DOI: 10.15415/jtmge.2010.11005

Journal of Technology Management
for Growing Economies

Volume 1 Number 1 **April 2010**

Certificate Path Verification in Hierarchical and Peer-to-Peer Public Key Infrastructures

Balachandra
Prema K.V.
Manipal Institute of Technology, Manipal, India

CHITKARA
UNIVERSITY

Certificate Path Verification in Hierarchical and Peer-to-Peer Public Key Infrastructures

Balachandra
Prema K.V.
Manipal Institute of Technology, Manipal, India.

Abstract

Authentication of users in an automated business transaction is commonly realized by means of a Public Key Infrastructure(PKI). A PKI is a framework on which the security services are built. Each user or end entity is given a digitally signed data structure called digital certificate. In Hierarchical PKI, certificate path is unidirectional, so certificate path development and validation is simple and straight forward. Peer-to-Peer(also called Mesh PKI) architecture is one of the most popular PKI trust models that is widely used in automated business transactions, but certificate path verification is very complex since there are multiple paths between users and the certification path is bidirectional. In this paper, we demonstrate the advantage of certificate path verification in Hierarchical PKI based on forward path construction method over reverse path construction method with respect to the time requirement. We also propose a novel method to convert a peer-to-peer PKI to a Depth First Search(DFS) spanning tree to simplify the certificate path verification by avoiding multiple paths between users, since the DFS spanning tree equivalent of peer-to-peer PKI contains only one path between any two Certification Authorities.

Keywords: PKI, Hierarchical PKI, Peer-to-Peer PKI, Certification Authority, Certificate verification, OpenSSL.

INTRODUCTION

During automated business transactions, one of the nontrivial security services to be established by a security system is the trust between the participating users. This is also called authentication of users. Trust can be realized by means of a *Public Key Infrastructure(PKI)*. The *Public Key Infrastructure (PKI)* is a set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates(Mazaher and Roe, 2003). Digital certificates have become an accepted method for securely binding the identity of an individual or a device to a public key, in order to support public key cryptographic operations such as digital signature verification and public key-based encryption. Digital signatures play an essential role for security on the Internet.

Journal of Technology
Management for
Growing Economies
Vol. 1 No. 1, April 2010
pp.77-92

PUBLIC KEY CRYPTOGRAPHY

Public key cryptography supports security mechanisms such as confidentiality, integrity, authentication, and non-repudiation. In order to successfully implement these security mechanisms, one must carefully plan an infrastructure to manage them. A Public Key Infrastructure is a foundation on which other applications, system, and network security components are built. Interoperability is a major issue in Public Key Infrastructures. There are three families of public key cryptography in common use today(Zuccherato, 2003). Firstly, the systems based on integer factorization. RSA cryptographic system is the most popular public key cryptographic system under this family. Secondly the systems based on the discrete logarithm problem(Thales, 2000). These algorithms can provide support for both digital signatures(Cronin et al., 2003) (DSA) and key agreement(Diffie-Hellman). Thirdly, the system based on arithmetic using elliptic curves (Thales, 2000). Elliptic Curve Cryptography is a relatively new family of public key algorithms under this family that can provide shorter key lengths and depending on the environment and application in which it is used it can provide improved performance over systems based on integer factorization and discrete logarithms. The de-facto cryptographic algorithm for digital signatures and encryption of symmetric keys is the RSA. Although RSA is widely used and provides high security, there are some potential problems with its use. In DSA, signature generation is faster than signature verification, whereas with the RSA algorithm, signature verification is very much faster than signature generation.

Digital Signature Schemes

Digital Signature schemes sign messages and verify the resulting signature with two different keys in such a way that it is difficult to sign without the signing key (Kaliski, 1993). Similar to public key cryptosystems, the verification key can be published without compromising security, and is called the public key; the signing key is called the private key.

Digital signature schemes provide integrity and origin authentication. Like public key cryptosystems, they do not require that parties first agree on a secret key, and they are generally somewhat slower than, for instance, secret-key cryptosystems and cryptographic hash functions.

RSA Digital Signatures

The RSA is based on the hard mathematical problem of integer factorization, i.e., given a number that is the product of large prime numbers, factorize the numbers to find the primes. RSA Digital signatures are generated by performing the encryption of some clear text using one’s own private key(Weise, 2001). This encryption allows one entity to send a message to many other entities that may then authenticate that message, without the need to first exchange secret or private cryptographic keys. The recipient simply decrypts the message with the originator's public key.

DSA Digital signatures

The DSA is based on the difficulty of computing discrete logarithms and is based on schemes originally presented by ElGamal and Schnorr. There are three parameters that are public and can be common to a group of users. A 160-bit prime number q is chosen. Next, a prime number p is selected with a length between 512 and 1024 bits such that q divides (p-1). Finally, g is chosen to be of the form $h^{(p-1)/q}$ mod p, where h is an integer between 1 and (p-1) with the restriction that g must be greater than 1. With these numbers in hand, each user selects a private key and generates a public key. The private key x must be a number from 1 to (q-1) and should be chosen randomly or pseudorandomly. The public key is calculated from the private key as $y=g^x$mod p. The calculation of y given x is relatively straight forward. However, given the public key y, it is believed to be computationally infeasible to determine x, which is the discrete logarithm of y to the base g, mod p.

Public Key Infrastructures

Public Key Infrastructures have been considered as an appropriate framework for the provision of security services such as data confidentiality, integrity, availability, authentication and non-repudiation in several business sectors. A PKI provides a foundation for other security services. The primary function of a PKI is to allow the distribution and use of public keys and certificates with security and integrity. In cryptography, a PKI is an arrangement that binds public keys with respective user identities by means of a Certificate Authority (CA).

Different business corporations deploy different types of PKIs such as Single CA, Hierarchical, Bridge, Hybrid, and Mesh PKI(Adams and Farrell, 1999; Adams and Lloyd, 2003). In Hierarchical PKI, certificate path is unidirectional, so certificate path development and validation is simple and straight forward. However, if the root CA is compromised, which is everyone's trust point, the security of the whole system is collapsed. Mesh architecture (also called Peer-to-Peer PKI or web of trust) is also widely used in applications such as MANET(Serranoa et al., 2007), but certificate path development is more complex than in a hierarchy. Unlike a hierarchy, building a certificate path from a user's certificate to a trust point is nondeterministic. The Bridge Certification Authority (BCA) architecture was designed to address the shortcomings of the above two basic PKI architectures, and to link PKIs that implement different architectures, but certificate path discovery is not simplified.

Since Hybrid PKI is the mixture of different PKI architectures(Pez et al., 2006), the complexity of certificate path verification is increased. The purpose of this paper is to show the benefit of certificate path verification using forward path construction method over reverse path construction method in Hierarchical PKIs. We also propose a novel method to convert a Peer-to-Peer PKI to a Depth First Search(DFS) spanning tree to simplify the certificate path verification by avoiding multiple paths between users. The rest of the paper is organized as follows:

In the paper, we introduce different types of PKI structures. Further the certificate path verification in Hierarchical PKIs based on forward and reverse path construction techniques is explained. Then the experimental results that show the benefit of certificate path verification in Hierarchical PKIs using forward path construction method over reverse path construction method in terms of path verification time is shown. After this the proposed method of converting a peer-to-peer PKI to its equivalent DFS spanning tree and its advantages is explained. This is followed by the conclusions.

PKI STRUCTURES

The organization of PKI components in a PKI environment is called a PKI structure or a PKI architecture or a PKI trust model. Here we emphasize only two PKI components, viz. Certificate Authorities and users or end entities. Different business corporations deploy different types of PKI trust models such as Single CA, Hierarchical, Bridge, Hybrid, and Mesh or Peer-to-Peer PKI.

Single CA PKI model

As shown in Figure 1, A Single CA PKI architecture is one that contains a single CA and provides the PKI services for all the users or the end entities (ENT1, ENT2, ENT3 and ENT4 in Figure 1) of the PKI. There is a single trust anchor that has to be trusted by all the users.

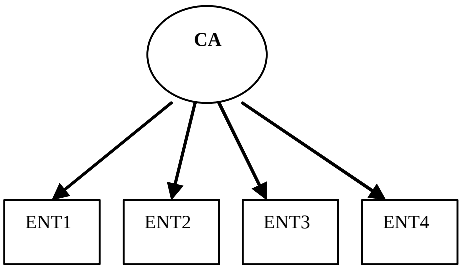

Figure 1: Single CA PKI architecture

Certificate path verification is very simple. In Single CA, PKI as all the users are certified by the same CA. However, if the CA's private key is compromised, the security of the entire system collapses. Even though this configuration is very easy to deploy, scalability is very poor if the community of users is very large.

Hierarchical PKI model

A PKI constructed with superior-subordinate CA relationships is called a Hierarchical PKI. In this type of PKI, as depicted in Figure 2, all of the subscribers / relying parties trust a single CA. This CA is called the Root CA (RCA in Figure 2) and is the most trusted anchor. The Root CA certifies its immediate descendants, which in turn certify their descendants, and so on. In this architecture, CAs issue certificates only for their lower level CAs and end entities. Typically, only one superior CA certifies each CA. Within this model, each participant must have knowledge of the Root CA's public key. Certificate path construction in a Hierarchical PKI is a straightforward process that simply requires the relying party to successively retrieve issuer certificates until a certificate is located that was issued by the trusted root. Hierarchical PKIs are scalable; certification paths are easy to develop and certification paths are relatively short(Koga and Sakurai, 2004). However, reliance on a single trust point may result in compromise of the entire PKI.

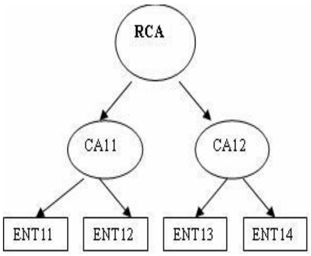

Figure 2: Hierarchical PKI

Merged Hierarchical PKI with cross-certifications at the root

When two or more business corporations are collaborated, then merging of their PKIs at the root level is a simple and straight forward approach. This is the best solution when the interoperability among the PKIs is temporary and dynamically change with the market requirements. The merging process needs to be low-cost, easily constructed and flexible. As shown in the Figure 3, the merged PKI is still a strict hierarchical PKI and thus the certificate path verification is also simple and straight forward.

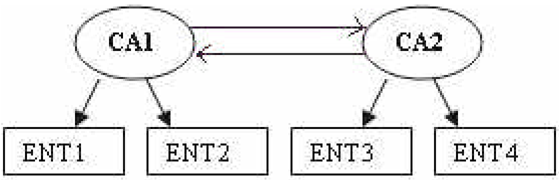

Figure 3: Merged Hierarchical PKIs with cross-certifications

However, this solution has the following drawbacks:

- It increases the cost required to maintain the security of root CAs as the merged PKI has more than one root CAs.
- The number of cross certifications is also more and it depends upon the number of PKIs to be merged.
- The certificate path length is increased and the cost required to verify the certificate is also more.

Mesh or Peer-to-Peer PKI model

A PKI constructed with peer-to-peer CA relationships is called a Mesh PKI or a Peer-to-Peer PKI. It is also referred to as a “web of trust”(Saxena, 2004). In a mesh style PKI, as depicted in Figure 4, each subscriber trusts its own CA. The CAs in this environment have no superior/ subordinate relationship. Each CA issues certificate to and is issued by a peer CA. Figure 4 depicts a mesh PKI that is fully cross-certified, however, it is possible to construct and deploy a mesh PKI with a mixture of unidirectional and cross-certifications(Lloyd et al., 2001). Compromise of a single CA can not bring down the entire PKI. Mesh PKIs can easily incorporate a community of users. However, certification path construction in a mesh PKI is more complex than in a hierarchical PKI due to the likely existence of multiple paths between a relying party’s trust anchor and the certificate to be verified, and the potential for loops and cycles in non-hierarchical certificate graphs.

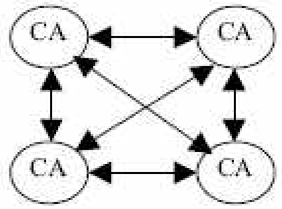

Figure 4: A Mesh PKI

HYBRID PKI MODEL

Hybrid PKI is the interconnection of different PKIs via cross certification. This enables relying parties of each to verify and accept certificates issued by the other PKI(Pez and et al., 2006). If the interconnection is between only hierarchical PKIs, Root CAs of all the participating PKIs cross certify each other facilitating interoperability between PKIs. Similarly, if the PKIs are mesh style, then a CA within each PKI is selected, more or less arbitrarily, to establish the cross certification. Usually the Federal PKI is considered as the arbitrator. Within each PKI, a CA can be selected based on the number of certificates issued. A CA with highest number of certificates issued may be selected to establish the cross certification. This results in the creation of a larger mesh PKI. However, the participating PKIs need not be of the same type. Figure 5 depicts a hybrid situation resulting from a hierarchical PKI cross-certifying a mesh PKI.

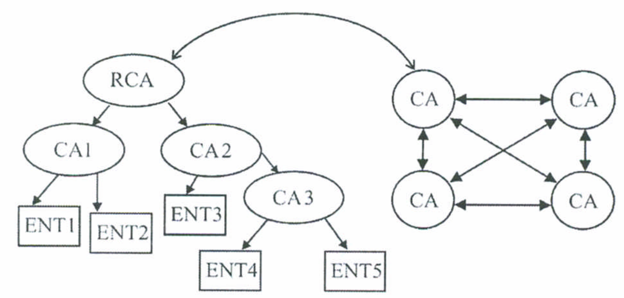

Figure 5: A Hybrid PKI

As the number of cross certified PKIs grows, the number of relationships between them grows exponentially resulting in complex certificate path verification.

BRIDGE PKI MODEL

Another approach to the interconnection of PKIs is the use of a “bridge” certification authority (BCA). A BCA architecture was designed to address the shortcoming of Hierarchical and Mesh PKIs(Adams and Lloyd, 2003). A BCA connects multiple PKIs to establish trust paths among them. The BCA is not intended to be used as a trust point by the users of the PKI. Its main task is to establish trust relationships between local CAs. As shown in the Figure 6, the BCA cross-certifies with one CA (known as a “principal” CA [PCA]) in each participating PKI.

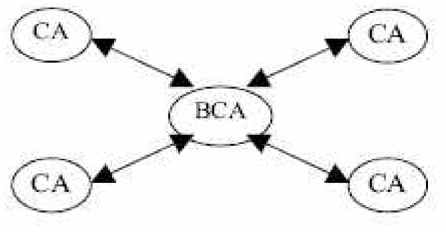

Figure 6: Bridge PKI

Since each PKI only cross-certifies with one other entity (i.e., the BCA), and the BCA cross-certifies only once with each participating PKI, the number of relationships in this environment grows linearly with the number of PKIs resulting in certification path discovery easier than mesh PKI. However, this model has a centralized component, BCA. If it fails, all cross-domain communication is unavailable. Also the certification path discovery in Bridge PKI model is more difficult than Hierarchical PKI.

A TRUST MODEL BASED ON GATEWAY CAS

A gateway CA(GWCA)(Guo et al., 2005) is a CA that is designed so as to allow certification to other different kinds of CA located anywhere in the global trust network, hence perhaps in different sub-networks, down to end entities (EE's) in these sub-networks. The GWCAs for their respective trust regions are connected in a ring fashion as shown in Figure 7. The Gateway CAs are connected in a ring configuration with each other, the intermediate and subordinate CAs may be connected in an hierarchical or bridge configuration. In this model, the Gateway CAs are the most trusted anchors.

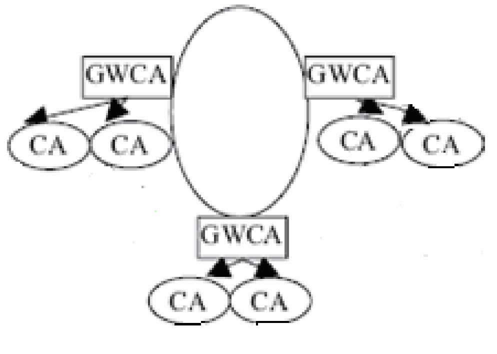

Figure 7: Gateway CA PKI

In this way, GWCA model provides a means of PKI interoperability in the national and international levels.

CERTIFICATE PATH VERIFICATION IN HIERARCHICAL PKI

Digital Certificates

In automated business transactions, each user or end entity is given a digitally signed data structure called *digital certificate*. A digital certificate(Saxena, 2004) is a digital credential that takes the following form:

$$C_{S,\,I} = Cert(\ n, S, I, sig_i\,, pk_{s,}\, D, A)$$

where *n* is the serial number, *I* is the issuer of the certificate, *S* is the subject of the certificate, sig_i is the signature of the issuer *I*, pk_s is the public key of *S* , *D* is the validity period of the certificate, and *A* represents the additional data. Digital certificates have emerged as a popular tool to provide authentication, privacy, integrity, non-repudiation and other security requirements in modern day transactions.

DIGITAL SIGNATURE GENERATION AND VERIFICATION

Digital signature generation and verification are important part of dealing with digital certificates. Key generation (or update) time, signature time, and verification time are all indicators of a signature scheme’s performance. However, no one aspect alone is enough to judge whether one signature scheme is better than another for all situations. Many earlier performance comparisons take an informal approach at resolving this problem by first looking at a specific situation and then picking which operation seems to be most important for it (Wiener, 1998). This works well for simple situations but does not help when it is unclear which operation is most important or in seeing the entire picture with regards to performance tradeoffs. It is very well known that for signature generation, DSA(Digital Signature Algorithm) is faster than RSA(Rivest, Shamir and Adleman) algorithm, where as for signature verification, RSA is faster than DSA (Wiener,1998; Kaliski,1993).

Certificate Path

A *Certificate path* is an ordered sequence of digital certificates where the subject of each certificate in the path is the issuer of the next certificate in the path. A certificate path begins with a trust anchor certificate and ends with an end entity certificate.

To describe the notation defined by X.509 to represent certificates, the following relationships between CAs RCA, CA12, and an end entity ENT14 in Figure 2 are initially assumed. ENT14 is an end entity certified by CA12, and CA12 is directly subordinated to RCA in the hierarchy. Therefore, the notation for the above path is represented as follows:

CA12 << ENT14 >> , RCA << CA12 >>

The processing of certificate paths may be a very complicated and time demanding operation, depending on the length of the certificate path and the possible inclusion of relations using cross-certification. Cross-certification is required when users from different PKIs are to be able to trust each other's certificates.

Certificate Path Verification

Certificate Path Verification is building a trusted path between the trust anchor certificate and the target entity certificate based on the trust relationship among the CAs of the PKI and validating the certificates. The longer a path becomes, the greater the potential dilution of trust in the certification path(M. Cooper et al., 2005). That is, with each successive link in the infrastructure (i.e., certification by CAs and cross-certification between CAs) some amount of assurance may be considered lost. The longer and more complicated a path, the less likely it is to validate because of basic constraints, policies or policy constraints, name constraints, Certificate Revocation List(CRL) availability, or even revocation.

Issues in the Certificate Path Verification

The processing of a certificate path in order to verify its validity is composed of two steps:

Path construction - the certificates are retrieved from a repository called LDAP(Lightweight Directory Access Protocol) Directory(Boeyen et al., 1999) and the path is constructed.

Path validation - the certificates in the path are checked for integrity, validity period and information related to semantics verified.

The path discovery process is both computation and communication intensive, since it involves discovering a chain of CAs in the path. The following factors contribute to the complexity:

- There are many possible trust topologies as described in section 2. Among these, the strict hierarchy of CAs offers the simplest form for path discovery and validation. The mesh and bridge-CA topologies necessitate the Relying Party(RP) software to be intelligent enough to navigate through the chain of certificates by avoiding loops and making appropriate decisions about the selection of links. Changes in the trust hierarchies over time also necessitate changes in the relying party software.

- The trust policy issue is also a major concern during path validation. Each RP has an acceptable trust policy for the chain of CAs that it is trying to validate for a given certificate (Lloyd et al., 2001). The policy would describe the minimum level of CA practices that it expects from each of the CAs on the path.
- The response time is also one of the issues for certificate path validation. An RP has to respond quickly to its users for the service they request. However, it cannot do so until the path construction and the subsequent certificate validation are complete.
- The certificate repository availability is also an important concern to an RP. All of the CAs and the relevant certificate and revocation information repositories need to be available for an RP to discover and validate a path. Data collection is a problem if a trust path is long, with several CAs, as some entities along the path may not be accessible during validation(Pinkas, 2001). In other words, even though a CA-CA certificate is valid, due to the inability to access a CA's data at the validation time, an RP may not be able to validate a certificate.
- The cost of path construction and validation should be minimum. The task of path discovery and validation is an expensive process. Substantial cost savings may not be realized if the entire process of validation is repeated for each certificate that the RP receives, even when delegated to a trusted validation server.

From the above discussion, it is clear that certificate validation is a complex process involving considerable amount of communication and computational overhead.

We can construct Certificate path in Hierarchical PKIs by two ways:

- Forward path construction in which the path is constructed from end entity certificate to the trust anchor certificate.

  The Forward certificate path construction is a straightforward approach whereby we start constructing the path from target(end entity) certificate to the root certificate via the intermediate CAs' certificates. It is a straightforward approach because the path is unidirectional and unambiguous. Each certificate contains its issuer's information and so it is a simple task.

- Reverse path construction in which the path is constructed from trust anchor certificate to the target(end entity) certificate.

Reverse path construction is not straightforward because it is difficult to determine the exact path from Root CA to the target certificate directly. To make a comparison between forward and reverse path constructions, we adopt the child-sibling approach from(Huang, 2005) to construct a binary tree, T ‘, from an arbitrary general tree, T, so that we can build the path without any ambiguity. In this child-sibling binary tree representation, the first child from left of any internal node of T becomes the left child of T ‘ and the next node in the same level of T becomes the right child of the previous node in T ‘. The link to the left child node of the binary tree T ‘ is labeled with a bit 0 and the link to the right child with a bit 1. Thus each CA node in the hierarchy holds a codeword consisting of the accumulated 0-1 sequence from the root to the target entity. Subordinate CAs get their codeword from their parent CA.

In Figure 8, T is the input tree and T ‘ is the equivalent binary tree.

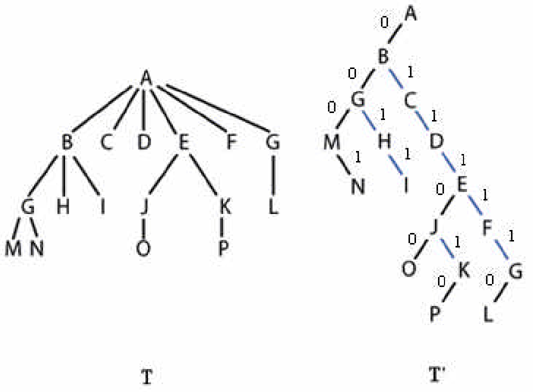

Figure 8: Conversion of Hierarchical Structure to a Binary Tree

In the binary tree T ‘, the left link is labeled with bit 0 and the right link with bit 1. The number of ‘0’s in the codeword represents the level in the tree. For example, the codeword of the node P in T ‘ is 0111010. Since the depth of the nodes increases after transforming the general tree to a binary tree, the path verification time also increases.

EXPERIMENTAL RESULTS

Forward Vs. Reverse Certificate Path constructions in Hierarchical PKI for Certificate Path Verification

We have implemented the methods of Forward and Reverse certificate path construction in Java with OpenSSL tool. It can be seen from Figure 9 that, the time required for certificate verification using Forward path construction method is less than that of the Reverse path construction method. This is because, using the reverse path construction method, the depth of the nodes increases after transforming the general tree to the binary tree.

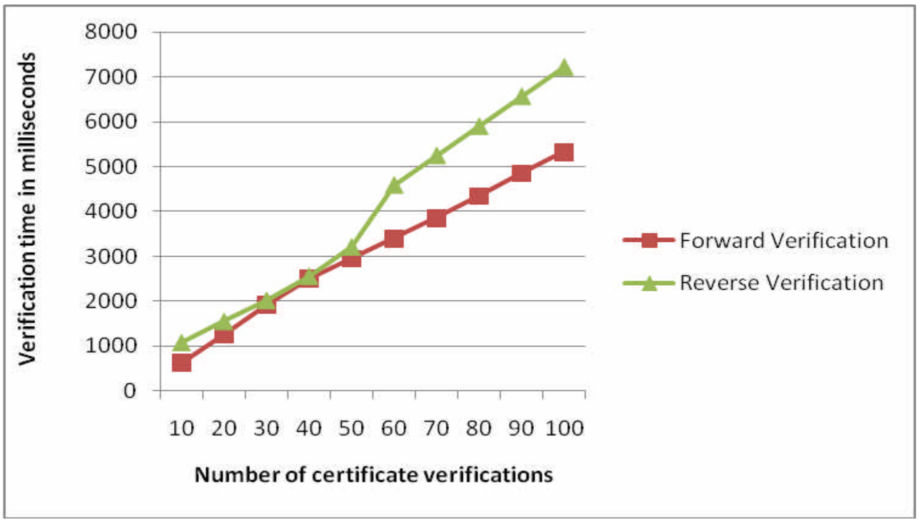

Figure 9: Path Verification time with Forward and Reverse Path Constructions

Proposed method of converting Peer-to-Peer PKI to DFS Spanning tree to simplify certificate path verification

In automated business transactions, the method of building a trusted path between the trusted anchor and the target entity constructs a path as each certificate is retrieved from a repository via LDAP, a protocol employed for repository access operations. The simplicity of such method resides in the fact that only one certification path is possible in the case of Hierarchical PKI. Also the path is unidirectional and simple.

But, building a path in Peer-to-Peer PKI is nothing but traversing a complex graph. However, from the simplest viewpoint, writing a path-building module can be nothing more than traversal of a spanning tree, even in a very complex cross-certified environment. In the proposed method, we traverse the graph representing Peer-to-Peer PKI in Depth First Search(DFS) order and construct a DFS spanning tree. In a DFS spanning tree, we have a single path between any two users. We construct DFS spanning tree as follows:

Let graph G represent the Peer-to-Peer PKI in question. Each vertex in the graph represents a CA. We shall start from a given vertex v in the graph, G. We will mark this vertex v as visited. The next step is to pick a new unvisited vertex(any one of the adjacent vertex can be selected). Call this vertex w. We then explore this new vertex depending upon its adjacent vertices recursively. The search procedure terminates after all the vertices are explored(i.e. visited). As each vertex is visited, it is added to the DFS spanning tree.

```
Algorithm Construct_DFSSpanningTree(v)
{
    //G is a Graph representing Peer-to-Peer PKI. Each edge is bidirectional
    //Visited[1:n] is an array to remember the visited information
    // v is the starting vertex

    Visited[v]=1; //mark the starting vertex visited
    Add v to the DFS spanning tree
    for(each vertex w adjacent to v)
            if(visited[w]=0)
    Construct_DFSSpanningTree(w); //continue to explore
}
```

For example, Figure 10(a) is a graph that represents a Peer-to-Peer PKI in which the path is bidirectional. The DFS order of the tree is: CA1, CA2, CA3, CA6, CA4, CA5, CA7.

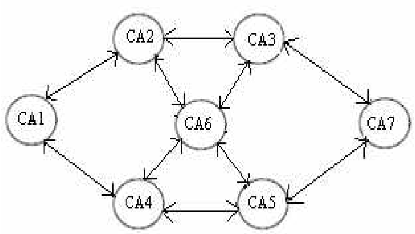

Figure: 10(a) A Peer-to-Peer PKI

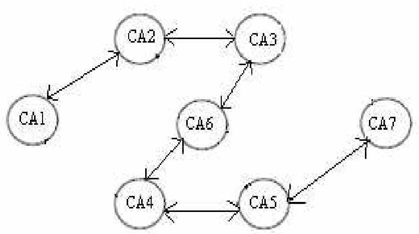

Figure: 10(b) DFS spanning tree

Figure 10(b) represents the DFS spanning tree of the graph shown in Figure 10(a). From Figure 10(b), we can observe that there exists single path between any two CAs. Since there exists single path between any two CAs in the spanning tree, the certificate path construction is simple and straight forward. The complexity due to multiple paths between CAs and ambiguity in choosing one of them is removed.

CONCLUSIONS

In Hierarchical PKI, certificate path is unidirectional, so certificate path development and validation is simple and straight forward. Certificate path verification using forward path construction is the most popular technique of building certificate path in Hierarchical PKIs. In this paper we have shown that the time required to verify certificates using forward path construction method is less than that of the path verification using reverse path construction method in Hierarchical PKI. In mesh or Peer-to-Peer PKI, certificate path verification is a complex task since there exist multiple paths between CAs. In the paper, we proposed an efficient method to convert a mesh or Peer-to-Peer PKI to its equivalent DFS spanning tree to simplify the certificate path construction. Thus the complexity of certificate path verification in Peer-to-Peer PKIs due to multiple paths between users can be removed.

REFERENCES

Adams, C. and Lloyd, S. (2003) *Understanding Public-Key Infrastructure: Concepts, standards, and Deployment Considerations,* (2nd edn.), Bostan, Addison Wesley.

Adams, S. and Farrell, S. (1999) 'Internet X.509 Public Key Infrastructure Certificate Management Protocols, Network Working Group Request for Comments 2510' (online). Available from http://www.ietf.org/rfc/rfc2510.txt

Boeyen, S. et.al. (1999) 'Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2', Network working group, RFC 2559

Cooper, M. et. al. (2005) 'Internet X.509 Public Key Infrastructure: Certification Path Building', *Network Working Group,* RFC 4158.

Cronin, E., Malkin, T. et.al (2003) 'On the Performance, Feasibility and use of Forward-Secure Signatures', CCS'03, Washington, DC, USA.

Guo, Z., Okuyama, T., et.al. (2005) 'A New Trust Model for PKI Interoperability', *Proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005),* IEEE.

Huang, H. (2005) 'On the Protection of Link State Routing and Discovery of PKI Certificate Chain in MANET', A Ph.D thesis submitted to the Graduate Faculty of North Carolina State University.

Kaliski, B. (1993) 'A Survey of Encryption Standards', *RSA Laboratories,* IEEE Micro.

Koga, S. and Sakurai, K. (2004), 'A Merging Method of Certification Authorities Without Using Cross-Certifications', *Proceedings of the International Conference on Advanced Information Networking and Application (AINA'04),* IEEE

Lloyd, S. et. al. (2001) 'CA-CA Interoperability', PKI Forum (online). Available from http://www.pkiforum.org/pdfs/ca-ca interop.pdf

Mazaher, S. and Roe, P. (2003) *A survey of state of the Art in Public Key Infrastructure,* Norway, Norsk Regnesentral.

Pez, R. Satizbal, C. et. al. (2006) 'Building a Virtual Hierarchy for Managing Trust Relationships in a Hybrid Architecture', *Journal of Computers,* 1:7, 60-68.

Pinkas, D. (2001) *'Delegated Path Validation and Delegated Path Discovery Protocols'*, Internet Draft.

Saxena, A. (2004) *Public Key Infrastructure Concepts, Design and Deployment,* New Delhi, Tata McGraw Hill.

Serranoa, J.H., Satizbal, C. et.al (2007) 'Building a virtual hierarchy to simplify certification path discovery in mobile ad-hoc networks', *Computer Communications*, 30: 7, 1498-1512.

Thales (2000), 'Elliptic Curve Cryptography', e-security white paper.

Weise, J. (2001) 'Public Key Infrastructure Overview', Sun BluePrints™.

Wiener, M.J. (1998) 'Performance comparison of public-key cryptosystems', *CryptoBytes,* 4(1).

Zuccherato, R. (2003) *'Using a PKI Based Upon Elliptic Curve Cryptography-Examining the Benefits and Difficulties'*, Entrust-Securing Digital Identities and Information.

Balachandra is Selection Grade Lecturer in Department of Information and Communication Technology, Manipal Institute of Technology, Manipal.

Dr. Prema K.V. is Professor and Head in Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal.

Chitkara University
Saraswati Kendra, Plot 11-12, Dainik Bhaskar Building
Sector 25-D, Chandigarh-160014, India
Email: journal@chitkarauniversity.edu.in
Website: www.chitkara.edu.in/journal/index.php