Matrix Powers algorithm for trust evaluation in PKI architectures

Jean-Guillaume Dumas, Hicham Hossayni

Matrix Powers algorithm for trust evaluation in PKI architectures

Jean-Guillaume Dumas* Hicham Hossayni*

July 8, 2011

Abstract

Public-Key Infrastructures (PKI) are considered as the safeguard of the security of communications on large scales networks like Internet. Different trust models have been proposed to interconnect the various PKIs components in order to propagate the trust between them. This paper provides a simple model for trust and reputation management in PKI architectures, and a new algorithm to assess trust relationships in a network using different trust evaluation schemes.

1 Introduction

Public Keys Infrastructures (PKI) are the keys of trust. The principle of a PKI is to establish (using certificates) a trust environment between network entities and thus guaranty the security of communications.

A number of PKI trust models have been proposed, among them one can cite hierarchical, cross-certification, mesh, PGP, Trust Lists PKI [23], see e.g. [19, 18] and references therein.

For example in a cross-certification PKI, an entity called Alice can establish a communication with another entity called Bob only after validating the Bob's certificate. For this, Alice must verify the existence of a certification path between its trust anchor and Bob's certification authority (CA). This certificate validation policy impose that each entity must have a complete trust in its trust anchor, and that this trust anchor has a direct or indirect relation with the other entity CA.

In fact, several risks exist in this current trust models. Ellison and Schneier identified the major risk of PKIs to be "Who do we trust, and for what?" which emphasizes the doubts about the trust relationship between the different PKI components [4]. The incident in which VeriSign issued to an fraudulent two certificates associated with Microsoft [8], confirms these doubts, and asks questions about the validity of certificates issued by different certificate authorities.

*Laboratoire J. Kuntzmann, Université de Grenoble. 51, rue des Mathématiques, umr CNRS 5224, bp 53X, F38041 Grenoble, France, Jean-Guillaume.Dumas@imag.fr, Hossayni.Hicham@gmail.com.

Overall, this reveals the risk of an imprecise use of the word 'trust'. To avoid this particular issue, a solution is the precise and rigorously specified use of trust in a trust model. In this context, several studies [1, 9, 10, 11, 14] analyzed, represented and quantified trust in different areas (PKI architectures, P2P environments, social networks, decentralized systems, ...).

In these studies, the focus is on the trust transitivity property and they propose algorithms to quantify the trust relationship between two entities in a network. Some of them evaluate trust throughout a single path, while others consider more than one path to give a best approximation of trust between those entities. However to the best of our knowledge they were restricted to simple networks trees and not to generic graphs.

In this paper, we propose to use the powers of the adjacency matrix (used e.g. to verify the graph connexity or to compute the number of [bounded] paths between nodes [21]). The approach is similar to that used also e.g. for community detection in graphs [6] and we use it to produce a centralized or distributed quantification of trust in a network. The complexity of this algorithm is $O(k.\varphi.\ n)$ in the worst case, polynomial in $n =$, the number of entities (nodes of the graph), φ : the number of trust relationships (vertexes), and k the size of the longest minimal path between entities. For instance the algorithm proposed in [10] required the approximate resolution of the Bounded Disjoint Paths problem, known to be NB-Hard [22].

The aim of our algorithm is the evaluation of trust using all existing (bounded) trust paths between entities as a preliminary to any exchanges between PKIs. This can give a precise evaluation of trust, and optimize the certificate validation time. The algorithm can also be adapted (under condition) to different trust metrics.

We present different trust metrics in section 2 and our algorithm in section 3. Then, we show that there is a complementarity between trust and reputation and propose a simple model for the trust management in PKI architectures, using both notions (trust & reputation). This model is for instance adapted to the cross-certification model and to PGP web of trust.

2 Measurement of trust

2.1 Concept of trust

Several definitions of trust have proposed, we here choose that of [11] for its genericity: *Trust is the psychological state comprising*:

1. ***expectancy***: *the trustor expects a specific behavior of the trustee such as providing valid information or effectively performing cooperative actions*;
2. ***belief***: *the trustor believes that the expected behavior occurs, based on the evidence of the trustee's competence and goodwill*;
3. ***willingness to be vulnerable***: *the trustor is willing to be vulnerable to*

that belief in a specific context, where the specific behavior of the trustee is expected.

According to the types of the expectancy in trust, there are two types of trust:

1. trust in performance: the trustor a trusts trustee b regarding b's performance(represented by x). It means that if information x is **produced** by entity b in context k, then entity a believes x in the same context k.

2. trust in belief: the trustor a trusts trustee b regarding b's belief (represented by x) in context k. It means that if information x is **believed** by entity b, then entity a believes also x in the same context k.

2.2 Transitive trust evaluation schemes

There are several schemes for evaluating the (transitive) trust in a network. Some presents the trust degree as a single value representing the probability that the expected action will happen [1, 25]. Others include the *distrust* degree indicating the probability that the opposite of the expected action will happen [9]. More complete schemes are introduced which include more parameters to evaluate trust. Jøsang [12, 14, 15] introduced the Subjective Logic notion which expresses subjective beliefs about the truth of propositions with degrees of "uncertainty". This notion is used in [13] to evaluate trust in certification chains. [10, 11] introduced a quite similar scheme with a formal, semantics based, calculus of trust and applied it to public key infrastructures (PKI). We will see next that a monoid structure, as in the the model of [11] for trust evaluation, is essential for our scheme.

2.3 Evaluation of trust

[11] represents the trust relationship as a triplet: (trust degree, distrust degree, uncertainty), where:

- **The trust degree** is the frequency rate of the trustor's positive experience among all encounters with the trustee. That is,

$$td^t(d, e, x, k) = \frac{n}{m}$$

  where m is the total number of encounters regarding an instanced expectancy x, n is the number of trustor's positive experience and t is the trust type (b for belief, and p for performance).

- **The distrust degree**: similarly we have

$$dtd^t(d, e, x, k) = \frac{l}{m}$$

  where l is the number of trustor's negative experiences.

- The uncertainty: denoted by ud is defined by:

$$ud(d,\ e,\ x,\ k) = 1 - td(d,\ e,\ x,\ k) - dtd(d,\ e,\ x,\ k).$$

In the following we will denote the trust relationship by $tr(a,\ b,\ x,\ k) = < td(a,\ b,\ x,\ k),\ dtd(a,\ b,\ x,\ k) >$

2.4 Propagation of trust

2.4.1 Sequential propagation

Theorem 1. *[10, Theorem UT-1] Assume that the trustor a trusts b's believes with:* $tr(a,b,x,k) =< td^b, dtd^b >$ *and b trusts c's performance with:* $tr(b,c,x,k) =< td^p, dtd^p >$*, then the trust relationship from a to c can be derived as follows:* $tr(a,c,x,k) =< td^p, dtd^p >$*, with:*

$$td^p(a,c,x,k) = td^b(a,b,x,k).td^p(b,c,x,k)$$

$$+dtd^b(a,b,x,k).dtd^b(b,c,x,k)$$

and

$$dtd^p(a,c,x,k) = dtd^b(a,b,x,k).td^p(b,c,x,k)$$

$$+td^b(a,b,x,k).dtd^b(b,c,x,k)$$

2.4.2 parallel propagation

Theorem 2. *[11, § 7.2.2] Assume that entity a trusts (directly or indirectly) entities* $b_1,\ \ldots\ , b_n$ *with a certain degree and that entities* $b_1,\ \ldots\ , b_n$ *trust the entity c with some degree (a may also have a direct trust to c), the aggregation of trust from a to the entity c is:*

$tr_g\,(a,\ c) = < td_g\,(a,\ c)\,,\ dtd_g\,(a,\ c) >$ *with:*

$$td_g\,(a,\ c) = 1 - \prod_{i=1..n} (1 - td_i).$$

$$dtd_g\,(a,\ c) = \prod_{i=1..n} dtd_i.$$

$$ud_g\,(a,\ c) = 1 - td_g\,(a,\ c) - dtd_g\,(a,\ c)\,.$$

where $< td_i\ ,\ dtd_i\ ,\ ud_i >$ *is the sequential aggregation of trust degree between a and c throughout the path* $a \to \cdots \to b_i \to \cdots \to c$ *.*

2.4.3 Algorithm for evaluating trust between two entities

Let TN be the trust graph of a network, i.e. the graph representing trust interactions between the network's entities. [10] proposes algorithm 1 for evaluating trust between two nodes in a DAG (directed acyclic graph).

Algorithm 1 [10, §6.3] Graph Aggregation

Input A, Z two nodes of graph TN.
Output Trust between A and Z

```
1: If edge (A, Z) is the only path from A to Z in TN then
2:   return (A,Z) edge weight.
3: else
4:   If A has and only has one path to Z then
5:     use sequence aggregation to aggregate;
6:     remove the last edge in this path to Z;
7:     add edge (A, Z) labeled by td*(A, Z) in TN;
8:   else
9:     If A has multiple disjoint paths to Z then
10:      use parallel aggregation to aggregate all paths from A to Z;
11:      remove the last edge in each path to Z;
12:      add edge (A, Z) labeled by td*(A, Z) in TN;
13:    else
14:      calculate N = neighbors(Z);
15:      For all n_i ≠ A in N do
16:        aggregate(A, n_i, TN);
17:      End For
18:      use parallel aggregation to aggregate all paths from A to Z;
19:      remove the last edge in each path to Z;
20:      add edge (A, Z) labeled by td*(A, Z) in TN;
21:    End If
22:  End If
23:  return td*(A, Z).
24: End If
```

3 Matrix Powers algorithm

In this section, we propose a new algorithm for evaluating trust in a network using the powers of the matrix of trust. This algorithm is uses techniques from graph connexity [21] and communicability in networks [6, 5, 7], via the exponential of a matrix. Our matrix powers algorithm can be implemented with different trust propagation schemes under one necessary condition: the transitivity property of the (sequential & parallel) trust propagation formulas. In the following, we adopt the trust notions and the propagation formulas of [11], presented in the last section, for the sake of simplicity.

3.1 Notions

3.1.1 Matrix of trust

Definition 1. *We call "matrix of trust" (denoted C) the matrix that contains the trust degrees felt by each entity of the network towards its neighbors.* $C_{i,j} =<$

$td(i,j), dtd(i,j) >$ *is the trust degree of the entity i towards the entity j. Without any relationship between two entities a and b, we choose* $C_{a,b} =< 0.0, 0.0 >$. *Also, since every entity is fully confident in itself, we choose for all i:* $C_{i,\ i} = < 1.0,\ 0.0 >$.

3.1.2 Monoids of trust

Definition 2. *Let G be the set* $G = \{(x,y) \in [0,1], x+y=1\}$, *equipped with two operations "+" and "." such that* $\forall (a,b),(c,d) \in G$ *we have:*

$$(a,b).(c,d) = (ac\ + bd,\ ad + bc\ ),$$

and

$$(a,b) + (c,d) = (1-(1-a)(1-c),\ bd).$$

We define as the monoids of trust the monoids (G, +, (0.1)) and (G,., (1.0)).

Remark 1. $(0,0)$ *is the absorbing element of the operation "." in G. This justifies a posteriori our choice of representation for the absence of a trust relationship.*

We can see that the set G corresponds to trust degrees $< td, dtd >$. In addition, the operations "." and "+" represent respectively sequential and parallel aggregations of trust. Therefore, we can deduce the following corollaries of theorems 1 and 2:

Corrolary 1 (of theorem 1)**.** *Let* $tr(a,b,x,k) =< td^b, dtd^b b >\in G$ *be the trust degree from a to b and* $tr(b,c,x,k) =< td^p, dtd^p >\in G$ *the trust degree from b to c. Then we can derive the trust degree from a to c as:*

$$tr(a,c,x,k) = tr(a,b,x,k).tr(b,c,x,k)$$

Corrolary 2 (of theoorem 2)**.** *Assume that we have two entities a and c with n intermediary nodes* $b_i$for$i \in \{1..n\}$ *and let* $tr_i(a,c,x,k) \in G$ *be the trust degree from a to c deduced from the sequential aggregation on the path through the node* b_i. *Then, the global trust degree derived from the parallel aggregation of* $tr_i(a,c,x,k) \in G, \forall i \in \{1..n\}$ *is*

$$tr(a,c,x,k) = tr_1(a,c,x,k) + tr_2(a,c,x,k) + \cdots + tr_n(a,c,x,k)$$

$$= \sum_{\text{i=1..n}} tr_i(a,c,x,k)$$

3.2 Evaluating trust using all paths of length lower than 2

Definition 3. *Consider the graph* $TN = (E,A)$, *representing the interactions (in terms of trust) between the entities of a network, where E is the set of nodes (entities), and A is the set of edges (relations). Let C be the trust matrix of this*

graph and consider the entities $i, j \in E$. Let $\overrightarrow{u}$ be the row vector $\overrightarrow{C_{i*}}$, and $\overrightarrow{v}$ the column vector $\overrightarrow{C_{*j}}$. We define the Cartesian product of $\overrightarrow{u}$ and $\overrightarrow{v}$ in the set G to be:

$$\overrightarrow{u}.\overrightarrow{v} = C_{i1}.C_{1j} + C_{i2}.C_{2j} + \cdots + C_{i,j-1}.C_{j-1,j} +$$

$$C_{i,j+1}.C_{j+1,j} + \cdots + C_{in}.C_{nj} = \sum_{k \in E}^{k \neq j} C_{ik}.C_{kj}$$

Lemma 1. *The Cartesian product $\overrightarrow{u}.\overrightarrow{v} = \overrightarrow{C_{i*}}.\overrightarrow{C_{*j}}$ is the parallel aggregation of all paths of length ≤ 2 connecting i to j.*

Proof. We prove first that $C_{ik}.C_{kj}$ is the sequential aggregation of trust between i and j throughout the path (of length≤ 2) $i \to k \to j$ with $k(\forall k \in \{1..n\})$:

Let k be an entity in the network

Case 1: $k = i$ or $k = j$:

* if $k = i$, then

$$C_{ik}.C_{kj} = C_{ii}.C_{ij} = (1,0).C_{ij} = C_{ij}$$

since $C_{ii} = (1,0) \forall i$, (according to the trust matrix definition) * if $k = j$, then

$$C_{ik}.C_{kj} = C_{ij}.C_{jj} = C_{ij}.(1,0) = C_{ij}$$

Therefore $C_{ik}.C_{kj}$ corresponds to the [sequential aggregation of] trust between i and j throughout the path (i, j) of length =1.

In the Cartesian product, we added the constraint $k \neq j$ in the sum to avoid taking C_{ij} twice into account.

Case 2: $k \neq i$ and $k \neq j$,

* if k belongs to a path of length = 2 connecting i to j, then: i trusts k with degree $C_{ik} \neq (0,0)$, and k trusts j with degree $C_{kj} \neq (0,0)$.

Indeed, $C_{ik}.C_{kj}$ corresponds to the sequential aggregation of trust between i and i throughout the path $i \to k \to j$.

* If there is no path of length 2 between i and j containing the node k, then we have $C_{ik} = (0,0)$ or $C_{kj} = (0,0)$, and $C_{ik}.C_{kj} = (0,0)$ is equivalent to the aggregation of trust between i and j on the path traversing the node k.

Finally, we can deduce that

$$\overrightarrow{u}.\overrightarrow{v} = C_{i*}.C_{*j} = \sum_{k \in E}^{k \neq j} C_{ik}.C_{kj}$$

corresponds to the parallel aggregation of trust between a and j using all paths of length ≤ 2

We can note that this value is equivalent to

$$\overrightarrow{u}.\overrightarrow{v} = \sum_{\substack{k \in Successors(i) \cap \\ Predecessors(j)}} C_{ik}.C_{kj} + C_{ij}$$

□

Definition 4. *Let $C_{(ij)}$ and $M_{(ij)}$ be two trust matrices. We define the matrix product $N = C * M$ by:* $\forall i, j \in \{1..n\}$

$$N_{ij} = \begin{cases} \overrightarrow{C_{i*}}.\overrightarrow{M_{*j}} = \sum_{k \in E}^{k \neq j} C_{ik}.M_{kj} & \text{if } i \neq j \\ (1,0) & \text{otherwise} \end{cases}$$

Corrolary 3. *Let (C_{ij}) be the trust matrix of a network of entities, which elements belong to the previously defined graph G. The matrix M defined by: $M = C^2 = C * C$ represent the global evaluation of trust between all entities pairs by aggregating all paths of length lower than 2.*

Proof. We have: $\forall i, j \in \{1..n\}$

$$M_{ij} = \begin{cases} \overrightarrow{C_{i*}}.\overrightarrow{C_{*j}} = \sum_{k \in E}^{k \neq j} C_{ik}.C_{kj} & \text{if } i \neq j \\ (1,0) & \text{otherwise} \end{cases}$$

If $i = j$: then $M_{ij} = (1,0) \Longleftrightarrow i$ has a total trust on itself.

Else if $i \neq j$: according to lemma 1, M_{ij} corresponds to the aggregation of trust between i and j using all paths of length ≤ 2. □

3.3 Evaluating trust using all paths of length $\leq$ n in a directed acyclic graph

We can generalize corollary 3 to evaluate trust using all paths of a given length:

Theorem 3. *Let (C_{ij}) be the trust matrix corresponding to an acyclic graph, whose elements belong to G. The matrix N defined by: $N = C^n$ represents the global evaluation of trust between all entities pairs by aggregating all path of length n.*

Proof. This theorem is proved by induction. * For $n = 2$: We have the previous result: $M = C^2$ represents the trust aggregation by using all path of length ≤ 2.

* for $n = 3$: We have

$$N = C^3 = \ C^2 * C$$

Therefore $\forall i, j \in \{1..n\}$

$$N_{ij} = \sum_{k \in E}^{k \neq j} C_{ik}^{(2)}.C_{kj} \quad \text{and} \quad N_{ii} = (1,0)$$

Consider k, a neighbor of j, ie. $C_{kj} \neq (0,0)$. We start by proving that $C_{ik}^{(2)}.C_{kj}$ is the aggregation of trust between i and j using all paths of length ≤ 3 through k.

According to the trust aggregation algorithm 1, this case corresponds to the presence of several paths that intersect at k. Now, to aggregate this trust relationship, we must apply the aggregation of trust between i and k: we then apply the sequential aggregation on the path $l_k = \{i \to k \to j\}$.

We know that $C_{ik}^{(2)}$ represents the trust aggregation using all paths of length ≤ 2 linking i to k. Thus $C_{ik}^{(2)}.C_{kj}$ is the sequential aggregation of trust on the path l_k and we have $length(l_k) \leq 3$.

Therefore, by taking in account all neighbors of j ($k \in E, k \neq j$), the value

$$N_{ij} = \sum_{k \in E}^{k \neq j} C_{ik}^{(2)}.C_{kj}$$

corresponds to the trust aggregation between i and j using all paths of length ≤ 3.

We have shown the correctness of the theorem for $n = 3$. Similarly, $C_{ij}^{(3)}$ may be used for $n = 4$ to evaluate the trust degree between i and j's trusted entities.

We can deduce in the same manner the correctness of the theorem for all $n > 3$: Suppose that C^n is the trust aggregation using all paths of length $\leq n$ and let

$$N = C^{n+1} = \ C^n * C$$

Then $\forall i, j \in \{1..n\}$, we have

$$N_{ij} = \sum_{k \in E}^{k \neq j} C_{ik}^{(n)}.C_{kj} \quad \text{and} \quad N_{ii} = (1, 0)$$

The term $C_{ik}^{(n)}$ represents the trust aggregation between i and k using all paths of length $\leq n$. Now, since C_{kj} is the direct trust degree of k to j, we have that the term $C_{ik}^{(n)}.C_{kj}$ (and consequently N_{ij}) designates the trust aggregation between i and j using all paths of length $\leq n + 1$ traversing node k in n^{th} position. □

Practical experiments show that this matrix powers algorithm seems to converge to a matrix C^N, where N is roughly the size of the longest path in the trust graph. This behavior is also encountered, and proved, in the next section algorithm, where we take also into account the presence of cycles in the network.

3.4 Evaluation of trust in the presence of cycles

In the presence of cycles in a network, the matrix powers algorithm reevaluates indefinitely the trust degrees between the nodes in a cycle. This implies that the algorithm will converge finally to the maximal trust degree 1.

Consider the example of figure 1.

1	a	0	0
0	1	b	0
0	0	1	c
0	d	0	1

Figure 2: Trust matrix C

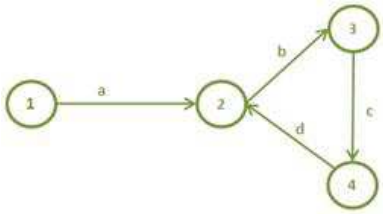

Figure 1: graph with one cycle

1	a	a.b	0
0	1	b	b.c
0	c.d	1	c
0	d	d.b	1

Figure 3: Trust matrix C^2

1	a	a.b	a.b.c
0	1	b	b.c
0	c.d	1	c
0	d	d.b	1

Figure 4: Trust matrix C^3

1	a+ a.b.c.d	a.b	a.b.c
0	1	b	b.c
0	c.d	1	c
0	d	d.b	1

Figure 5: Trust matrix C^4

1	a.b.c.d + a	a.b+a.b.c.d.b	a.b.c
0	1	b	b.c
0	c.d	1	c
0	d	d.b	1

Figure 6: Trust matrix C^5

C^5				R			
1	a.b.c.d + a	a.b + a.b.c.d.b	a.b.c	-	1,2,3,4	1,2,3	1,2,3,4
0	1	b	b.c	-	-	2,3	2,3,4
0	c.d	1	c	-	2,4	-	3,4
0	d	d.b	1	-	2,4	2,3,4	-

Figure 10: Trust matrix C^5

C^2				R			
1	a	a.b	0	-	1,2	1,2,3	-
0	1	b	b.c	-	-	2,3	2,3,4
0	c.d	1	c	-	2,4	-	3,4
0	d	d.b	1	-	2,4	2,3,4	-

Figure 7: Trust matrix C^2

C^3				R			
1	a	a.b	a.b.c	-	1,2	1,2,3	1,2,3,4
0	1	b	b.c	-	-	2,3	2,3,4
0	c.d	1	c	-	2,4	-	3,4
0	d	d.b	1	-	2,4	2,3,4	-

Figure 8: Trust matrix C^3

C^4				R			
1	a+ a.b.c.d	a.b	a.b.c	-	1,2,3,4	1,2,3	1,2,3,4
0	1	b	b.c	-	-	2,3	2,3,4
0	c.d	1	c	-	2,4	-	3,4
0	d	d.b	1	-	2,4	2,3,4	-

Figure 9: Trust matrix C^4

Consider the graph of figure 1, with a, b, c, d the trust degrees corresponding to the links $1 \to 2,\ 2 \to 3,\ 3 \to 4,\ 4 \to 2$. Its trust matrix C is given by figure 2. By applying the matrix powers algorithm on this matrix, we obtain the results shown by the figures 3, 4, 5 and 6.

For instance, the value $C^5_{1,3} = a.b + a.b.c.d.b$ corresponds to the trust aggregation on the paths $1 \to\ 2 \to\ 3$ and $1 \to\ 2 \to\ 3 \to\ 4 \to\ 2 \to\ 3$ linking 1 to 3. If we continue iterations for $n > 5$, we find that the algorithm re-evaluates the trust on the loop $3 \to\ 4 \to\ 2 \to\ 3$ infinitely.

To solve this issue, we change the matrix multiplication procedure, so that each node will be used only once in the assessment of a trust relationship. For this, we use a memory matrix R_{ij}. This stores, for each pair of nodes, all nodes traversed to evaluate their trust degree.

The computation of $C_{ij}^{(n+1)}$ for $n \geq 1$, becomes:

Algorithm 2 Matrix powers for generic network graphs

Input An $n \times n$ matrix of trust C.
Output Global trust in the network.

1: $R_{ij} = \{\emptyset\}$.
2: **While** $R \neq [1..n] \times [1..n]$ **do**
3: $\quad C_{ij}{}^{(n+1)} = \sum_{k \in E}^{k \neq j,\ k \notin R_{ij}} C_{ik}^{(n)}.C_{kj}$
4: $\quad R_{ij} = R_{ij} \cup \{ R_{ik} \cup \{k\} \ / \ k \in E,\ C_{ik}^{(n)}.C_{kj} \neq < 0, 0 > \}$
5: $\quad If\ n > 2\ then\ C_{ij}{}^{(n+1)} = C_{ij}{}^{(n+1)} + C_{ij}{}^{(n)}$
6: **End While**

First, we initialize R_{ij} with the empty set. Then, at each iteration, we add to R_{ij} all the nodes used to aggregate the trust between i and j.

To compute $C_{ij}^{(n+1)}$, only the nodes that have not been used in previous iterations are be considered. Consequently, at each iteration, the aggregated paths are completely independent from those of the previous iterations. This is why we apply the parallel aggregation between the old and new found paths in the last step of the iteration.

By applying the new algorithm on our example, we now obtain the results shown by the figures 7, 8, 9 and 10.

3.5 Cycling evaluation of trust

In the practical case, the evaluation of trust between two nodes A and B need not consider all trust paths connecting A to B for two reasons:

1. First, the mitigation is one of the trust properties, ie. the trust throughout trust paths decreases with the length of the latter. Therefore after a certain length L, the trust on paths becomes weak and thus should have a low contribution in improving the trust degree after their parallel aggregation.

2. Second, if at some iteration $n \geq 1$, we already obtained a high trust degree, then contributions of other paths will only be minor.

Therefore, it is possible to use the matrix powers algorithm to assess trust in a network, by limiting the number of iterations by a constant L, which is the maximum path length to take in consideration. This is bounded by the longest minimal-length path between any two nodes of the network.

4 Trust evaluation model for PKI architectures

4.1 Trust vs Reputation

The reputation can be defined as in [24]: "*a peer's belief in another peer's capabilities, honesty and reliability based on the other peers recommendations.*" Currently, the reputation is implemented in several areas: e-commerce , mailing (combating spams), search engines (Pages classification), P2P networks, ... Reputation can for instance be used to help peers distinguish *good from bad partners*.

However, most researches about trust treat separately the notions of trust and reputation. Yet these two concepts are complementary and both are necessary for a better quantification of the "credibility" of an entity on a network.

On the one hand, it is necessary to have at least one trust path between two entities to evaluate their trust degree. This cannot always be guaranteed in large networks. If this is indeed not the case in a given network, the degree of reputation gives us a significant indication that will allow users to take the decision to communicate (or not) with other peers.

On the other hand, a low degree of reputation cannot be conclusive on the credibility of an entity. Since reputation depends on the number of incoming trust relationships, this may discriminate the least "popular" entities. In this case, the trust degree is more significant.

There exist several reputation evaluation systems like EigenTrust [17], inspired from the famous Google's PageRank [20]; the Spreading Activation Model for trust propagation [26], etc. An important advantage of reputation evaluation systems is their performance. They are very fast compared to the binary trust evaluation on a network, usually in time quadratic in the size of the network.

In the following, we propose a model combining the trust and the reputation concepts, for an efficient evaluation of trust in PKI architectures: with complexity roughly cubic in the size of the network.

4.2 Centralized vs Distributed model

To implement a centralized trust management model, we would need a new entity called a "trusted authority" (TA: Trust Authority). This entity will assess trust in PKI architectures. Its role will be to retrieve the trust degrees expressed by all CAs, and to evaluate the trust and reputation degrees in the network.

The main advantage of the centralized model is that the TA may have a global vision throughout the network, allowing estimating with high accuracy the trust degrees between entities.

The reliability of this centralized model is based entirely on the reliability of the TA. This implies that all entities on the network must have "total trust" on the TA. This might not be applicable to very large network like the whole Internet.

Another approach would be to use a distributed trust management model, where each entity must contact others to share some trust degrees. This will enable each entity to evaluate the trust in its neighborhood. The main advantage of such a distributed model is that it can be applied to large networks, while preserving for each entity a relatively low computational cost.

The main drawback of this distributed model is that each entity might have only a limited view of the whole network. Therefore each trust degrees computed by an entity will only be approximations.

Another possibility is also to distribute even the computation of the trust matrix: each entity would be responsible of the computation of a sub-matrix of the global network. Then the entity could receive some other (potentially overlapping) sub-matrices, signed by trusted entities. The remaining problem here is the validation of the chosen initial trust matrix.

For all these reasons, in the following we already propose a way to handle distributed trust computations of PKIs. Trust degrees can e.g. be expressed in the certificates, even as a shared secret [3].

4.3 Model description

In this section we thus propose a trust management model for the cross-certified PKIs and PGP web of trust.

4.3.1 Expressing the trust

In general, our model is applied to a PKI system, which consists of a number of entities with certificates, and in which any entity may sign other entities' certificates. In fact, a cross certification takes place between two entities when they sign each other's certificates, as shown on figure 11.

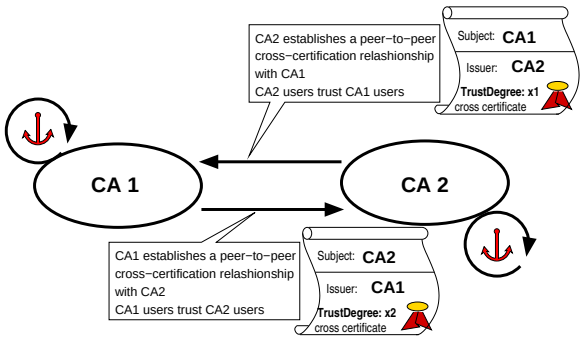

Figure 11: Peer to peer cross-certification between two CAs

This cross-certification supposes the verification of a cross-certification policy and some from the signatory to the owner of the signed certificate. Thus, each entity may express its trust degrees in the certificates it has to create and sign.

We therefore suppose that the signatures and the trust degrees must appear together on (signed) cross certificates. For this, we need to provide to the CA administrators (or PGP users) a way to express their trust degree.

This solution allows us to bypass some known vulnerabilities and avoid some attacks of trust and reputation systems: whitewashing attack, Sybil attack, impersonation and reputation theft, ballot stuffing and bad mouthing, etc. [16, 2].

Due to the difficulty of expressing precisely a trust degree a first rating could use a scale from 0 to 10 to express the triple (trust, distrust, uncertainty). As shown for example on figure 12, one can evaluate a trust degree is evaluated to 7/10, usually a small value for the known false emissions of the trustee (say 0/10 or here 1/10) and then the uncertainty is obviously deduced as $1 - trust - distrust = 1/3$.

Figure 12: Example of a scale of trust

4.3.2 Cross-certified PKI architectures

In the case of cross-certified PKI architectures, the CAs play the main role. In the context of this article, we assume that the relations $[CA \rightarrow users]$ are based on complete trust. In this case, only the inter-CAs relationships are evaluated.

Each CA creates an initial trust matrix from its certificate store and saves it locally. This matrix corresponds to the sub-graph of the CAs neighborhood. A network discovery mechanism could be established to expand the trust sub-graph and to have a broader view on the network.

The CA evaluates the trust and reputation in this matrix using the matrix powers algorithm of section 3 and a reputation evaluation scheme. Then it can also decide to forward some of this information to its users, via e.g. its SCVP service (Server-based Certificate Validation Protocol), in response to their certificate validation requests.

4.3.3 PGP Web Of Trust

In the case of PGP networks, the same rating system could replace the actual system (full trust, marginally trusted, no trust). This will allow to assess more precisely the trust degrees between users. Each user creates its own trust matrix, which will be initialized from certificates (public keys) in the keyring. A network discovery mechanism could also be established to expand the local network of trust. Finally, trust and reputation are assessed through the trust matrix.

The PGP client settings:*COMPLETS_NEEDED*, *MARGINALS_NEEDED* which are used to compute the required number of signatures generated by keys with full or marginal trust could be replaced by: *MINIMAL_TRUST* and *MINIMAL_REPUTATION*, representing the trust and reputation degrees needed to

validate a public key. The values of these parameters will depend of course on the used trust propagation system and personal policies.

4.3.4 Network Discovery Mechanisms

An essential mechanism of our trust management is the network discovery. Its role is to widen an entity view on the network. We do not detail this critical mechanism here but some of the following principles could ease the overall distributed management:

- Extension of the trust matrix from the trusted neighbors keyrings to avoid the problem of malicious groups.
- This extension may exceed the direct neighborhood to the neighbors of neighbors and so on. One can set a depth limit to be considered as the maximum length of certification paths to consider.
- Update the trust matrix when interacting with new entities. For the case of cross-certified PKIs, the interaction can be for example the request for verification of a certificate from another CA.
- ...

5 Conclusion

The actual public-key infrastructure models assume that the relations between the PKI entities are based on an absolute trust. However, several risks on the PKI procedures are related to these assumptions. In this article we introduce a simple distributed trust model in order to quantify and manage the trust in cross-certified PKIs and in the PGP web of trust. We use the formal semantics based calculus of trust introduced by [10, 11] and apply it to the PKIs. We also propose a new matrix powers algorithm efficiently evaluating the trust between entities of a PKI. This algorithm applies a selective aggregation on all [bounded] trust paths. Overall, our model combines the reputation and the trust notions to give a precise indication about the credibility of the PKI entities.

Further improvement includes a dedicated Network Discovery Mechanism, used to expand the trust sub-graph and to guaranty the safety in the trust model. Also the trust degrees could be a sensitive information. Therefore, the join use of trust matrices and homomorphic cryptosystems enabling a private computation of shared secret would be useful.

References

[1] T. Beth, M. Borcherding, and B. Klein. Valuation of trust in open networks. In *Proceedings of the Third European Symposium on Research in Computer Security*, ESORICS '94, pages 3–18, London, UK, 1994. Springer-Verlag.

[2] E. Carrara and G. Hogben. Reputation-based systems: a security analysis. Technical report, European Network and information Security Agency, Dec. 2007. http://www.enisa.europa.eu/act/it/oar/reputation-systems/reputation-based-systems-a-security-analysis/at_download/fullReport.

[3] S. Dolev, N. Gilboa, and M. Kopeetsky. Computing multi-party trust privately: in o(n) time units sending one (possibly large) message at a time. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1460–1465, New York, NY, USA, 2010. ACM.

[4] C. Ellison and B. Schneier. Ten risks of PKI: What you're not being told about Public Key Infrastructure. *Computer Security Journal*, 16(1):1–7, 2000. http://www.counterpane.com/pki-risks.pdf.

[5] E. Estrada and N. Hatano. Communicability in complex networks. *Physical Review E*, 77, 2008.

[6] E. Estrada and N. Hatano. Communicability graph and community structures in complex networks. *Applied Mathematics and Computation*, 214(2):500–511, 2009.

[7] E. Estrada, D. J. Higham, and N. Hatano. Communicability betweenness in complex networks. *Physica A-statistical Mechanics and Its Applications*, 388:764–774, 2009.

[8] F. Gomes. Security alert: Fraudulent digital certificates. Technical report, SANS Institute InfoSec Reading Room, June 2001. http://www.sans.org/reading_room/whitepapers/certificates/security-alert-fraudulent-digital-certificates_679.

[9] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 403–412, New York, NY, USA, 2004. ACM.

[10] J. Huang and D. Nicol. A calculus of trust and its application to pki and identity management. In *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, IDtrust '09, pages 23–37, New York, NY, USA, 2009. ACM.

[11] J. Huang and D. Nicol. A formal-semantics-based calculus of trust. *IEEE Internet Computing*, 14:38–46, September 2010.

[12] A. Jøsang. Artificial reasoning with subjective logic. In *Proceedings of the Second Australian Workshop on Commonsense Reasoning*, 1997.

[13] A. Jøsang. An algebra for assessing trust in certification chains. In *Network and Distributed System Security Symposium*. The Internet Society, 1999.

[14] A. Jøsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems.*, pages 279–311, June 2001.

[15] A. Jøsang. Probabilistic logic under uncertainty. In *Proceedings of Computing: The Australian Theory Symposium (CATS'07)*, January 2007.

[16] A. Jøsang and J. Golbeck. Challenges for robust of trust and reputation systems. In *Proceedings of the 5th International Workshop on Security and Trust Management (SMT 2009), Saint Malo, France*, 2009.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 640–651, New York, NY, USA, 2003. ACM.

[18] J. Linn. Trust models and management in public-key infrastructures. Technical report, RSA Laboratories, 2000. `ftp://ftp.rsa.com/pub/pdfs/PKIPaper.pdf`.

[19] T. Moses. Pki trust models. Technical report, IT University of Copenhagen, 2003. `http://www.itu.dk/courses/DSK/E2003/DOCS/PKI_Trust_models.pdf`.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, Nov. 1999.

[21] S. Pemmaraju and S. Skiena. *Computational discrete mathematics: combinatorics and graph theory with Mathematica*. Cambridge Univ Pr, 2003.

[22] M. K. Reiter and S. G. Stubblebine. Resilient authentication using path independence. *IEEE Trans. Comput.*, 47:1351–1362, December 1998.

[23] H. Rifà-Pous and J. Herrera-Joancomartí. An interdomain pki model based on trust lists. In J. Lopez, P. Samarati, and J. Ferrer, editors, *Public Key Infrastructure*, volume 4582 of *Lecture Notes in Computer Science*, pages 49–64. Springer Berlin / Heidelberg, 2007. `http://dx.doi.org/10.1007/978-3-540-73408-6_4`.

[24] Y. Wang and J. Vassileva. Trust and reputation model in peer-to-peer networks. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing*, P2P '03, pages 150–, Washington, DC, USA, 2003. IEEE Computer Society.

[25] R. Yahalom, B. Klein, and T. Beth. Trust relationships in secure systems-a distributed authentication perspective. In *Proceedings of the 1993 IEEE Symposium on Security and Privacy*, SP '93, pages 150–, Washington, DC, USA, 1993. IEEE Computer Society.

[26] C.-N. Ziegler and G. Lausen. Spreading activation models for trust propagation. In *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, EEE '04, pages 83–97, Washington, DC, USA, 2004. IEEE Computer Society.