tificate in its database as the resource record of a CER T type. When a DNS server receives a CRL from the certification authority, it verifies the validity of this CR

L in the same way and stores in its database as the resource record of a CRL type. The certification authority's name is used as the name for this CRL resource record. The DNS server also deletes from its database any public key certificates which were revoked in the received CRL.

A PGP user registers his public key by sending the public key and one more user IDs associated with this public key to the PGP certification authority. The certification authority generates one certificate for each user ID in the user ID packet. For example if a public key is registered with two user IDs, shim@hongik.ac.kr and shim@cs.berkeley.edu, two CERT type resource records will be prepared as follows:

shim.hongik.ac.kr.	CERT	RDATA
shim.cs.berkeley.edu.	CERT	RDATA

The type field of RDATA has the value PGP and the data field has the public key, the user ID packet containing both shim@hongik.ac.kr and shim@cs.berkeley.edu, and the signature packet generated by the certification authority. The certification authority sends these two CERT type resource records to the associated DNS server. This DNS server first checks the signature of two certificates and stores them in the appropriate DNS servers. So the first certificate will be stored in the DNS server in charge of the zone 'hongik.ac.kr' and the second certificate will be stored in the DNS server managing the zone 'cs.berkeley.edu'. So this registration requests should be forwarded to the properDNS servers as in the case of DNS queries.

A user revokes his public key by sending a key revocation certificate to the PGP certification authority. Because this revocation certificate is already signed by the user, the certification authority checks the signature, extracts all the user IDs associated with this public key, generates one CRL type resource records for each user ID, and stores them in the appropriate DNS servers as when PGP certificates are stored in the appropriate DNS servers. So if shim's public key is revoked, then the following two CRL resource records will be generated by the PGP certification authority and stored in the appropriate DNS servers.

shim.hongik.ac.kr	CRL	RDATA-1
shim.cs.berkeley.edu	CRL	RDATA-1

A DNS server has two interfaces: an operational interface and a management interface. The operational interface is used by a DNS client to submit DNS queries and receive DNS replies and uses UDP in general. The management interface is used by a certification authority to store certificates and CRLs in a DNS server. Instead of defining a new interface for the management, we use the extended feature of the DNS which allows dynamic updates of DNS databases[11]. When a user or manager wants to send some request to a DNS server, he prepares a query message by calling the resolver library routine 'res_mkquery' with appropriate parameter values. The res_mkquery is used as follow:

res_mkquery(op, dname, class, type, data, datalen, newrr, buf, buflen)

Among the many parameters, two parameters need extensions: operation_code(op) and type. Operation_code specifies what operation is to be requested. The usual values for this parameter are QUERY or IQUERY specifying a standard query or an inverse query, respectively But this parameter has been already extended for the dynamic updates of DNS servers and can have values such as UPDATEA, UPDATED, etc. The value UPDATEA is used to request to add some resource records and the value UPDATED is used to request to delete a specific resource record. To store CERT or CRL type resource records, the management request just needs to specify the operation_code of the request message to be UPDATEA. The type parameter specifies on what type of resource records the specified opertaion should be performed. And the possible values for the type parameter should be extended to include two new resource record types: CERT and CRL. And this extension should be made in the file /include/arpa/nameserv.h. Packets containing certificates will be short but packets containing CRLs can be long in X.509 because a CRL can contain many revoked certificates. So we use TCP for the management protocol.

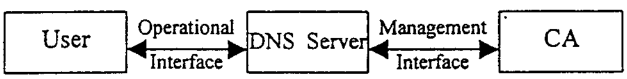

Figure 4. Interfaces for DNS server

4.3 Retrieving certificates from DNS servers

To retrieve certificates or CRLs from a DNS server, a

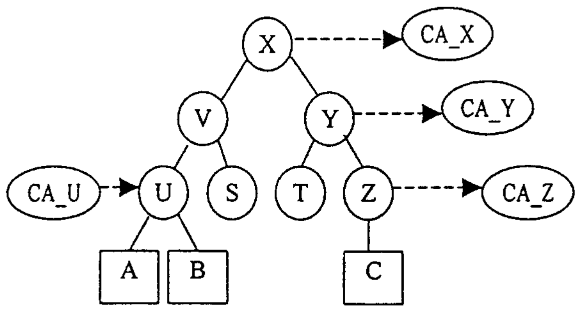

Figure 5. An example hierarchy of DNS servers

DNS client makes a query message using the resolver li brary routine, res_mkquery. The operation_code parame ter is set to be QUERY and the type parameter is set t o be either CERT or CRL depending on whether the cli ent wants to retrieve a certificate or CRL.

First we explain how a X.509 certificate can be retri eved from DNS servers. Figure 5 shows a hierarchy of DNS servers named S, T, U, V, X, Y, and Z. Among them, the DNS servers, U, X, Y, and Z, have their asso ciated certification authorities. Two users, A and B, ar e registered in the zone U.V.X and one user C is regist ered in the zone Z.Y.X. We assume that the user A ha s the certificates of its own certification authority CA_ U and the root certification authority, CA_X. If A wan ts B's certificate, the query is sent to the DNS server U and U returns B's certificate signed with CA_U's priv ate key. Because A has the public key of CA_U, it ca n check the validity of the reply.

If A requests the certificate of the user C who is not in the same zone, the request can be processed in an it erative method or a recursive method. In the case of t he iterative method, the query is processed as in Figure 6 and each message carries information as follows:

(1) U requests the root DNS server X of C's certificate.
(2) The root returns the address of Y.
(3) U requests Y of C's certificate.
(4) Y returns the address of Z
(5) U requests Z of C's certificate.
(6) Z returns C's certificate

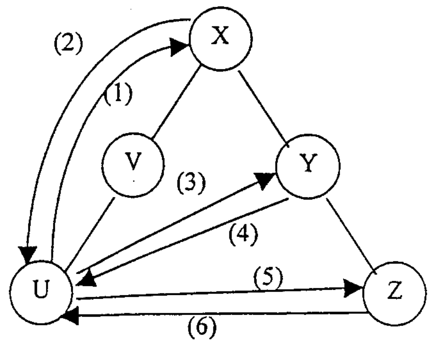

Figure 6. Processing a DNS query in the iterative method

Now in the message (6), Z will return C's certificate si gned by CA_Z's private key. But because the user A does not have CA_Z's public key, he cannot check the validity of the received certificate. So he needs CA_Y' s certificate to check the validity of CA_Z's public key, because CA_Z's public key will be signed by CA_Y.

In summary he needs the certificate path as follows:

CA_X<<CA_Y>>, CA_Y<<CA_Z>>, CA_Z<<CA_C>>

The user A can get this complete certificate path by ma king each DNS server return the certificate of its associ ated certification authority whenever it returns the addre ss of other DNS servers or users' certificates. So X, Y, and Z will return the certificate of CA_X, CA_Y, CA_ Z, respectively. Because A has the root certification au thority's certificate, the root DNS server need not retur n the certificate of the root certification authority. All t hese certificates will be gathered at the user A and the validity of the C's certificate can be verified. C's certi ficate is stored in the answer section and the certificates of all the certification authorities will be stored in the

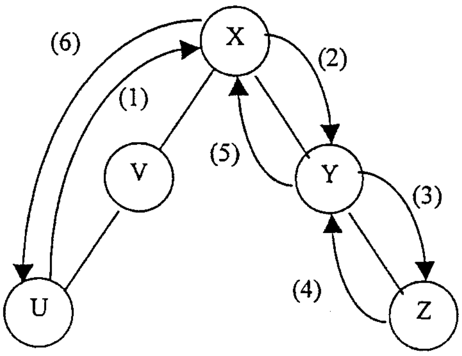

Figure 7. Processing a DNS query in the recursive method

additional section in the DNS reply message.

In the case of the recursive method, the query is proces sed as in Figure 7 and each message contains informati on as follows:

(1) U asks the root server, X, of C's certificate
(2) X asks Y of C's certificate.
(3) Y asks Z of C's certificate.
(4) Z returns C's certificate to Y.
(5) Y relays C's certificate to X.
(6) X relays C's certificate to U.

The user A requires the same certificate path as in the iterative method. It can be made possible by making e ach DNS server append its associated certification autho rity's certificate when it returns a user certificate or rela ys an answer. So in our example the user A will find the certificates of CA_X, CA_Y, and CA_Z in the addit ional section of the DNS reply in addition to the certifi cate of the user C in the answer section of the same D NS reply.

But when a user wants a PGP certificate, he just need s to retrieve the certificate and does not need the whole certificate path as in the X.509.

When a DNS server cannot find the requested certifi cate, an error message will be returned. And the integrit y of the data authenticity of the reply message includin g this error message will be assured by the security fea tures of the secure DNS. If the integrity of this reply message is not maintained, an attacker can capture any reply message including a certificate, throw this reply a way, and send a false reply saying that the DNS could not find the requested certificate. And this scenario wil l be a denial of service attack to the DNS servers.

When a user receives a certificate from a DNS serve r, he can store it in its public key table and use it agai n in later times. But a problem arises, because he cann ot know whether the public key has been revoked after he received the public key from the DNS server. We can consider several solutions to this problem. The firs t method is not to use the information in the public key table and ask the DNS server to bring the requested p ublic key every time a user requests the public key. T he second method is to contact the DNS server which gave the certificate and bring the CRLs from it. Then the user checks whether the public key has been revoke d using the CRLs. The disadvantage of this second me thod is that he needs to contact a DNS server and brin g CRLs which can be very long whenever he intends t o use a public key in its public key table and this meth od incurs more communication overhead than the first method. But an advantage of this second method is tha t when it gets CRLs, it can find and delete any certific ates which were received from the same server but wer e revoked thereafter. The third method is for a DNS se rver to distribute the CRLs to clients whenever it receiv es a new CRL from a certification authority. But this method is not practically possible to implement, becaus e it is difficult for the name server to determine the list of clients who will be interested in this new CRL.

5. CONCLUSION

In this paper we explained how the DNS could be us ed as a distributed repository of public key certificates and CRLs in a world-wide public key infrastructure. We assumed that a certification authority is associated with one DNS server and stores all the certificates and CRLs it publishes in that DNS server. We use the e-m ail address of a user as the subject name of a certificat e. And the DNS uses CERT and CRL resource record types to store certificates and CRLs. A certification a uthority uses the extended dynamic update protocol in t he DNS and TCP to store certificates and CRLs. New parameter values are added to the resolver library routin es and the procedures which processes the DNS query t o retrieve certificates are extended so that a user can re quest certificates from the DNS and receive the certific ate path. The security features of the secure DNS are used to maintain the integrity and data authenticity of t he DNS queries and replies. One of the problems that remain to be solved is how to efficiently CRLs to the i nterested users.

6. REFERENCE

[1] W. Ford, Computer Communications Security: Prin ciples, Standard Protocols and Techniques, PTR Pr entice Hall, 1994.

[2] S. Garfinkel & G. Spafford, We Security & Comm erce, O'Reilly & Associates, Inc., 1997.

[3] IETF RFC 1422, Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key M anagement, 1993.

[4] IETF RFC 1424, Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and R elated Services, 1993.

[5] IETF RFC 2065, Domain Name System Security E xtensions, 1997.

[6] IETF Draft, Storing Certificates in the Domain Na me System, 1998.

[7] IETF Draft, Internet X.509 Public Key Infrastructu re Certificate and CRL Profile, 1998.

[8] IETF Draft, Internet X.509 Public Key Infrastructu re Certificate Management Protocols, 1998.

[9] C. Kaufman, R. Perlman, & M. Speciner, Network Security: Private Communication in a Public Worl d, PTR Prentice Hall, 1995.

[10] W. Stallings, Network and Internetwork Security: P rinciples and Practice, Prentice Hall, 1995.

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