A Scalable PKI Based on P2P Network

Zhiwei Gao[1,2], Jinsheng Fan[1],Yufeng Jia[1], Li Zhang[1],
[1]*Department of Computer Science,*
Shijiazhuang Railway Institute of Technology, Shijiazhuang, 050043, China
[2]*Department of Computer Science,*
Beijing Institute of Technology, Beijing, 100081, China
E-mail: gao_zhiwei@163.com, fanjsh@sjzri.edu.cn,
jiayufeng@163.com, zhangli@163.com

Abstract

Public key infrastructure (PKI) is a powerful tool for protecting information. Currently a PKI system shows a trend toward an emerging global PKI which becomes more complicated. The global PKI has to handle an enormous number of queries for cryptographic certificates which attest the authenticity of public keys. So a decentralized organization of the PKI is advantageously. Therefore we developed a specialized Peer-to-Peer-PKI model realizing efficient search and transfer of certificates and trust-recommendations. Our model based on our own rigorous binary tree algorithm and has four advantages. First, there is no any bottleneck problem when establishing a certification path or authenticating. Second, the authentication path is short with two constant logic steps. Third, in our model the entities need to mutual authenticate don't need to inquire or download the CRL. Fourth, it's easy to extend and suitable for large-scale network.

1. Introduction

We present P2P-PKI, a decentralized public key infrastructure (PKI) based on the web of trust model. Public key infrastructure is a powerful tool for protecting information. Currently PKI systems shows a trend toward an emerging global PKI, where individual PKI domains establish trust relationships via cross -certification technology. However, as a PKI becomes more complicated, so does the work require for validating an individual certificate. The first step is certification path discovery: constructing a "chain of certificates". It is challenging to locate appropriate resources to establish a candidate path and to maximize its chance of being valid [1].

The global PKI spans many countries and consists of many domains, CAs, repositories, and users. PKI protocols need to be robust in such a complex network environment. By establishing trust relationships between domains, cross-certification confronts us with a complex "certificate topology". Moreover, users in different PKI domains may display completely different behaviors that may impact the effectiveness of PKI protocols.

Prior research has analyzed certification path discovery. But there are three questions in prior works. First, it adopted a tracing mechanism when authenticating identity, so the nearer to the root CA, the easier to bring a bottleneck problem. Second, each local CA does things in its own way, so it becomes complex for users to authenticate if they belong to different CA domains. Although it's possible for users who are in different CA domains to authenticate each other by using some techniques such as bridge-CA, strict hierarchy CA, cross-certification and cross-reorganization etc, these techniques have not solved the tracing bottleneck problem in the CA models. Third, there are querying bottleneck issues in the management of certificates distributing and certificate revocation lists. With the increasing of the end-users in a CA domain, the load of the CA server storing and checking certificates will be difficult to bear, and tend to bring querying bottleneck.

In this study, we developed a specialized model realizing efficient search and transfer of certificates and trust-recommendations. It is based on a combination of traditional PKI and our own efficient and scalable Peer-to-Peer lookup protocol [2]. We make four contributions: First, there is no any bottleneck problem when searching or authenticating in our framework. Second, The authentication path is short with no more than two entities intervened. Third, in establishing certification path, we do not need to inquire about a certificate revolution list. Fourth, our framework is scalable and easy to extend and suitable for large-scale network.

The remainder of this paper is organized as follows: Section 2 discusses the related works. Section 3 gives the preliminary knowledge of basic PKI principles and certification path discovery. Section 4 describes the main concept of RBT-P2P network model. Section 5 and 6 presents our design theory, and implementation. Section 7 discusses the security issues and Section 8 concludes.

2. Related Works

Prior research has been done on P2P-PKI. Thomas wlöfl designed a PKI Based on a P2P Network [14], but his work is based on Chord [13] and don't incorporate certificate revocation and expiration into his P2P-PKI. Meiyuan Zhao and Sean W. Smith [1] propose and implement a simulation framework and a probability search tree model for systematic performance evaluation. Their model is now the largest simulated PKI architecture. Elley et al. [4] presented a comparison of two directions for path building, and concluded that building in the reverse direction is often more effective than building in the forwarding direction. Russell et al. [11] analyzed the performance issues for constructing and validating long certification paths in cross-domain PKI systems, and proposed the concept of virtual certificates and synthetic certificates to avoid re-constructing and re-verifying certification paths. Lloyd published a white paper [12] that discussed options for effective and efficient certification path construction algorithm. Arnes implemented a simulation to evaluate certificate revocation performance [3]. Mu˜noz et al. implemented CERVANTES, a testbed for certificate validation [7]. Unlike these studies, our framework based on our own efficient and scalable Peer-to-Peer lookup protocol [2].

3. Preliminaries

3.1 Basic PKI principles

PKI was first proposed [6] for securely distributing a user's public keys. From a formal perspective a PKI offers a statement about the binding of a certain public key to a certain user at a particular time. In this context a user can be a system, a person or an organization which is participating in the PKI. It has now evolved to architectures providing comprehensive services for public key certificates; these services include storing and retrieving certificates, maintaining and updating certificate status, and validating certificates. In a traditional X.509 [8] PKI system, the certificate storage service is provided by a repository that supports protocols for users to store and retrieve directory information; the protocol used most commonly here is the Lightweight Directory Access Protocol (LDAP) [10]. The certificate status information (CSI) service communicates the validity status of certificates. A certificate is typically considered as "valid", "revoked", or "unknown". Classical approaches to CSI includes periodically updated data structures such as a certificate revocation list (CRL) [8], and online protocols such as online certificate status protocol (OCSP) [9].

3.2 Certification Path Discovery

The user who tries to validate a certificate is referred to as relying party. A certificate validation service handles certification paths, sequences of certificates representing a trust path to the certificate of interest. In a path, consecutive certificates are linked together by having the subject of the previous certificate match the issuer of the next certificate.

A certificate validation service is composed of two stages: certification path discovery and certification path validation. The latter stage is well established. RFC3280 defines an algorithm to validate a certification path. Basically, the algorithm examines each certificate in the path to decide if they satisfy all required conditions. Unfortunately, the algorithm for actual construction of candidate certification paths is not well defined. Several issues affect the practicability and efficiency of the certification path discovery process [1]. One critical issue is the increasing complexity of PKI architectures. The trends toward bridging and cross-certification hasten the emergence of a global PKI architecture. However, this architecture creates new challenges for certification path discovery; algorithms must construct a path by traversing different PKI domains, dealing with different PKI policies and handling different protocols.

4. RBT-P2P network model

To give readers a better understanding of our model, we first provide the general concepts and terminology in our P2P model based on rigorous binary tree algorithm (RBT-P2P model). The details of RBT-P2P model are described in our prior work [2].

Definition 1: *Rigorous binary tree. For a random node of a binary tree, if it has one child node, then its left child node and right child node must exist at the same time.*

Definition 2: *Rigorous binary tree extension. After a random leaf of a rigorous binary tree produces two child nodes, the original rigorous binary tree becomes a new rigorous binary tree.*

Definition 3: *Rigorous binary tree code algorithm: The letter* T *represents a rigorous binary tree, "*A*" represents a random node in* T*, h_a represents the depth of node* A*, and N_a represents its code. The code of* T*'s root node was set as 0. The code of* A*'s left child is equal to N_a. The code of* A*'s right child is equal to ($N_a + 2^{h_a}$), The depth* A*'s child is h_a +1.*

Theorem 1: *Rigorous binary tree mapping theorem: For any one integer I (I >=0), there is one and only one leaf node X whose code (N_x) and depth (h_x) can accord with $N_x = I\%2^{h_x}$, among all leaf nodes in a fixed rigorous binary tree. (Here % denotes modular arithmetic.)*

The proof process of theorem 1 is included in our prior work [2]. In that work we propose a P2P model based on rigorous binary tree and here we name this P2P model as RBT-P2P model.

5. PKI based on RBT-P2P network

This section describes the design principles of our PKI-P2P model.

5.1 The base of trust

In our work, P2P-PKI realizes a user-centric trust model (“web of trust”, cf. [16]). Each node is directly and totally responsible for deciding which other nodes to trust.

We make the following assumption: a user A already knows the public key PK_B of some user B, and furthermore that A trusts B to issue certificates. Then A can validate C's public key if C can present a certificate $CERT_B(C; PK_C)$ to A. Here $CERT_B(C; PK_C)$ denotes the signature signed by B. If there is no problem, then A can trust C.

5.2 The web of trusted

We select those authority-users who have held their certificates as super-peers. The trusted web is started with one super-peer coded as the root node in a rigorous binary tree. When a super-peer needs to extend, it selects a new super-peer, sets a code value and a depth value for the new member, renews own code value and depth value.

At the same time, to establish an initial trust relationship, the original super-peer with a certificate issued by a CA and the new super-peer with a certificate issued by a CA need to authenticate each other according to the CA mechanism. Then the original super-peer signs a signature for the new super-peer. Via the signature, the new super-peer can build a credit relationship with the other super-peers in the trusted web.

According to this manner, the number of the members in the trusted web can automatically increase with the scalability of the network.

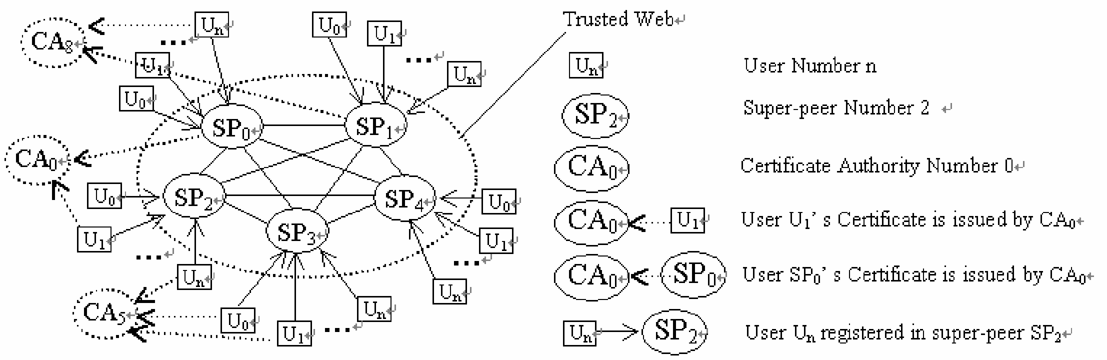
Trusted Web
CA8
CA0
CA5
SP0
SP1
SP2
SP3
SP4
U0
U1
Un
User Number n
Super-peer Number 2
Certificate Authority Number 0
User U1' s Certificate is issued by CA0
User SP0' s Certificate is issued by CA0
User Un registered in super-peer SP2

Figure 4.1 The architecture of users and super-peers in a trusted web.

To establish an initial trust relationship, according to the CA mechanism, any user holding a certificate issued by a CA can authenticate each other with a super-peer in the trusted web. Obtaining the initial trust, the user will receive a signature signed by the super-peer. Holding the signature, according to **Theorem 1**, the user will register its information about its certificate in a corresponding super-peer in the trusted web. According to this fashion, all users are distributed in various super-peer areas. The structure is described as in Figure 4-1.

After the extending the original super-peer will distribute approximately half of its registered users’ information to the new super-peer by a signature manner. Using the extending mechanism, our model can overcome the checking bottleneck problem.

When a CA wants to revoke a user X's certificate, it can issue the revocation message to a super-peer. Via the super-peer, according to **Theorem 1**, the revocation message will be forwarded to a corresponding super-peer, in which the user whose certificate should be revoked has registered in the trusted web, i.e., a super-peer can know the state information about its registered users' certificates.

When a user wants to authenticate each other with another user in the trusted web, it at most involves two super-peers. One is the user's registered super-peer and the other is another user's registered super-peer. When they check the corresponding super-peer, they can also get the state information about their certificates. So the authentication path is shorter and it doesn't need to query the CRL in our model.

6. System implementation

6.1 Initialization

We suppose that the sponsor of the trusted web is the authority user F. The sponsor acts as the first super-peer and it's code and depth (N_f , h_f =0). F produces two tables: the super-peer table and registered-user table.

The super-peer table is used for noting the correlated information of all super-peers in the trusted web and held by every super-peer. The registered-user table is used for noting the correlated information of all users registered in a super-peer. As table 5-1 and table 5-2 show.

Table 5-1 The super-peer table.

Name	Depth	Code	Certificate	Public key	Identity number	Signature
F	h_f	N_f	Cer_f	k_f^e	ID_f	$E(k_f^d(hash(ID_f,k_f^e)))$

Table 5-2 the registered-user table of the super-peer F

Name	Certificate	Public key	Identity number	Signature	Certificate state
X	Cer_x	k_x^e	ID_x	$E(k_f^d(hash(ID_x, k_x^e)))$	$S(0/1)$

In our paper, we make the following rules:

hash(*): operating hash function on (*);
k_x^e : the public key of an entity X;
k_x^d : the private key of an entity X;
$E(k_x^d(*))$:computing on (*) using k_x^d with an algorithm E.

The identity number of a user X is created by operating hash function on the signature item in the user X's certificate. We use 1/0 marking the state of a user's certificate, 1 denoting revoked, 0 denoting not revoked.

6.2 Joining the P2P-PKI

For any user to join the trusted web, the user needs to know at least one super-peer. Through the registered user, the user can know a super-peer. In this paper, we did not describe how to know a registered user.

We suppose that a user B has known a super-peer A whose code and depth are (N_a, h_a), and B wants to join the system. The process was shown in Figure 5-1:

Step1. Establish the initial trust relationship between B and A. According the traditional CA mechanism, B and A authenticate each other. Then, A gets B's identity number ID_b using the hash(B's IP-Adress).

Step2. A calculates and finds a super-peer corresponding with ID_b. In the light of **Theorem 1**, there is one and only one super-peer X exists, and its code and depth (N_x, h_x) satisfy $N_x = ID_b \% 2^{h_x}$.

Step2.1. If $N_x = N_a$, then X is the super-peer A itself.

Step2.2.If $N_x \neq N_a$, then A select a super-peer from its super-peer table, and calculate $ID_b \% 2^{h_x}$, compare the result with N_x . This process continues until finding the proper super-peer X .

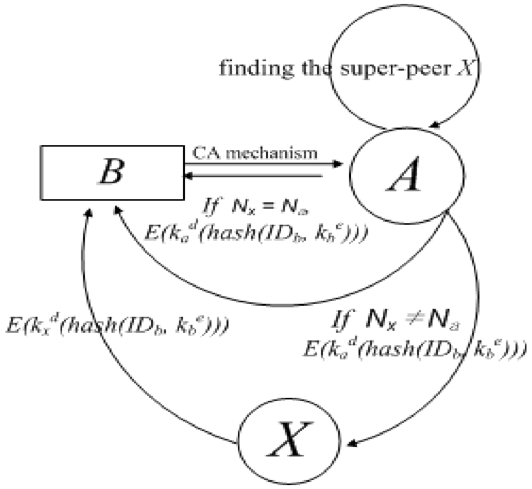
finding the super-peer X
B
CA mechanism
A
If $N_x = N_a$
$E(k_a^d(hash(ID_b, k_b^e)))$
$E(k_x^d(hash(ID_b, k_b^e)))$
If $N_x \neq N_a$
$E(k_a^d(hash(ID_b, k_b^e)))$
X

Figure 5.1 The process of joining the P2P-PKI

Step3. After finding the exact super-peer X, A can signs B's information and forwards it to the super-peer X. In addition, A signs X's information and forwards it to B.

Step4. The super-peer X checks B's information, then makes a signature $E(k_x^d(hash(ID_b, k_b^e)))$ for B, sends it to B and adds B's information in its registered-user table.

Step5. According to Step1, B believes A's signature. When B receives X's information signed by A, then B can believe X. When B receives the signature in Step4, B can validate it. If there is no any problem, B saves it and k_x^e. Registration is over.

During the registration, if an attacker intercepts and captures the signature and wants to imitate it, in PKI theory, it is impossible.

6.3 Expanding the P2P-PKI model

In our work, the P2P-PKI model is easy to extend and suitable for large-scale network.

When the number of the users registered in a super-peer reaches the quantity limit, the super-peer needs to select a new super-peer, code it and distribute some of its registered users to the new super-peer.

Here we suppose that super-peer X needs to extend, Y is the new selected super-peer, X's code and depth are (N_x , h_x). The expanding process is:

Step1. In the light of the CA mechanism, X and Y authenticate each other.

Step2. According to **Definition 3**, Recalculate the code and depth, update the super-peer table and issue the updated information. The example was described in Figure 5-2. X appends Y's information in the super-peer table, signs the super-peer table and sends it to Y. X signs the updated information and forwards it to all the other super-peers in the trusted web.

Step3. Each super-peer in the trusted web credits each other. After checking the updated information about the super-peer table signed by X, each super-peer can credit the new super-peer Y. Each super-peer updates its super-peer table.

Step4. Manage the users' information in the registered-user table. X divides the users' information into two parts, and signs those users whose ID according with N_y to Y, and X signs Y's information and sends it to those users.

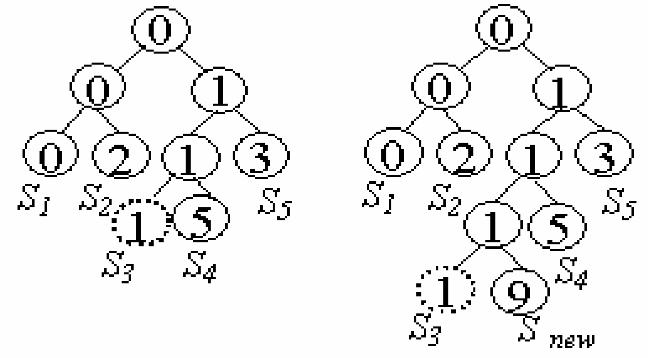

Fig 5.2 The extending of the super-peer S3

Step5. After receiving the information from X, Y checks the signature with X's public key and saves the super-peer table. Then Y makes a signature for each user whose information was sent to Y by X, and sends it to the user.

Step6. For each user whose information is sent to Y by X, when it receives Y's information signed by X, it can check the signature with X's public key. If the check is right, the user can believe Y.

In this way, one part of the users' information is transferred to the new super-peer Y, and the related super-peer lightens its load.

6.4 Authentication between two entities.

For clarity, we provide an example to describe the finding process. We suppose that user X has registered in super-peer A, user Y in super-peer B, X trusts A and Y trusts B, as described in Figure 5.3.

X sends a message M_1{X, ID_x, k_x^e, $E(k_a^d(hash(ID_x, k_x^e)))$, A, k_a^e, t_1, sign()} to Y. Here, "t_1" denotes period of validity and it is used to ward playback attack. "sign()" denotes the signature with the sender's private key on the entire message.

Y checks M_1. If there is no problem, Y can ensure M_1 is created by X. However, Y can not believe X, because Y don't believe A. To check X and A, Y sends a message M_2{Y, ID_y, k_y^e, X, ID_x, k_x^e, A, k_a^e, t_2, sign()} to B.

B validates M_2 and sends M_3{B, k_b^d, ttl_1, $E(k_b^d(hash(A, k_a^e, ttl_1)))$, X, ID_x, k_x^e, t_3, sign()} to A.

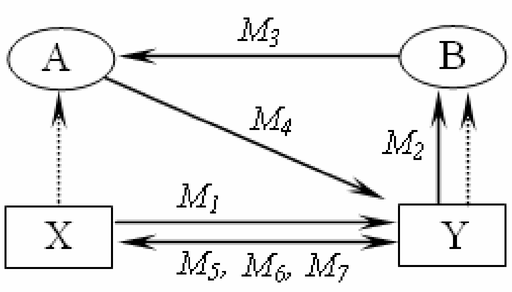

Figure 5.3 The authentication process

A validates M_3, fetches ID_x, finds X's certificate and checks the state of X's certificate status. If X's certificate has been revoked, A responds Y a message "X's certificate has been revoked". Else, A makes a signature $E(k_a^d(hash(A, k_b^e, ttl_2)))$ and then sends a message M_4{ A, k_a^e, ttl_1, $E(k_b^d(hash(A, k_a^e, ttl_1)))$,B, k_b^e, ttl_2, $E(k_a^d(hash(B, k_b^e, ttl_2)))$, t_4, sign()} to Y.

Y validates M_4. If there is no problem, Y can trust A, because Y trusts B. But Y can't ensure the other side is X. To validate X, Y creates M_5 {Y, ID_y, k_y^e, $E(k_b^d(hash(ID_y, k_y^e)))$, B, k_b^e, ttl_2, $E(k_a^d\ (hash(B, k_b^e, ttl_2)))$, $E(k_x^e\ (random\text{-}num_1))$, t_5, sign()} and sends it to X.

X validates M_5. If there is no problem, then X will trust B, i.e., X believes k_b^e. X checks $E(k_b^d(hash(ID_y, k_y^e)))$ with k_b^e. If there is no problem, then X can believe Y. To validate the other side, X decrypts $E(k_x^e\ (random\text{-}num_1))$ with k_x^d and sends a message M_6{$E(k_y^e\ (random\text{-}num_1))$, $E(k_y^e\ (random\text{-}num_2))$,t_6, sign()} to Y.

Y decrypts $E(k_y^e\ (random\text{-}num_1))$ and $E(k_y^e\ (random\text{-}num_2))$ with k_y^d and gets random-num_1 and random-num_2. Now, Y can ensure that the other side is X. Y sends a message M_7{$E(k_x^e\ (random\text{-}num_2))$,$t_7$, sign()} to X.

X decrypts $E(k_x^e\ (random\text{-}num_2))$ with k_x^d and gets random-num_2. Now X can ensure the other side is Y.

From now on we can see that at most involves two super-peers when two users authenticate each other. One is the local registered super-peer and the other is the remote registered super-peer. So the authentication path is shorted.

6.5 Certificate revocation.

From time to time, a previously published public key will need to be invalidated, generally when the corresponding private key has been exposed. This problem is easily solved in our model.

In the registered-user table, the state of a user's certificate can be noted. If a user's certificate needs to be invalidated, then what needs to do is deleting the user's record in the registered-user table of the corresponding super-peer. For some reasons, if a CA wants to revoke a user X's certificate in its period of validity, the CA can issue a certificate revocation message including the user X's certificate. After authenticating each other with a super-peer in the light of CA mechanism, the CA sends the certificate revocation message to the super-peer. Computing on X's certificate included in the revocation message, the super-peer can calculate ID_x. Then, according Theorem 1, the super-peer can find the super-peer corresponding with ID_x. Finally the super-peer can send the certificate revocation message to the latter. The latter then changes the value marking the state of the user X's certificate.

If a user itself wants to revoke its certificate, it can directly send a revocation application to its registered super-peer. Then the super-peer will directly change the value marking the state of the user X's certificate.

In this way, if a user X whose certificate has been revoked wants to communicate with another user Y, when authenticating each other, Y can receive a message about X's certificate state from a related super-peer and know "X's certificate has been revoked".

The revocation about a super-peer's certificate is relatively complex. If a super-peer discloses its private key, then a new super-peer should be selected replacing the disclosed one and hold the disclosed one's code and depth. The updated message should be sent to each super-peer in the trusted web. All the users registered in the disclosed super-peer need to afresh register.

Now we can see a user do not need to inquire about a certificate revolution list and could establish a certificate path in efficient way.

7. Security analysis

In this section, we argue that our P2P-PKI has very strong security properties: the system is resilient to a wide range of attacks.

7.1 Managing the failure of peers

A peer's failure can make a value/key pair unavailable if the value/key pair has been stored in the peer. To avoid this case, we adopted a redundancy mechanism; i.e., when a super-peer wants to fetch a peer from its registered-peer table to store a value/key pair, it fetches two peers instead of one. Thus, if a peer fails, the value/key stored in that peer is still available from the other one.

To manage two peers storing the same value/key pair, we can make them contact each other periodically. If one peer detects that the other is unreachable over a limited time period, the former can message its local super-peer. Then, the local super-peer can select another peer instead of the failed one to store the same value/key pair.

A super-peer's failure can affect a subset of peers in a logic area.

To resolve this problem, in a similar fashion, we adopted a redundancy mechanism. When any one super-peer wants to produce a new super-peer, it selects k high-powered peers as a new group of super-peers with the same contents.

In one way, the redundancy mechanism increases the size of the super-peer table, but on the other hand, it decreases the average load of each super-peer and makes the system more reliable.

7.2 Prevention of DoS attack

To prevention of DoS attack we adopted the traffic model [15] that captured the key characteristics of how queries flow through the system. The Model's various traffic management policies that contained the effects of malicious nodes issuing a potentially large number of "useless" queries. It's also defined metrics by which those traffic management policies could be evaluated and we used those metrics to build an understanding of the trade-offs between using various policies.

An important thinking behind design is: while this system is under attack, the techniques that we have developed can be used to provide users with a degraded quality of service while detection algorithms are concurrently being used to identify offending nodes and disconnect them from the network. The P2P-PKI model will not have "turtle shell" architectures in which if just a few malicious nodes get through an outer shell of defense, they can easily attack a soft inner core.

8. Conclusion

In order to solve the performance issue on certification path discovery, we present a distributed model Based on the rigorous binary tree code algorithm and Our model has the following advantages: First, there is no any bottleneck problem when establishing a certification path. Second, its authentication path is short with no more than two entities intervened. Third, not need to inquire about a certificate revolution list. Fourth, it's easy to extend and suitable for large-scale network. But we are just getting started on understanding the performance of PKI services in complicated systems. Although our scheme is resistance to several possible attacks[2,5], but real life experiences are still needed to identify other potential attacks, and We will also examine the algorithm in more realistic environments.

9. Acknowledgement

This work was supported by the National Key Basic Research and Development Project (973) of China (No: 2003CB314805).

10. References

[1] Meiyuan Zhao, Sean W. Smith. Modeling and Evaluation of Certification Path Discovery in the Emerging Global PKI. EuroPKI 2006, LNCS 4043, pp. 16–30, 2006.

[2] Hongjun Liu, Ping Luo, Zhifeng Zeng. A structured hierarchical P2P model based on a rigorous binary tree code algorithm. April 2006 Elsevier.

[3] André Ãrnes, Mike Just, Steve Lloyd, and Henk Meijer. Certificate Revocation Performance Simulations. project paper, June 2000.

[4] Yassir Elley, Anne Anderson, Steve Hanna, Sean Mullan, Radia Perlman, and Seth Proctor. Building Certification Paths: Forward vs. Reverse. In The 10th Annual Network and Distributed Systems Security Symposium (NDSS'01), February 2001.

[5] K.Aberer, A.Datta and M.Hauswirth. "A decentralized public key infrastructure for customer-to-customer ecommerce",International Journal of Business Process Integration and Management, Vol. 10, No. 10, 2004 .

[6] Loren M. Kohnfelder. Toward a Practical Public-Key Cryptosystem. bachelor's thesis, Dept.Electrical Engineering, MIT, Cambridge, Mass. 1978.

[7] Jose L. Mu˜noz et al. CERVANTES—A Certificate Validation Test-Bed. In First European PKI Workshop: Research and Applications (EuroPKI 2004), volume 3093 of LNCS, pages 28–42, Samos Island, Greece, June 2004. Springer-Verlag.

[8] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 Public Key Infrastructure Certificate and CRL Profile. RFC3280, http://www.ietf.org /rfc3280.txt, April 2002.

[9] M. Myers et al. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol. RFC2560, http://www.ietf.org/rfc/rfc2560.txt, June 1999.

[10] M. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol (v3). RFC2551, http://www.ietf.org/rfc/rfc2551.txt, March 1997.

[11] Selwyn Russell, Ed Dawson, Eiji Okamoto, and Javier Lopez. Virtual Certificates and Synthetic Certificates: New Paradigms for Improving Public Key Validation. Elsevier Computer Communications, 26:1826–1838, 2003.

[12] Steve Lloyd. Understanding Certification Path Construction. PKI Forum White Paper, September 2002.

[13] I.Stoica et al. Chord: A scalable peer-to-peer lookup service for internet applications, In Proceedings of the ACM SIGCOMM '01 Conference (SanDiego, California, August 2001).
[14] Thomas wlöfl. Public-Key-Infrastructure Based on a Peer-to-Peer Network. IEEE Proceedings of the 38th Hawaii International Conference on System Sciences – 2005.
[15] Neil Daswani. Denial-of-Service (DoS) Attacks and Commerce Infrastructure In Peer-to-Peer Networks. PhD Thesis, Department of Computer Science, MIT, 2005.
[16] C. Adams and S. Lloyd, Understanding PKI, 2nd ed., Addison-Wesley, Boston, 2003.

Authors

Zhiwei Gao

is a Ph.D. student in the Department of Computer Science at Beijing Institute of Technology, Beijing, China. He is an associate professor at the Department of Computer Science, ShiJiaZhuang Railway Institute. His current research interests are in network security, distributed computing and peer-to-peer systems.

Jinsheng Fan

received his Bachelor degree from Shijiazhuang Railway Soldier Institute, China, in 1979, and his refresher Master lesson was finished at Department of Computer Science, Beijing Institute Technology University, China, in 1985. He is a professor at the Department of Computer Science, Shijiazhuang Railway Institute. His research interests are computer arithmetic, computer network and its application, secure Ecommerce, distributed database systems.

Yufeng Jia

is a lecture at the Department of Computer Science, ShiJiaZhuang Railway Institute. His current research interests are in e-commerce, distributed computing and peer-to-peer systems. He received his master's degree in computer science from ShiJiaZhuang Railway Institute, China.

Li Zhang

received her Master degree from school of information and computer technology, Beijing Jiao Tong University. Her research interests include distributed and Internet systems, network security.