Submitted to Computer Communications
Special Issue: Securing Computer Communications with Public Key Infrastructure

Virtual Certificates and Synthetic Certificates: New Paradigms for Improving Public Key Validation

Selwyn Russell[1], **Ed Dawson**[1], **Eiji Okamoto**[2], **Javier Lopez**[3]

[1] Information Security Research Centre, Queensland University of Technology, Brisbane, Australia
e-mail: s.russell,e.dawson@qut.edu.au
[2] University of Tsukuba, Japan e-mail: okamoto@is.tsukuba.ac.jp
[3] University of Malaga, Spain e-mail: jlm@lcc.uma.es

Abstract The certificate paradigm is applied recursively to obtain the public keys of a number of Certification Authorities and, accordingly, to obtain the public keys of a number of final entities. Thus, validation of the authorized public key of a party in a network transaction is commonly based on processing the certificate chain descended from a trusted root issuer, involving non-negligible time and cost. Those chains become long in communications between large organizations, which is the typical case of e-commerce and e-government applications. The process of validation of extensive chains introduces performance problems in two aspects: signature verification and revocation checking. That is, the repeated processing of long chains of certificates creates severe efficiency problems. This fact causes that most of the advantages provided by Public Key Infrastructures (PKIs) are not conveniently exploited. In this paper we analyze the scenarios in which large volumes of digitally signed transactions between commercial entities exist. These cases require of interoperation among PKIs. We show that solutions available in those scenarios still involve processing of too long chains of certificates, either at the receiving computer or by an outsourced entity. For this reason, we propose new concepts of *virtual certificate* and *synthetic certificate* for faster and less costly processing of certificate chains. In this way, communications in a certificate-based intercommunity can be highly improved. We also show how these types of certificates can be applied in practice.

1 Introduction

The main goal of a *Public Key Infrastructure* (PKI) is to facilitate the use of public key cryptosystems. A PKI is a vital element because it enables the application of those cryptosystems to the exchange of digitally signed information between parties that do not have a face to face interaction. It does this by providing an efficient and trustworthy mean to manage public-key values, thus simplifying the process of determining the public key of a particular entity in the community.

As it is known, in order to prevent impersonation attacks, the key must be provided in a *digital certificate* which is itself signed by a well known issuer, a *Certificate Authority*, or also *Certification Authority* (CA).

A single issuer of public key certificates is not a scalable solution for a worldwide user base. More issuers are needed, and this led to the concept of a hierarchy of issuers. The justifying logic is based on the transitive assumption that if A has certified the public key of B in a certificate C_{AB} and B has issued to C a public key certificate C_{BC}, and D trusts A to correctly issue certificates, then D believes C's public key is as in certificate C_{BC}. The commonly accepted term for this is *chaining of certificates* (also, *certification path*).

Chains arise in PKIs which seek to provide strong assurance of the identity of the certificate subject, such as might be used in e-commerce and e-government activities. Validation of these chains introduces performance problems in two aspects: signature verification and revocation checking. That is, the repeated processing of long chains of certificates, which is typical in communications between large organizations, creates severe efficiency problems in the systems.

This fact causes that most of the advantages provided by PKIs are not conveniently exploited. This work analyzes in detail that problem and presents a solution based on the use of two new types of data structures, *virtual certificates* and *synthetic certificates*. The paper is structured as follows. In section 2 general problems with certificate hierarchies are reviewed. Section 3 focus on the analysis of scenarios with single domain PKIs. Section 4 examines potential problems of hierarchical systems when attempting to communicate with another domain, namely PKI interoperation. The techniques of cross certification and the use of a certificate bridge are discussed here too. In section 5, we define virtual certificates and synthetic certificates, and show how communications in a certificate-based inter-community can be improved. Finally, section 6 ends the paper with the conclusions and future work.

2 Problems With Certificate Hierarchies

In the simplest case, Alice and Bob each have a certificate issued by the same Certificate Authority. The first time either sends a message to the other, the message is accompanied by the single certificate of the sender. In order to verify the signature on the certificate, the public key of the certifier is already known. There is no doubt that when considering only this type of scenario, management of public keys becomes an easy task because a unique CA is involved in certificates issuance.

However, and as stated, more authorities are needed because it is unlikely that a unique CA is capable to establish adequate relationships with all Internet users. A more realistic scenario is that one in which sender and receiver of digitally signed information do not share the same CA. If Alice sends a message to Charles, who has a certificate issued by a different authority, she has to ensure that Charles knows the public key of her CA. This can be achieved by including a second certificate, a self-signed certificate from Alice's authority. However, if Charles has no prior information on Alice's Authority, a risk for Charles is that the message is not from Alice but from an impersonator who has self signed a certificate using the Certificate Authority's name and then issued another fraudulent certificate, with Alice as the subject, with a public key created by the impersonator.

2.1 Compromise of Root CA Key

A practical risk here is that the Certificate Authority's private key could be compromised and then all of the current certificates which have ever been issued with that key will need to be re-issued. The certificates which had been issued by the revoked root key will in turn need to be revoked and new certificates issued, and so on down to the end certificates.

One technique to reduce the damage caused by a compromise is to use a separate root key especially for members of a particular community, then a compromise of that key affects only that community. An example which is found in common desktop browsers is the set of special purpose root keys produced by the Digital Signature Trust Company for use by some of their clients: one for the National Retail Federation, one for the United Parcel Service, another for the ANX Network. [1]

Another way to reduce the fallout from a *Root CA* (RCA) key compromise is to structure an issuer as a multi-stage hierarchy. The principle is that a cryptanalyst will be less likely to succeed if the number of pairs of plaintext - ciphertext in her possession is kept small, so minimize the usage of higher root keys. A top level key is advertised in self-signed certificates but rarely used and is kept off line for highest possible security. A lower level key which is not advertised is used many times for creating certificates for lower level entities. It is at a higher risk because a cryptanalyst has more information, but its compromise results in less pain than the compromise of a single root key. This chaining reduces the risk, as shown in figure 1. In this case the Certificate Authority has generated another key pair, and has used the private key of the secondary pair to sign the certificates of Alice and Bob.

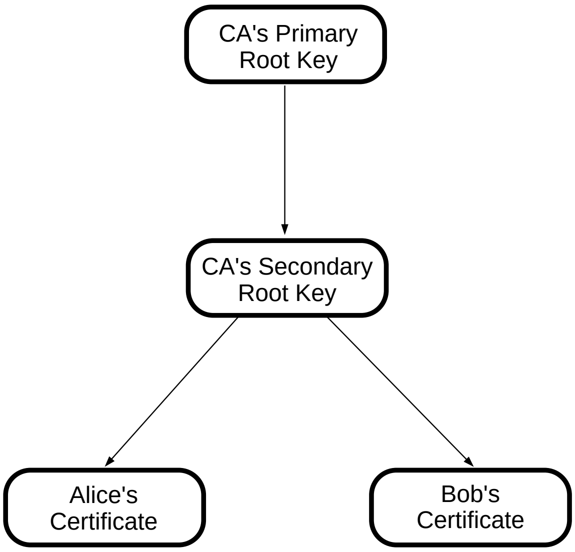

Fig. 1 Top Level and Low Level Root Keys

Under this multi-stage strategy, when Alice applies for a certificate, she receives $C_{RCA_{Secondary},Alice}$, a certificate for herself signed by the CA using the secondary key, and another stating the public key of the secondary level, signed by the top level, $C_{RCA_{Primary},RCA_{Secondary}}$. The top level public key is the one that is advertised to the world in $C_{RCA_{Primary},RCA_{Primary}}$ and usually has a long expected life of ten or more years. Bottom level certificates such as Alice's are likely to be issued with a life of only one year.

In practice, this can be extended further, so there are two or more secondary level root keys, say $N_{Secondary}$. The primary root key is used only to sign the N certificates, so there are at most only $N_{Secondary}$ sets of plaintext - ciphertext available to an attacker. Because it is rarely used, it can be kept off line, which enhances its security. On the other hand, each secondary certificate may be used to sign thousands of certificates, giving an attacker a much larger plaintext - ciphertext collection

[1] In MicroSoft Internet Explorer, go to *Tools* → *Internet Options* → *Content* → *Certificates* → *Trusted Root Certification Authorities*

for attacks. If a secondary root key is compromised, all of its certificates are rendered valueless, but the certificates produced from the other $N_{Secondary} - 1$ secondary root keys are unaffected.

A Certificate Authority might use more than two levels for added protection. It might have $N_{Primary}$ primary root keys, and primary root key $PrimK_i$ might have N_{PrimK_i} immediate descendants, $SecK_j$, and each secondary key might have immediate descendants, etc.

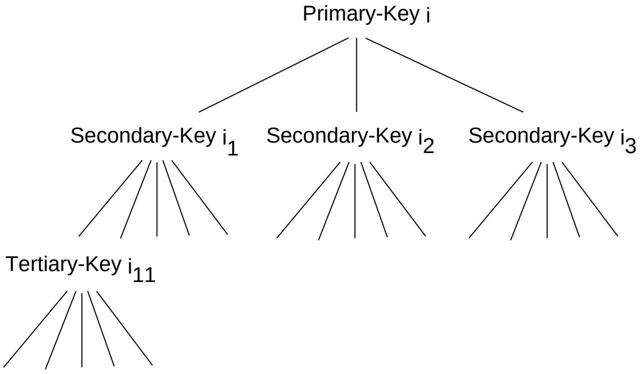

Fig. 2 Extension of levels for added protection

2.2 The Case of Large Enterprises

When large enterprises are involved, there are likely to be additional internal Certificate Authorities in the system, because an enterprise certificate may be obtained from an outside Certificate Authority, and issue certificates itself to staff members within the enterprise, acting as its own *Registration Authority* (RA).

In addition, the levels above the enterprise may be significant. For instance, if in South Korea [11], the enterprise would receive the certificate from one of the "Licensed Certificate Authorities", so it would start as at least a third level starting from the KISA Root Certificate Authority. For the reasons stated above, it might not use that certificate for signing certificates for staff, but create one or more subsidiary key sets which are used for staff certificates.

If the enterprise is large and is dispersed geographically, it may have a separate internal Certificate Authority for each region or functional group. In figure 3, which includes a national root and a separate level of public Certificate Authorities, the asterisks indicate the location of the advertised public keys.

If the enterprise has multiple businesses, each business will most likely have its own public key and manage its own internal certificate issuance. The tree grows as showed in figure 4.

In that tree, there is one National reference, which certifies M Major Issuers. Major Issuer 1 has been approached by the enterprise in question and has certified Enterprise Issuer 1 through Enterprise Issuer N, being N different parts of this enterprise. Each Enterprise Issuer has certified K internal regional issuers which certify local staff of the enterprise.

2.3 Processing Difficulties

A long chain of even version 1 X.509 certificates [5] can be difficult to process and will require well designed software. A sender may work in more than one section of an enterprise and may have a number of certificates which represent different paths from the root. A message may be accompanied by all of these certificates, and the receiver's software has to determine a valid path to the root. Complications may arise, for example a certificate in one path may have been recently revoked, but the other paths appear to be valid. Perhaps they have been revoked but have not shown up yet in the public records. Should the receiver seek reconfirmation of at least one of the other paths? What if three are invalid but one is valid? What if an attacker submits a "chain" with a loop?

Using version 3 and later X.509 certificates[8], makes automated processing more complex. Descriptions of policies may be included in the certificates and these should be tested for consistency. An obvious check is that each issuer of a certificate should be marked in its certificate as being authorized to do so. Extensions can be ambiguous, further complicating automated processing. Moreover, each extension contains a criticality indicator, that

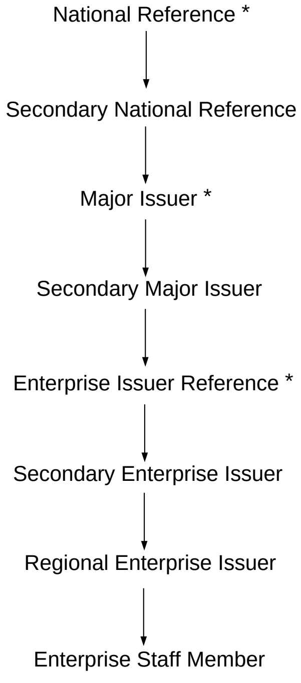
National Reference *
Secondary National Reference
Major Issuer *
Secondary Major Issuer
Enterprise Issuer Reference *
Secondary Enterprise Issuer
Regional Enterprise Issuer
Enterprise Staff Member

Fig. 3 Example of certification levels for a large enterprise

is, a flag that indicates whether an occurrence of an extension is critical or noncritical. Thus, a certificate may have been designed for a particular application and contain extensions marked as critical. Attempts to use that certificate for other purposes will cause a validation failure, even though the other content of the certificate is entirely satisfactory.

Processing time problems will not improve as hardware is advanced. The basis of cryptography is that an operation should be relatively time consuming otherwise an attacker can stage an exhaustive search, as has happened to DES [3], even though the DES algorithm is intact. As hardware runs faster, key sizes will be increased, other algorithms will emerge, etc.

3 Single Root Domain

A single hierarchy is the simplest way to issue certificates and the first implementation employed by most PKI builders. It is attractive for its simplicity and efficiency. Because the same root Certificate Authority is at the top of the certificate chain for each end certificate holder, it can act as a common point of trust, allowing all of the end points to communicate with one another.

A single CA/RA in a country essentially means it is controlled by the government, and objections have been expressed against government controls [2], including

- General distrust of monopolistic collector, fear of misuse
- General distrust of Government Departments / Agencies and what they might do with information about an individual which has been collected from many sources and aggregated to a single database. Some countries have strict laws regarding the collection of personal information. The European Union is generally stricter than the United States on this issue. Some jurisdictions forbid the collection of personal information on children under the age of twelve years.
- Some government agencies do not trust other agencies, so there cannot be a single top level point.
- Commercial vendors fear lost business opportunities if they are prevented by law from participating.

A single root system was attempted in the USA in 1993, when the Internet Engineering Task Force (IETF) published four Requests For Comments for a proposed Privacy Enhanced Mail (PEM) system, RFC 1421 through 1424 [13] [10] [4] [9]. The RFCs were quite comprehensive and proposed a hierarchical worldwide certificate issuing architecture. Technically the proposal was very good and could have led beyond email to other network applications but there was insufficient interest and an excess of disagreement over the structure. In its place we have a series of commercial interests issuing general certificates to paying customers, and government groups issuing to entities certificates which are intended to allow them to transact electronically with the government groups, e.g. renew a driving licence.

So the situation varies with the country, but at the moment each nation has, or expects to have, one or more certificate trees, with the root certifiers within a nation and between nations being independent and autonomous. Holders of certificates within each tree represent a community with some common interest in using their public key software. While these isolated islands of holders does not affect communications within a community, it raises barriers which hinder dialogue between groups.

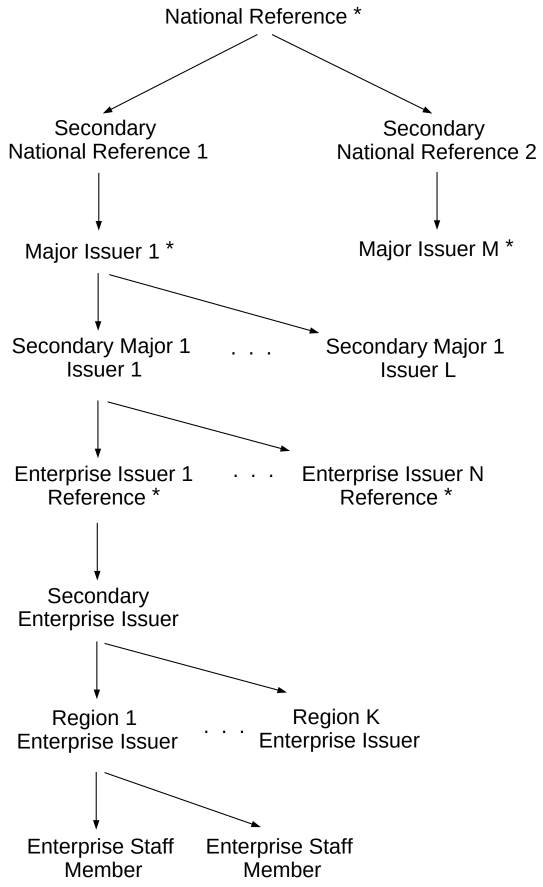
National Reference *
Secondary National Reference 1
Secondary National Reference 2
Major Issuer 1 *
Major Issuer M *
Secondary Major 1 Issuer 1
. . .
Secondary Major 1 Issuer L
Enterprise Issuer 1 Reference *
. . .
Enterprise Issuer N Reference *
Secondary Enterprise Issuer
Region 1 Enterprise Issuer
. . .
Region K Enterprise Issuer
Enterprise Staff Member
Enterprise Staff Member

Fig. 4 Large enterprise with multiple businesses

4 PKI Domain Interoperation

4.1 Strategic Options

There comes a time when a member of a community will wish to communicate securely with a member of another. This may be internationally where the person resides in a country with a single national top level Root Certificate Authority, or it may be inside the one country where independent Root Certification Authorities exist, e.g. the USA. If they do not share a common trust point, they will have difficulty with secure dialogue. This section considers some of the recommendations which have been made to solve this problem.

In the following we use this notation, A_i refers to a subsidiary CA of the root CA A, and $A_{i,j}$ refers to the j-th end user at the level served by the subsidiary CA A_i.

A member $A_{j,k}$ at level j of a group with top level Root Certificate Authority A has available to it the certificates $C_{A,A}$, C_{A,A_1}, C_{A_1,A_2}, ..., C_{A_{j-1},A_j}, $C_{A_j,A_{j,k}}$. A is the RCA, and A_1, ... A_j are the subordinate issuers of the lower level certificates. End entity $A_{j,k}$ has a certificate $C_{A_j,A_{j,k}}$ issued to it by issuer A_j.

If a member of a community served by RCA B is to rely on a communication from a stranger outside the group, the public key of the outsider has to be known with confidence. There are several ways to gain assurance.

1. A trusted external entity states or implies that the external certificate is trustworthy, e.g. X approves $C_{A_j,A_{j,k}}$ of the sender in the other community in the above example.
2. A trusted external entity states or implies that a higher member of the certificate chain of which the stranger is part, is trustworthy, implying that the external certificate is trustworthy, e.g. X approves $C_{A,A}$ or a C_{A_{m-1},A_m} in the above chain. This is the most common situation, with a browser manufacturer acting as an external trusted reference X providing self-signed root certificates $C_{Z,Z}$ already accessible to the browser in the installation package. These are being accepted without question by the public:
3. A trusted member of the community indicates that the stranger is trustworthy, e.g. the root of the community or an issuer higher than the receiver of the message.
4. A trusted member of the community states or implies that a higher member of the certificate chain of which the stranger is part, is trustworthy, implying that the external certificate is trustworthy. In this case, the target could be $C_{A,A}$ or any of the descendant certificates. The most efficient way would be to refer to the root certificate, then all of the descendants are also included.

In the following subsections, we discuss specific strategies which fit the above grouping.

4.2 Shared Root

A shared root at the top of a hierarchy is the most obvious and desirable arrangement and has been tried numerous times, e.g. in Privacy Enhanced Mail of [13]. The 1998 Australian Gatekeeper strategy [17] assumed that a Policy and Root Registration Authority would be established in Australia “as a part of the national electronic authentication infrastructure”, but it has failed to appear.

More recently, some private companies have promoted themselves directly or indirectly as a global root [22] [23]. However, strong arguments have been advanced against a single root, e.g. a group within the US government reasons that a single root even within the US government for inter-agency use is impractical [2].

If a global root were implemented, it could be useful for individuals, as are national passports, but it does not solve the problems of the lengths of the certificate chains or the difficulties associated with the processing of the chains.

After years of discussion, it seems the global root concept will never become a reality. Another proposal which has received less opposition is for a system of CAs linked by cross certification, which we will now examine.

4.3 Cross Certification

According to IETF PKIX Working Group, a *cross certificate* is a certificate issued by one CA to another CA which contains a CA signature key used for issuing certificates. This definition can be applied to either authorities of the same PKI or to root authorities belonging to different PKIs. In this section, we mean cross certification for the second case.

It has been recommended that root Certificate Authorities of independent domains issue certificates to one another [1] [8]. This “cross certification” is a bi-directional example of the fourth assurance method in the earlier subsection (4.1), carried out by the root Certificate Authorities of pairs of certificate trees (although RFC 2510 [1] allows for a uni-directional process only, recognizing that achieving bi-directional processes has proven to be very difficult in practice). The root Certificate Authority A of certificate chain $C_{A,A_1}.C_{A_1,A_2}...$ creates a certificate $C_{A,B}$ for the root Certificate Authority B of another community, and makes it available so it can be used by any A community members who receive communications from members of the B community. Certificate Authority B performs a similar process for its community, creating $C_{B,A}$.

Making a cross certificate available could be done by A providing $C_{A,B}$ to every member of its community directly, e.g. in email, so they will possess it in case a member of the B community wishes to contact them.

A second way is for A to send $C_{A,B}$ to all members of B’s community, in case they want to use it. This would be difficult in practice because their identities are probably not easily available to A. Alternatively, B could send $C_{A,B}$ to all members of the B community. The final way is for A to provide it indirectly, e.g. in a directory, so it will be accessible to all communities.

4.3.1 Performance. When a member of the B community receives a communication from a member $A_{j,k}$ of the A community, a set of certificates, C_{A,A_1}, C_{A_1,A_2}, ..., C_{A_{j-1},A_j}, $C_{A_j,A_{j,k}}$ might be received. This will apply if the public key of the RCA A is considered to be available, e.g. pre-installed in a browser or registry which came with a desktop system; however, for high security applications the key should be checked for revocation. Otherwise the root key will need to be supplied, e.g. in a root certificate $C_{A,A}$. This is the maximum sized chain another member of the A community might receive from $A_{j,k}$, and if they have a common issuer below the root, it could be noticeably shorter, even down to one certificate. For the member of the B community, the cross certification certificate $C_{B,A}$ from B is needed to form a verifiable chain. The chain length for the B member is thus one greater than the worst case needed when $A_{j,k}$ communicates with a member inside the A community. For internal messages, B might have no certificates if the network is trusted, or a minimum of one certificate otherwise, so the communication with $A_{j,k}$ may be considerably slower.

4.3.2 Burden on Certificate Authority. A participating CA has a number of problems to face. In most cases, the CA will easily verify the identity of the other CA, in the registration phase. However, what if the CA considers the other to be financially weak, as in the 2000-2001 dot com shakeout? Does it certify anyway, and risk finding later in court that some of its members incurred financial losses as a result? Some might argue that cross certifying a CA implies financial trustworthiness. A more serious and probable conflict is in regard to the Certification Policies of the CAs. Often, different grades of certificates are issued, depending on the intensity of the registration process. Should a CA requiring high reliability identification certify one which has a casual identification, e.g. email address?

The issuer needs to locate and reach agreement with other root CAs. After certification, the issuer needs to be vigilant and revoke the certificate if there are any problems with the other CA, or if its private key is changed.

Notifying those interested in cross communications when there is a change is another problem, which will vary with the legal environments in countries around the world.

Many other Certificate Authorities will be outside of the legal jurisdiction in which the CA operates. Should it attempt to cross certify them?

4.3.3 Other Practical Problems. Cross certification solution rapidly becomes less and less attractive as the number of root Certificate Authorities grows. For N CAs, there are $N(N-1)$ cross certificates. Frequently, each CA will have more than one root certificate. If each CA has M root certificates, the number grows to $MN((MN)-1)$. The "PKI Education" section of the "Resources Links" page of the PKI Forum's web site lists eleven CAs in the Americas, five in the Asia/Pacific, thirty in Europe and the Middle East (including nine European top level CAs), and another eight as global, a total of fifty four major public CAs. When the multiple root certificates held by most CAs are counted, the number of effective cross certifiers becomes even worse. In MicroSoft Internet Explorer version 6.0, there are 117 separate entries under "Trusted Root Certification Authorities" and another 12 under "Intermediate Certification Authorities", and this list does not include all of the CAs listed by the PKI Forum.

The process is not just a matter of issuing certificates. As mentioned in 2.3, there are technical complications resulting from certificate extensions, particularly ones marked as critical, as well as legal and management issues arising from differences between the Certification Policies of those involved, many of which cannot be resolved.

4.4 Certification Bridge

In practice, cross certification seems to have not been undertaken by any pair of major CAs, not even within a government [15].

A *bridge certificate authority* is an alternative to cross certification where another party which is not a CA, in the sense that it is not the peak of a certificate hierarchy, engages in cross certification with real CAs. A real CA does not attempt to cross certify with other CAs.

Because it is not a real CA running a CA business, it is not a direct competitor to commercial CAs, and may encounter less commercial resistance. However, it could be viewed by some CAs as supporting competitive CAs and they might choose to not be involved.

4.4.1 Performance and Practical Problems. Certificate chain lengths are one more than for the cross certification case. For recipients in PKI group A to be able to verify certificates in PKI group B, there needs to be a certificate $C_{A,Bridge}$ and another $C_{Bridge,B}$. For reverse direction validation, there need to be certificates $C_{B,Bridge}$ and $C_{Bridge,A}$. For cross certification, there needed to be only $C_{A,B}$ and $C_{B,A}$.

The burden on a Certificate Authority is considerably reduced, to a single cross certification, and does not grow with the number of participants in the system, as in the cross certification scenario. The burden is shifted onto the bridge CA.

However, although the major impediment for the deployment of a BCA based PKI architectures is political, there are also some technical drawbacks. Firstly, a compromise of Bridge Private key will cause all inter-PKI communications to be discontinued and will be more disruptive than a compromise for the cross certification network.

Secondly, efficient discovery and validation of certification paths and the interoperability of large PKI directories are not easy to solve within this scenario [19]. It is clear that certification path discovery and validation is basically harder in mesh PKIs than in hierarchical ones because there are multiple trust points within the PKI, and also because a possibility of non-termination trust cycles exists. A BCA based PKI architecture includes some mesh PKI segments within its overall structure. Thus, all PKI users must be able to develop and validate complex certification paths. Additionally, the BCA must use certificate information to constrain trust relationships between different enterprise PKIs. Therefore, certificates become more complex, and the PKI users must be able to process and use this additional trust information during the validation of certification paths.

Finally, another problem of BCA based PKIs is the distribution of certificates and certificate status information in a way that is useful to users and their applications. Users must be able to easily obtain CA and user certificates and the corresponding certificate status information from a distribution mechanism. Early PKI designers expected a global X.500 directory to emerge and solve this problem. However, the global X.500 directory did not emerge. PKIs are being deployed using Lightweight Directory Access Protocol (LDAP) directories, web servers, and ftp servers to distribute certificates and certificate status information. This motivates that obtaining PKI information becomes a difficult problem when connecting established PKIs because they most likely will have used different certificate and certificate status information distribution mechanisms.

4.4.2 Example: Federal Bridge Certificate Authority. US government has tried and has opted for a bridge [2], the Federal Bridge Certification Authority (FBCA), which is discussed in more detail later in 4.4.2. The authors of the FBCA report claim that "When considering the number of Federal Agencies planning to set up PKIs, a mesh would be impossible to maintain" because of the number of cross certificates.

The FBCA was started in the United States of America on 7 June 2001, consisting of twenty five members, thirteen of whom were from government organizations. The FBCA operations are outsourced to five Certificate Authorities.

The services provided by the FBCA are policy mapping, cross certification, and interoperability. An online directory holds entries which cross certify the FBCA and each of the participating Certificate Authorities. The entries are created offline.

For consideration for admission to the FBCA program, a CA must issue certificates which conform to a nine page specification, including a six page certificate profile and a two page CRL profile.

The initial demonstration [7] of interoperability in April 2000 involved two commercial CAs and is reported [15] to have needed two weeks of hand crafting of certificates by NIST personnel to achieve a working system.

4.5 Cross Recognition

The term "cross recognition" is used by some as a less desirable but more likely alternative to cross certification. Sometimes it is used in the legal sense, that a digital certificate or signature which is legal in a particular jurisdiction is accepted as legal in another, but here we refer to the usage that a root certificate is recognized by the possessor as being genuine, without the assurance of a cross certificate or a bridge certificate. The term is used in some GateKeeper documents [18].

An everyday example is in the common web browser where the user accepts the certificates which have been supplied with the browser software. The user "recognizes" the root certificate of the issuer, in this example without any evidence other than the assurance that the browser distributor supplied it.

4.6 Validation Authority

To remove the burden of validating a chain of certificates from a user, it has been proposed that a *Validation Authority* (VA) could perform the process as an outsourced service. A VA might build and maintain a database of certificates and revocation information, or it might rely on access to directory information and online revocation information servers, e.g. *Online Certificate Status Protocol* (OCSP) servers [16]. A VA could be internal to an enterprise, particularly if the enterprise is large, or it might be a public concern, probably offering a service for a fee. In either case, it still has to carry out the checking of all entries in the chain, and even more so if it is charging for its service. A performance improvement may be perceived if the VA processing is on a lightly loaded large machine rather than on a heavily loaded workgroup machine. A VA becomes attractive when the certificate chain is likely to contain numerous complex certificates which can be difficult to handle automatically, so the end user relies on the VA to have perfect software rather than risk flaws in local processing.

A good example is ETRI's VA system [11]. It will be based on distributed VAs which are strategically located and collect information from CAs through Certificate Revocation Lists, OCSP responders, and *Simple Certificate Validation Protocol* (SCVP) [14] servers.

A VA works fine when there are no attackers or security incidents, i.e. when the VA is not needed. The complexity of extensions in X.509 version 3 and later combined with serious risks associated with the common insecure desktop systems is such that a skilled attacker is likely to be able to thwart a VA or its use, resulting in financial loss to someone. The likelihood of legal action when an error occurs will probably discourage the establishment of public VAs in many countries.

Even though the VA has removed the difficulties of processing from the end user, there is still the problem of secure communications between the VA and the end user. A man-in-the-middle attack could be very profitable for an impersonator of a VA.

For applications within an enterprise, the use of a Validation Authority reduces the loading on the receiver by shifting it elsewhere but does not reduce the total work required within the enterprise.

5 Improving Performance of Inter-community Communications

Options analyzed in the previous section still involve processing of a chain of certificates, either at the receiving computer or by an outsourced entity. A shared root potentially enables the shortest chain. Cross recognition involves no additional certificates in a chain, while cross certification and bridge certification add two or one more certificates to the chain. For high volume repetitive transactions, they provide no relief from the costs of repeated processing of the same chains.

For individuals around the world, it is desirable for anyone to be able to contact anyone else anywhere, but for enterprises this is more general than is usually required, particularly in a global PKI. For global trading, an enterprise is more likely to have a core group of business partners, suppliers and customers, with which it has volumes of messages. The entities in businesses which are involved in the secure communications will likely be at or near the bottom of a certificate chain and these will be in relatively large numbers. The security checks on these messages or transactions will involve repeated processing of long certificate chains, from the top down to the bottom levels, effectively the worst case. Communications between higher level executives may involve shorter chains but these will be much fewer in number.

When a communication is received from a stranger for the first time, the chain has to be processed to verify the association between the end entity identified in the last certificate and the related public key, but a complete reverification for subsequent messages is not necessary, as shown below, although some time variable items will need attention.

Let's consider the scenario of two workgroups in different PKI communities. They may have a common top level root, or may be in different trees, perhaps linked by cross certification or a bridge. Communications between the bottom entities in the workgroups represents the worst case for chain length within this or any similar scenario.

To improve the process of frequent verification of a chain, we introduce two new concepts, a *virtual digital certificate* and a *synthetic certificate*, as described below. We then continue by showing how these may be applied in practice.

5.1 Virtual Digital Certificates

5.1.1 Introduction. Consider when an entity X in one community receives a message supposedly from Y in another community. A chain of certificates may be received indicating that Y has a certain public key $PublicKey_Y$. Entity X wants to determine if that is really the public key of Y, and if the digital signature of the message indicates its source is Y. If X's community has not encountered Y before, the certificate chain will have to be processed by X or another entity in X's community or by their agent, or X might ask a trusted reference for the public key of Y. For subsequent messages from Y to X, the first method is time and resource consuming, and we will pursue the option of asking a trusted reference.

In theory, it is not necessary to have a long chain of certificates from a root CA to the end entity Y, the root CA could issue a certificate directly to Y, and then the X community would have a simple task of verification. Of course, this ideal single certificate is not scalable or practical, but it would speed up processing by the receiver if it did.

In this article, we propose an entity in the X community which acts as if such an ideal certificate for Y did exist and it has possession of it, thereby being able to divulge the public key of Y very efficiently to X and to others who enquire. Because this single level certificate does not really exist, we call it a "virtual certificate" [21].

Next we provide a more formal specification and details of components and validation, followed by the related concept of a "synthetic certificate".

5.1.2 Specification.

Definition 1 *A "virtual digital certificate" (or "virtual certificate") is a data set which is derived from a chain of certificates, containing the information which would be in a digital certificate issued by the first CA in the chain to the end entity if one had been issued, but such a certificate has not been issued.*

Definition 2 *For a chain of certificates, which are compatible in policies and in extensions, beginning at entity E_1 and extending to entity E_N, a virtual certificate VC_{E_1,E_N} exists at a time T if and only if there exist valid certificates C_{E_1,E_2}, C_{E_2,E_3}, ..., C_{E_{N-1},E_N}, where $C_{X,Y}$ indicates that a digital certificate has been issued by entity X attesting to the public key of entity Y and is valid at time T.*

Notes:

- The data set is used by the entity which created it, the *Virtual Certificate Manager* (VCM), and by other entities which trust the work of the VCM.
- The data set is not expected to be made available to other parties, but retained by the VCM, which acts as an authority on the value and status of the

public keys of the end entities for which it has built virtual certificates. Other parties who want the data set information should instead seek the corresponding "synthetic certificate", that is presented and discussed below.
- The contents of a VC are probably not released by the VCM.
- The formation of the VC may require complex processing to determine the compatibility of policies and extensions.
- From definition 2, a virtual certificate ceases to exist if any of its component certificates expires or is revoked. This is very important because it greatly simplifies validations after the first, avoiding the complex rechecking of policies, paths, etc., as discussed in 5.1.4.

5.1.3 Components. Components of a virtual digital certificate include standard items:

- Issuer
  optional, the issuer of the first certificate in the chain. Might not used in practice.
- The commencement validity date
  is the latest of the commencement validity dates of the component certificates.

  Definition 3 $CVD_{VC} = latest(CVD_{C_m})$ *for all* m *certificates* C_m.

  with CVD_{C_i} being the commencement validity date of the i-th certificate.

- The expiry validity date
  is the earliest of the expiry validity dates of the component certificates.

  Definition 4 $EVD_{VC} = earliest(EVD_{C_m})$ *for all* m *certificates* C_m.

- The Subject
  is the identity of the final entity in the base certificate chain.
- The public key information
  (PK) is the public key information in the final certificate of the chain, designating the algorithm concerned and the actual public key.

  Definition 5 $PK_{VC} = PK_{C_{i,j}}$

The X.509 [8] items "version", "serialNumber", "signature", have no meaning or practical use here. The items "issuerUniqueID", and "subjectUniqueID" seem to be rarely used and are not essential to the principles given here.

Another new component, the *component certificate revocation list*, will be added, as it is described below.

5.1.4 Validation

Definition 6 *Validation at a particular time means the process of determining if a certificate is intended by its signer to be current at that time and has not been revoked, or, in the case where a chain of certificates is involved, whether every one of the certificates in the chain is intended to be current at that time and has not been revoked.*

Notes:

- For purposes of validation of a chain of real certificates, it suffices to determine the existence of the virtual corresponding certificate.
- If a virtual certificate exists, the corresponding chain of real certificates is valid.
- If a virtual certificate is known to exist at a time T, then its existence at time $T + dt$, provided $T + dt$ is earlier than its expiry date, is true if and only if every one of its components has not been revoked.

In practice, the first noted observation should reduce the re-validation of a certificate path to a series of revocation checks, without requiring a repetition of the N hash and digital signature calculations. Therefore to speed up the revocation checks, the Virtual Certificate will need a new item, a *Component Certificate Revocation List* (CCRL), a component certificate list which identifies all of the components from which it was formed, so that each can be checked if required for revocation. For each component certificate of the virtual certificate, the Issuer and the unique identity assigned by the Issuer to the component certificate, along with the status/revocation check point (or, for CRLs, an issuing distribution point) if available, should be adequate for the purpose.

Definition 7 Component Certificate Revocation List:
$CCRL_{VC} = set(Issuer_m, CertID_m,$
$RevPointType_m, RevPoint_m)$
for all m *certificates* C_m.

Revocation checks could be carried out by a message receiver but we expect that the Virtual Certificate Manager would provide a service whereby the receiver would make a revocation enquiry of the Virtual Certificate Manager and the Virtual Certificate Manager would run the revocation checks and report the result, storing it for re-use over the short term. Most users are not interested in the content of the Virtual Certificate, only whether the claimed public key which they have received from the sender can be trusted. As such, it is better to adhere to the information hiding principle, and have repetitious functions performed by the Virtual Certificate Manager.

Revocation checks can be conducted securely without using public key calculations under certain circumstances [20] which could apply to this case.

5.2 Synthetic Certificates

5.2.1 Introduction. A VC for entity Y is managed by a VCM and would probably not be circulated to other entities. Other processing entities which trust the VCM may wish to have a single level certificate for the entity Y, so the VCM, which is convinced of the public key of Y, could issue a certificate for Y, $C_{VCM,Y}$, which could then be used by the other processing entities. It would contain a reference to the VCM as the point to which enquiries for revocation should be directed. We use the term *synthetic certificate* or *synthesized certificate* to describe this certificate issued by a VCM which has no direct relationship with the party whose public key is being certified. It has most of the content of a virtual certificate and is signed by an entity, the Synthetic Certificate Manager (SCM). A difference from the virtual certificate is that the synthetic certificate normally has no revocation list, only the revocation contact point for the issuer. However, a revocation list could be provided in an extension field, but this shifts some of the repeated processing to the receiver, which we are trying to avoid. Revocation checks are conducted by the Synthetic Certificate Manager.

A synthetic certificate may be available even though the corresponding Virtual Certificate is not. Even though a Synthetic Certificate and a Virtual Certificate are theoretically related, in practice one could be in use without the other, but we envisage that a Virtual Certificate Manager would have available the Synthetic Certificate, and a Synthetic Certificate Manager would make available the Virtual Certificate.

5.2.2 Specification

Definition 8 *A* Synthetic Certificate *(SC) is a digital certificate constructed by an entity which is trusted by some other parties, stating the link between an entity to which it is not directly related and its public key, the content having been derived from other sources.*

Definition 9 *The* Synthetic Certificate Manager *(SCM) is the entity which constructs the synthetic certificate and provides revocation status information to enquirers.*

Notes:

- A synthetic certificate could be easily created by a VCM from a VC.
- The SCM normally would not be involved in the certificate chain, $C_{E_1,E_2}, C_{E_2,E_3}, ..., C_{E_{N-1},E_N}$ from which the virtual certificate, VC_{SCM,E_N}, was derived.

5.2.3 Components. Components of a synthetic digital certificate include standard items:

- Version, at the discretion of the Virtual Certificate Manager.
- SerialNumber, a unique identifier at the discretion of the Virtual Certificate Manager.
- Issuer, now the Virtual Certificate Manager.
- Signature (algorithm) used by the Virtual Certificate Manager.
- The commencement validity date, as in the Virtual Certificate, but probably unnecessary in practice if checks are carried out through the Virtual Certificate Manager.
- The expiry validity date, as in the Virtual Certificate.
- The Subject, as in the Virtual Certificate.
- The public key information, as in the Virtual Certificate.
- A revocation check point, e.g. a server process ID or port.
- The type of revocation check point, e.g. OCSP [16], SCVP [14], etc.

The X.509 items *issuerUniqueID*, and *subjectUniqueID* can be omitted here.

5.2.4 Validation. Note that a synthetic certificate is valid only if the corresponding virtual certificate is valid.

The entity certifying the public key of the end entity of the certificate chain has synthesized a single certificate from the information contained in the public chain after validating each and every one of the components in the chain. Thus the user of a synthetic certificate need not repeat the expensive validation checks already run by the issuer of the synthetic certificate.

If a synthetic certificate is known to be valid at time T, then it is valid before the expiry date at time $T+dt$ if and only if the virtual certificate has not been revoked.

Because of the above observation, an entity which has accepted the synthetic certificate at one time, and seeking to revalidate it at a time prior to expiry, need only check with the signer, i.e. the VCM, for revocation.

5.3 Uses for Virtual Certificates and Synthetic Certificates

A virtual certificate or a synthetic certificate is of use to members of a group who require frequent validation of the public key information associated with an entity, particularly when the information is fixed for long durations.

Consider the case where an entity X in an enterprise is frequently communicating with an entity Y in another community, and the certificate chain presented by the other entity is $C_{A_1,A_2}C_{A_2,A_3}...C_{A_N,Y}$. The entity X has to be assured that each and every certificate in the chain is correctly signed, has not expired and has not been revoked. There may be further complexities involved, such as policy information which specify the policy followed by an issuer and the purposes for which the key pair associated with the certificate are authorized, but if these have validated once (during the creation of the virtual certificate) and there is no change to the composition of the chain, then repeated checking of them is unnecessary.

A virtual certificate may be used by those who trust its manager and who have secure communications channels, and the synthetic certificate is for others who require a digital signature on the content. In both cases, the bulk of the repitious validation of a certificate chain is moved to another entity, which also performs efficient continual revalidation on request triggered by requests from the receivers, using only revocation checks.

5.3.1 In-house Applications. For in-house operations at one site, the computer network is generally considered low risk, and there is a predefined relationship amongst the nodes. Often little need is seen for public key security for internal communications. For applications in this environment, a workgroup or enterprise server might construct a virtual certificate for use by an end user and store it for later use. There is relatively little benefit to be gained from having the enterprise server conduct the first processing, perhaps a short time due to processing on a faster machine. The benefits grow from the repeated requirement to process the same chain, in which case the contents of the corresponding Virtual Certificate expedite the decision.

For the first processing of the chain, the Virtual Certificate is constructed, stating that the public key of remote entity X is $PK.Val$, and some auxiliary information to be used later if required. Later, when a chain with an end target identified as X is received, the current set of Virtual Certificates is consulted and those containing X are examined for one with a public key of $PK.Val$. If a suitable unexpired Virtual Certificate is located, it is sufficient to validate it in accordance with the method outlined above. Probably in most cases, entity X will have only one certificate chain and therefore only one active Virtual Certificate, so the later validation of the public key of X will be quite fast, particularly if the relationship between the validator and the revocation reference sites allows hash based communications [20].

More explicitly, on the first occasion, the operations are

- validate the end entity using conventional methods, involving checking of hash values, digital signatures, policy information, and any extensions.
- create the virtual certificate with its contents of Issuer, commencement validity date, expiry validity date, subject identity, public key information, revocation information list, and anything else deemed to be necessary for local validation later.

On later occasions, given an identity, a presumed public key, and perhaps the identity at the top of the certificate chain,

- find a virtual certificate with a matching subject identity
- compare the offered public key with that in the Virtual Certificate
- if there is no match, look for another Virtual Certificate.
- Complete the validation by ensuring that no revocations of components have occurred

Because there is no need to reprocess the whole chain, later verifications of the public key should be relatively fast.

5.3.2 Public Applications. In the above example, an internal network of satisfactory security was assumed, and the users of the Virtual Certificate were internal entities. Where the receiver of a frequent certificate chain can have prior registration with the Virtual Certificate Manager, hashing of communications can provide efficient secure communications, and a Virtual Certificate can be used by entities outside of the Virtual Certificate Manager's organization, i.e. the public. For many situations, prior registration will not be feasible, and other means need to be used for secure communications. In most cases, providing the machines involved are satisfactorily secure, the use of a network link secured by SSL/TLS [6] and signed data structures, e.g. the synthetic certificate, are probably adequate.

5.3.3 Abolition of Certificate Chain after Initial Contact. Initially the VCM uses the certificate chain from entity Y to verify the identity of the communicant and to build the VC for Y. Thereafter, when a member X of the VCM's community receives a message from Y, there is no need for Y to send the chain again. This is of benefit where the communication bandwidth between X and Y is limited, as in wireless applications. For verification of a message allegedly from Y, X enquires of the VCM for the public key of Y. If Y's public key remains the same, the enquiry will be processed quickly by the VCM. If there has been a change in the public key of Y, the VCM will still return the same key (unless there has been a revocation somewhere in the chain) but the signature check will fail, and the VCM will be called upon to re-initialise the VC for Y or issue an additional one, depending on the situation.

5.3.4 Wireless Networks and C Limited Devices. These devices suffer from resource limitations because of their size and weight, factors which are determined by the public rather than by technical considerations. Because of their available resources they would have difficulties in processing a chain of certificates. A VCM would relieve the device of processing of a chain, and even processing of a single certificate, and would act as a reference which could be consulted to determine if a received public key is still valid.

5.3.5 Example System. Consider the imaginary case of the Export section of the First Fisheries Bank (FFB) which has a large customer who has just signed a two year contract with the Specialty Fish Cuisine Corporation on the other side of the world. Specialty uses the New World Finance Bank (NWFB) for its imports. Although each bank has its own secure network, there is no direct link between them and they choose to utilize for communications the Internet for its worldwide reach and low cost, but employ public key cryptography to provide confidentiality, integrity, authentication, and non-repudiation to a legally satisfactory level. Each bank issues its own certificates internally from an enterprise CA which has obtained its credentials from a national CA.

There are daily shipments, the contents of which depend on the season and catch. Transactions are accounted daily and funds need to be remitted immediately to cover operating costs. Some messages concern details of a shipment and the itemised costing, others relate to payment details.

Because of the security requirements and audit recording needs, each message between the employees in each bank is accompanied by the corresponding chain of certificates. Each bank is keen to provide fast responses to its valuable staff, whose time is charged to the customer, and is aware of its need to minimise billing to customers in the competitive climate.

Each bank decides to establish a Virtual Certificate Manager to avoid the repetitive processing of certificate chains and to provide faster confirmation to its staff regarding incoming messages. In each case, a half dozen virtual certificates is adequate for daily operations. When transactions with banks for other customers are added, the VCM maintains several thousand virtual certificates at any time, and responds to over ten thousand public key enquiries per day.

5.4 Virtual Certificate and Synthetic Certificate Directories

For mobile telephones, the Service Provider is a trusted party, and the telephone network is relatively secure compared with the Internet, with the exception of the air link to the closest Base Station.

The Service Provider or Network Operator could establish a VCM as a value added service for a subscriber. One option would be to maintain VCs for senders specified by the subscriber, e.g. business associates and staff. Another is to maintain a general directory, a *Virtual Certificate Directory* which could be queried on demand, and could contain public key information of frequently requested entities, as revealed by statistics.

As for synthetic certificates, they are stored in a *Synthetic Certificate Directory.* It is similar to the above Virtual Certificate Directory in that information on a public key is available without processing a chain of certificates, but here it is not essential to contact or work with the VCM unless/until revocation checking is required.

6 Conclusions

Certificate chains are built by the recursive application of the certificate paradigm. This allows, based on transitive assumptions, to obtain the public keys of a number of CAs and, what is probably more important, the public keys of a number of final entities. This is desirable for anyone to be able to contact anyone else anywhere, but for enterprises this is more general than is usually required, particularly in the cases of interoperation of PKIs or in the case of a global PKI. The entities in businesses which are involved in the secure communications will likely be at or near the bottom of a certificate chain and these will be in relatively large numbers. The security checks on these messages or transactions will involve repeated processing of long certificate chains, from the top down to the bottom levels, effectively the worst case. The repeated validation of a chain of certificates can be time consuming and expensive over a period of time when done without memory of previous validations.

In this paper, we have analyzed available solutions for PKIs interoperation. We have shown that they still involve processing of long chains of certificates, either at the receiving computer or by an outsourced entity. Thus, a shared root potentially enables the shortest chain. Cross recognition involves no additional certificates in a chain, while cross certification and bridge certification add two or one more certificates to the chain. For high volume repetitive transactions, they provide no relief from the costs of repeated processing of the same chains.

We have presented new concepts of virtual certificate and synthetic certificate for faster and less costly processing of certificate chains, and showed how these types of certificates can be applied in practice. When a communication is received from a stranger for the first time, the chain has to be processed to verify the association between the end entity identified in the last certificate and the related public key, but a complete reverification for subsequent messages is not necessary, although some time variable items will need attention. Converting a chain into a virtual certificate improves validation within an enterprise. Converting a chain into a synthetic certificate extends the usefulness to clients in other communities who trust the Synthetic Certificate Manager. In this way, communications in a certificate-based intercommunity can be highly improved.

Further research is underway into techniques of generating and using virtual and synthetic certificates and appropriate directories, for both private and public environments.

7 Acknowledgment

This work is part of the on-going research activities undertaken on the research project *Global Public Key Infrastructure*, that is supported by the Telecommunications Advancement Organization of Japan.

References

1. C. Adams and S. Farrell. *RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols*, March 1999.
2. Peter Alterman. The U.S. Federal PKI and the Federal Bridge Certification Authority , 7 May 2001.
3. Find an author for this. Find a reference to this. *Unknown*, 0(0), May 1999.
4. D. Balenson. *RFC 1423: Privacy Enhancement for Internet Electronic Mail: Part III: Algorithms, Modes, and Identifiers*. Trusted Information Systems, Glenwood, Maryland, USA., February 1993.
5. CCITT. *The Directory - Authentication Framework*. Number CCITT X.509. International Telegraph and Telephone Consultative Committee, Switzerland, November 1988.
6. T. Dierks and C. Allen. *RFC 2246: The TLS Protocol Version 1.0*, January 1999. RFC 2246.
7. FPKI FBCA. Report of Federal Bridge Certification Authority Initiative and Demonstration (Electronic Messaging Association Challenge 2000), 2000.
8. ITU-T. Recommendation X.509: Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks, March 2000. ”Version 3, http://www.itu.int/rec/-recommendation.asp?type=folders&lang=e&parent=T-REC-X.509”.
9. Burt Kaliski. *RFC 1424: Privacy Enhancement for Internet Electronic Mail: Part IV: Key Certification and Related Services*. RSA Laboratories, February 1993.
10. Stephen T. Kent. *RFC 1422: Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management*. BBN, February 1993.
11. Heesun Kim, Yeongsub Cho, Seunghun Jin, and Kyoil Chung. Current Status and Trends of PKI in Korea. In Kim [12], pages 1 – 21.
12. Kwangjo Kim, editor. *Proceedings of First International Workshop for Asian PKI*, ICU, Daejeon, Korea, 19–20 October 2001. International Research Center for Information Security, Korea and Institute of Industrial Science, Japan.
13. J. Linn. *RFC 1421. Privacy Enhancement for Internet Electronic Mail: Part I: Message Encryption and Authentication Procedures*, February 1993.
14. Ambarish Malpani, Russ Housley, and Trevor Freeman. *Simple Certificate Validation Protocol (SCVP)*, March 2002. draft-ietf-pkix-scvp-08.txt.
15. Robert Moskowitz. PKI at a Crossroads. *Networkcomputing.com*, 1 May 2001. http://www.-networkcomputing.com/1108/1108colmoskowitz.html.
16. M Myeres, R Ankney, A Malpani, S Galperin, and C Adams. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, June 1999.
17. Office of Government Information Technology. Gatekeeper, a strategy for public key technology use in the Government. online, 6 May 1998. http://www.govonline.gov.au/publications/-GatekeeperStrategy.pdf.
18. Office of Government Information Technology. Gatekeeper Accreditation Certificate. online, 2000. http://-www.govonline.gov.au/projects/publickey/gac.htm.
19. WT Polk and NE Hastings. Bridging Certification Authorities: Connecting B2B Public Key Infrastructures. Role of Federal Bridge CA, September 2000.
20. Selwyn Russell. Fast Checking of Individual Certificate Revocation on Small Systems. In Jeremy Epstein, editor, *Fifteenth Annual Computer Security Applications Conference*, Radisson Resort Scottsdale, Phoenix, Arizona, 6–10 December 1999. Annual Computer Security Applications Conference. http://www.acsac.org/1999/-papers/thu-a-1300-russell.pdf.
21. Selwyn Russell, Ed Dawson, Eiji Okamoto, Javier Lopez. Improving Performance in Global PKI using Virtual Certificates. In *2002 Symposium on Cryptography and Information Security*, Shirahama, January 2002, pp. 149-154
22. Thawte. The Cross-Certification and Chained Certificate Authority Program. Accessed 5 Nov 2001. http://-www.thawte.com/certs/chained/whitepaper.html.
23. WiseKey. The World Internet Security Company. Accessed 5 Nov 2001. http://www.wisekey.com/wisekey/-commonroot.htm.