Research Journal of Applied Sciences, Engineering and Technology 5(18): 4444-4448, 2013
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: July 27, 2012 Accepted: September 03, 2012 Published: May 05, 2013

A Certificate-Path Construction Algorithm Based on CA Quantification

Xin Wang, Limin Cheng, Hua Jiang and Jianming Liu
School of Computer Science and Engineering,
Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

Abstract: Certificate-path construction is a procedure to generate logical links between PKI users. In this study, we designed a value to quantify a given CA (Certificate Authority) and the value is used as a reference in path constructing process. Then a reverse certificate-path construction algorithm is described. The affections of this quantification are discussed by applying the new algorithm in selected PKI structures. After all, the result of those experiments indicates the new algorithm have improvements compares with the breadth-first search algorithm.

Keywords: CA quantification, certificate-path construction, hybrid trust model, PKI

INTRODUCTION

Any digital certificate must be verified by a procedure before it is being use in a Public Key Infrastructure (PKI). The first step of this procedure is to generate a logical chain, lies between source certificate and target certificate. This chain is known as certificate-path and the procedure is named certificate-path construction (Steve *et al*., 2002). Better performance of a PKI system can be achieved through improving the efficiency of the certificate path construction.

Breadth-First Search (BFS) algorithm is commonly used in the PKI certificate path construction. It uses the queue, which following the first-in-first-out principle, as a storage structure. In order to improve the flexibility in constructing process, this study attempts to quantify the CA and provide a value as the reference that would be used by the constructing process.

PKI STRUCTURES AND CA QUANTIFICATION

PKI structures: A PKI system structure is depending on how CAs has been organized. CAs will issue certificates both to the end users and other CAs. Hierarchical, mesh and hybrid trust models are three commonly used structures (Cooper *et al*., 2005). Within an enterprises or functional community, which always have a centre-subordinate structure, the hierarchical trust model is wildly deployed and the hierarchical entity often been considered as an independent trust realm. Mesh structure can be build by issuing CA certificates between CAs, the process is called cross-certification and these CAs have no hierarchical features. If multiple hierarchical PKI realms want to combine to form a larger system without having a root CA at top of all realms' own root, then a hybrid structure can be used. A four-branch hybrid model is shown in Fig. 1. In that hybrid system CA_1 and CA_4 of the branch is a strict hierarchical, CA_2 and CA_3 have no subordinates.

Quality values of CA: If the algorithm wants to improve the efficiency of the certificate path construction, which will require more relevant information of the PKI system as references. This study introduces a way to quantify the CA nodes and the result is called Quality value. The detail of how to calculate this value will be discussed in the following.

Supposed a PKI system has d nodes, which are $\{CA_1, CA_2$ and $CA_3 \ldots CA_d\}$, the amount of terminal certificates issued by these CAs are $\{n_1, n_2, n_3 \ldots n_d\}$ respectively, the total amount of terminal certificates is N. If the probability of each terminal certificates becomes the target certificate is equals to 1/N. According to the classical probability model, the probability of each CA becoming the target is as following expression:

$$p_i = \sum_{i=1}^{n_i} \frac{1}{N} = \frac{n_i}{N}, i = 1, 2, 3 \ldots d \quad (1)$$

Obviously, for those branches which own more terminal certificates, they get higher probability that would contain the target CA. The new algorithm is about to use this characteristic through "Quality value".

In a PKI structure, there are two classes of CA nodes: CAs has subordinates (CA_1, for CA_4, $CA_{4.2}$, in Fig. 1) and those don not (CA_2, $CA_{1.2}$, $CA_{4.2.1}$, etc., in Fig. 1). So there are 2 ways of calculating the Quality value:

- **The first class of CA nodes:** Assuming the PKI system has a total of N terminal certificates, the

Corresponding Author: Xin Wang, School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, Guangxi, China

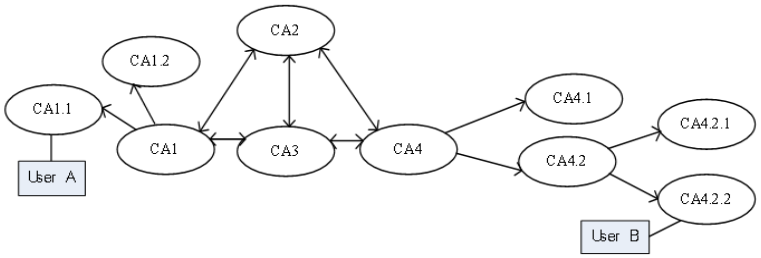

Fig. 1: A hybrid structure

CA_i is a first-class node, which has issued n_i terminal certificates. Let α_i be a suitable coefficient, the CA_i Quality could be calculated as expression:

$$Quality = \alpha_i \frac{n_i}{N} \quad (2)$$

- **The second class CA:** Suppose CA_i is a second class node that contains s direct subordinates. Each one of them has Quality value $\{Q_1, Q_2, Q_3 \ldots Q_s\}$, the number of CA's own terminal certificates is m_i, let α_i be a suitable coefficient, the Quality value could be described as expression:

$$Quality = \alpha_i(\frac{m_i}{N} + \sum_{j=1}^{s} Qj) \quad (3)$$

For example, in Fig. 1, if there are 5000 terminals certificates the PKI system, in which $CA_{1.1}$ issued 100, then its Quality would be: $(100 \div 5000) \times \alpha_{1.1} = 0.02\alpha_{1.1}$. If $CA_{1.1}$ has Quality value: $0.02\alpha_{1.1}$ and $CA_{1.2}$'s value is $0.07\alpha_{1.2}$, CA_1 owns 50 terminal certificates, then CA_1's Quality would be: Quality = $\alpha_1 (50/5000 + 0.02\alpha_{1.1} + 0.07\alpha_{1.2}) = \alpha_1 (0.01 + 0.02\alpha_{1.1} + 0.07\alpha_{1.2})$.

A system in Fig. 2 has six CAs, those CAs are divided into four branches. The amount of terminal certificates correspond to each CA is n_i. For those 1st-class nodes: A, B, C, D1, D2, their Quality value is:

$$Quality = \alpha i \frac{ni}{N}$$

Since expression (1), then Quality becomes:

$$Quality = \alpha_i \frac{n_i}{N} = \alpha_i p_i$$

When α_i take a real number, which is greater than 0, the 1st-class CA Quality value is proportional to p_i. The p_i responses the probability of the one being a target.

If a second-class node CA_i has issued m_i terminal certificates and contains s first-class subordinates, the subordinates have Quality value as $\{Q_1, Q_2, Q_3 \ldots Qs\}$. CA_i's Quality value would be:

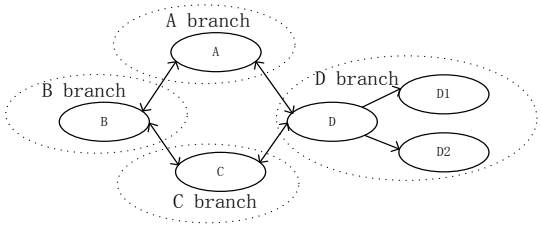

Fig. 2: Six CA's hybrid structure

$$Quality = \alpha_i(\frac{m_i}{N} + \sum_{j=1}^{s} Qj) = \alpha_i(\frac{m_i}{N} + \sum_{j=1}^{s} \alpha_j \frac{n_j}{N})$$

where,

α_i, α_j : Coefficients to CA_i and subordinates respectively

n_j : For the amount of CA_i subordinates owned terminal certificates

According to expression (1), let $p_i = m_i/N$ become the probability that CA_i become the target, $P_j = n_j/N$ responses for the probability of those subordinates containing the target, then CA_i's Quality expression would become:

$$Quality = \alpha_i(p_i + \sum_{j=1}^{s} \alpha_j p_j)$$

That means Quality value of the second-class CA is proportional to the probability of the one itself or those CAs within the same branch to become a target.

ALOGOTITHM IMPLEMENTATION

CA structure and search lists: Through the CA's certificate directory services (Housley *et al*., 2002), CA can deliver the certificates to users. In order to make the process can be carried out, during the path construction, PKI information is dynamically gathering by accessing those certificate services. Two lists are used to save the information for search. These lists are:

- Clues List for saving CAs as clues which will be used in subsequent searches. In this list, CAs are

sorted with Quality descending order, Quality value of one can be accessed from the CA certificate-service server

- Vst List for saving those CAs that has been tested

Considering the search process will involve inserting and deleting frequently then Clues List could be implemented with double-linked list data structure. Vst List has insert operations only, so it can be built in single linked list.

Algorithm pseudo-codes: In the following, a reverse certification path construction algorithm will be described. The so-called "reverse" refers to the starting point of the constructing process is at trust anchor and the path will be finished on target certificate issuer (target CA), otherwise known as "forward". The reason of using reverse direction is discussed in the literature (Yassir *et al*., 2001) which says the reverse certification path construction often works more effectively. In the following discussion, Source on behalf of the trust anchor, Target represents the target, Current on behalf of the currently standing point, Neighbor on behalf of the neighbor nodes relative to the Current.

The CA node structure in this algorithm can be designed as follows:

```
Struct CANode
{
char * CAID;
int Quality;
char * PreVst;
}
```

In which the CAID usually uses CA's distinguished name; Pre Vst field is used to indicate which one has visited this CA. Pseudo-codes of the algorithm are given as following:

```
Initiation: Clues List, V t List;
            Clues List.Add (Source);
            Vst List.Add (Source);
```

Step 1:

```
while (true): {
if (Clues List.Empty () == FALSE):
Current = Clues List.Pop_Front ();
else: break;
for (All Neighbor of Current):
if (Neighbor == Target):
Vst List.Add (Neighbor);
Break;
else if (Neighbor Not In Vst List):
        Neighbor:: CAID = Neighbor's ID;
        Neighbor:: Quality = Neighbor's Quality;
        Neighbor:: PreVst = Current;
        Clues List.Add (Neighbor);
        Vst List.Add (Neighbor);
        }
```

Table 1: CA's quality values

CA's DN	Terminal certificates	Quality
$CA_{1.1}$	100	0.10
$CA_{1.2}$	150	0.15
CA_1	50	0.30
CA_2	50	0.05
CA_3	150	0.15
$CA_{4.1}$	100	0.10
$CA_{4.2.1}$	100	0.10
$CA_{4.2.2}$	200	0.20
$CA_{4.2}$	0	0.30
CA_4	100	0.50

Table 2: Contents in clues list during path construction

	Current	Clues list
Init	NULL	$CA_{1.1}$ (0.1)
1	$CA_{1.1}$	CA_1 (0.3)
2	CA_1	$CA_{1.2}$ (0.15), CA_3 (0.15), CA_2 (0.05)
3	$CA_{1.2}$	CA_3 (0.15), CA_2 (0.05)
4	CA_3	CA_4 (0.5), CA_2 (0.05)
5	CA_4	$CA_{4.2}$ (0.3), $CA_{4.1}$ (0.1), CA_2 (0.05)
6	$CA_{4.2}$	$CA_{4.2.1}$ (0.1), $CA_{4.1}$ (0.1), CA_2 (0.05)

Step 2:

```
If (Target Is In Vst List):
Out put the path by chasing the CANode::Pre Vst;
else: Path construction failed;
```

Suppose user A in Fig. 1 needs to verify the user B's terminal certificate. User A's certificate is issued by the $CA_{1.1}$, user B's is issued by the $CA_{4.2.2.}$ Algorithm needs to find a certificate-path between $CA_{1.1}$ and $CA_{4.2.2.}$ Then $CA_{4.2.2}$ is called the target CA. In practical term, the trust anchor can be flexibly selected, in order to facilitate elaborated, user A's terminal certificate direct issuer $CA_{1.1}$ is selected as a trust anchor. Taking the coefficient α_i of each CA's equal 1, the system contains a total of 1,000 terminal certificates. The amount of terminal certificates and Quality value of each CA nodes are shown in Table 1 and 2 lists the contents in Clues List during constructing process. The result certificate path would be: $CA_{1.1}$->CA_1->CA_3->CA_4->$CA_{4.2}$->$CA_{4.2.2}$.

RESULT ANALYSIS

A problem needs to be concerned during certificate path construction process is the loop detection. In new algorithm, by setting Vst List for saving visited CAs and set a loop testing before each CA being added into the Clues List, the result is loop less.

As the algorithm informs that all of the known CAs will be added to the Clues List, then one of the conditions to break the constructing process is Clues List being empty. That means: if a CA system is connected or the target belonging to the same connected

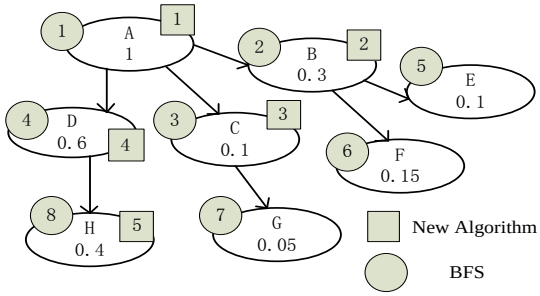

Fig. 3: Construction comparison

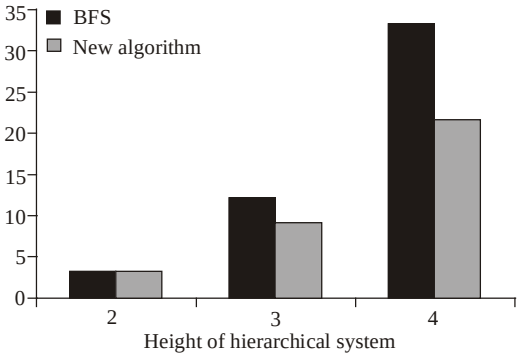

Fig. 4: Experiment result on hierarchical structure

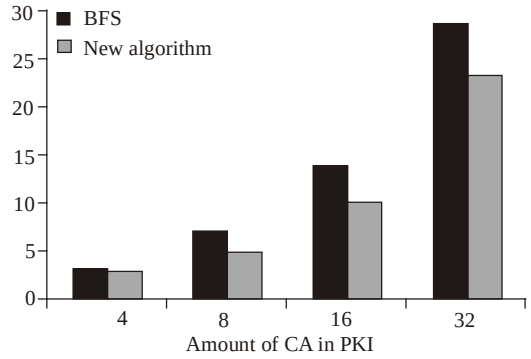

Fig. 5: Experiment result on mesh structure

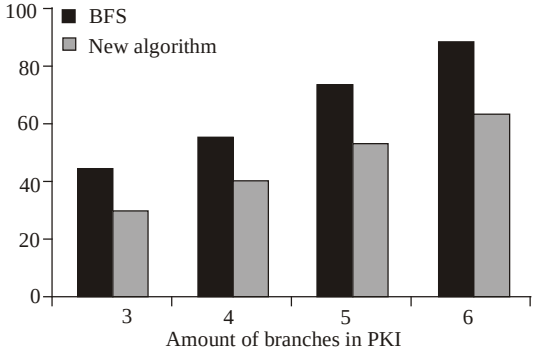

Fig. 6: Experiment result on hybrid structure

component with trust anchor, the algorithm will be able to construct a certification path finally.

CA's α_i greater than 0: In breadth-first algorithm, the test sequence between CAs is depending on the order while they join the search queue. In this new algorithm, the position of one in Clues List is sorted with the Quality descending order. As discussed above: the CA node with the greater Quality value has higher probability to become a target or the node in the certificate-path. In other words: for those who are more likely to become the target CA would have higher test priority. An example of these differences is shown in Fig. 3.

When using a breadth-first algorithm on the system shown in Fig. 3. A request is about to find a path from A to H. The test order of nodes would be: A->B->C-> D->E->F->G->H, a total of eight tests. After the introduction of the Quality value as a reference by new algorithm, the test sequence would be: A->B->C->D-> H, a total of five tests. The key is that the Quality value of D is greater than B's, which makes the D direction been tested earlier than the B branch. Therefore, would reduce the search process.

More importantly, all CA nodes, which being used in a constructing process, are accessed from the CA's certificate directory server via the network. In practical term, the size of data for network transmission is proportional to the amount of CA that needs to be tested. As long as the amount of test node getting greater, the workloads in a given network are higher. So, this algorithm will shorten the search process, while also reducing the burden on the network.

The following experiments apply both breadth-first algorithm and the new algorithm on the hierarchical, mesh and hybrid structure respectively. Assuming a total amount of terminal certificates is 1000 and 1000 simulated requests are generated randomly. The branch factor of the selected hierarchical structure is three; in mesh structure there are 4, 8, 16 and 32 CAs respectively; the hierarchical components (branch), which are leading by a root CA, in hybrid structure would have branch factor and height are no more than three. The average amounts of tested CAs for each simulated request are shown in Fig. 4, 5 and 6.

All CA's quality equals the same: In a particular case, if all CA's α_i takes the appropriate value to make the individual Quality value equals with each other, then all CAs in Clues List are having the same test priority. That means the test order is completely dependent on

the sequence of CA nodes joining search queue. At this time, the algorithm is equivalent to a breadth-first algorithm.

Customize the search priority: In the discussion above, it is assumed that each terminal certificate has the probability 1/N to become a target certificate. However, due to business expansion or security policy, the user contained in some branches of the PKI system may show more active than the others. That means those terminal Certificate issuer (CA) within these active branch will get a higher possibility involving in path construct requests. In this situation, through modifying the α_i values in particular CAs would customize the CA's search priority and then make the PKI works better.

CONCLUSION

The efficiency of the certificate path construction is an important factor in the use of the PKI system. In order to improve the performance of certificate path construction, a way to quantify a CA is discussed. Through taking an appropriated α_i, the algorithm shows more efficiency and flexible than breath-first algorithm. In the future, further research can focus on:

- How to depict a CA node with more related factors, such as the status of the network or using certificate information caching policy (Takahiro *et al*., 2010).
- Design a Quality value for the bridge structure environment.
- Study the performances of this Quality value based algorithm on other kind of trust structure (Zhiwei *et al*., 2007).

ACKNOWLEDGMENT

This study is supported by Guangxi Key Laboratory of Trusted Software (Guilin University of Electronic Technology).

REFERENCES

Cooper, M., Y. Dzambasow, P. Hesse, S. Joseph and R. Nicholas 2005. Internet X.509 public key infrastructure Certification Path Building. Retrieved from: http:// tools.ietf.org/rfc/rfc4158.txt

Housley, R., W. Polk, W. Ford and D. Solo, 2002. nternet X.509 public key infrastructure Certificate and Certificate Revocation List (CRL) Profile. Retrived From: http://www.ietf.org/rfc/rfc3280.txt

Steve, L., A. Nash R. Housley, J. Linn and N. Magnus, 2002. Understanding Certificate Path Construction. Retrived from: http:// www. oasispki. org/ pdfs/ Understanding_Path_construction-DS2.pdf.

Takahiro, F., A. Sato, Y. Kumagai, T. Kaji and K. Okada, 2010. Development of hi-speed X.509 certification path validation system. IEEE 24th International Conference on Advanced Information Networking and Application Workshop, pp: 269-274.

Yassir, E., A. Anne, S. Hanna, S. Mullan, R. Perlman, S. Proctor, 2001. Building Certification Path: Forward vs Reverse. Retrieved From: www. isoc. org/isoc/conferences/ndss/01/papers/elley.pdf

Zhiwei, G., L. Ping, G. Zhimin and L. Hongjun, 2007. A new and scalable certification path discovery model in the emerging global PKI. IEEE International Conference on Multimedia and Ubiquitous Egineering 2007.