SDSIrep: A Reputation System Based on SDSI*

Ahmed Bouajjani[1], Javier Esparza[2], Stefan Schwoon[2],
and Dejvuth Suwimonteerabuth[2]

[1] LIAFA, University of Paris 7, Case 7014, 75205 Paris cedex 13, France
[2] Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany

Abstract. We introduce SDSIrep, a reputation system based on the SPKI/SDSI authorization system. It is well-known that a system of SPKI/SDSI certificates corresponds to the formal model of a pushdown system (PDS). Our system, SDSIrep, allows principals to express trust and recommendations in the form of so-called certificates with weights. By interpreting weights as probabilities, we obtain a random-walk model of the reputation of a principal. Thus, SDSIrep represents an application of the theory of probabilistic PDSs to the field of computer security. We present an algorithm to compute the reputation of each principal. An extension of SDSIrep also provides for so-called intersection certificates, by which, loosely speaking, a principal gains reputation if recommended by all members of a given group of principals. On a formal-methods level, this extension makes SDSIrep correspond to probabilistic alternating PDSs, and we extend the underlying theory of PDSs to handle this case. As an example we sketch a small academic reputation system that combines information from different reputation sources, like conferences, coauthors, and rankings.

1 Introduction

In many Internet applications, notions of trust and reputation play an important role. In particular, in an open-world scenario where we do not know all the participants beforehand, we often need to decide whether to trust other persons without having met them before. Examples include systems like Ebay, where hitherto unknown participants engage in financial transactions; peer-to-peer file-sharing networks where people download files from one another and the academic world, where one often needs to assess a candidate's scientific merit.

If one cannot judge somebody else's trustworthiness oneself, a common solution is to assess their reputation: while *trust* is a "local" notion about the relation between two parties, *reputation* means somebody's "global" standing within some community. In the first two of the above scenarios, so-called *reputation systems* can be employed where a participant's reputation is computed from the experiences that other participants have made in prior transactions. A survey of reputation systems can be found in [1]. A concrete, well-known

* This work was partially supported by SFB 627 Nexus, Project A6.

C.R. Ramakrishnan and J. Rehof (Eds.): TACAS 2008, LNCS 4963, pp. 501–516, 2008.

example of a reputation system is the one used by Ebay, where every transaction can be assigned a rating that is stored in a central server, which can then authoritatively compute each participant's reputation. Such a system, which is based on the ratings users give to each other, is also called *user-driven.* In contrast, some domain-specific reputation systems such as the one proposed for rating Wikipedia contributors [2] are called *content-driven*, because they rate users based on how their contributions evolve in the system. Our proposal is not geared towards any specific domain, and we follow the user-driven approach.

People often place trust in individuals and in the recommendations given by well-reputed institutions. (Notice that trust and recommendations are closely related: normally, we would recommend a person if we trust them to be good at something.) For instance, when a well-reputed university hires a new researcher, we can interpret this as a recommendation that the university gives to the researcher. Likewise, when a well-reputed magazine publishes papers by a certain author, it implicitly recommends the contents of those papers, enhancing the reputation of the author. Such recommendations tend to "add up"; the more such sources of reputation we know of, the more we would trust a researcher.

In the following, we propose a framework for expressing trust and reputation in such a scenario. This framework allows to make statements that express trust in individuals as well as in (hierarchical) organizations. For instance, one can state that one recommends the employees of a certain university, or the coauthors of one's own papers. Moreover, such statements can be given weights to denote the degree of the recommendation.

For this, we borrow from and modify the SPKI/SDSI framework [3]; we therefore call our system SDSIrep. SPKI/SDSI was designed to denote naming policies in a distributed environment. Simply put, it allows to define groups of principals, which are described in distributed, hierarchical name spaces, and this idea is well suited for our purposes. We also show, by re-interpreting the notion of delegation in SPKI/SDSI, how an important distinction can be made between our trust of a person and our trust of their ability to judge the reputation of others. Our framework also borrows ideas from EigenTrust [4], a reputation system that has been proposed for peer-to-peer networks. We discuss the similarities and differences in Sections 3 and 6.

Previous work has shown that SPKI/SDSI has a strong connection to the theory of pushdown systems [5,6]. Moreover, SPKI/SDSI (and the associated pushdown theory) has been extended with weights, allowing to solve authorization problems with quantitative components. However, the extensions considered so far were not powerful enough to capture situations where trust "adds up" along multiple paths as in our scenario (see above). For example, the framework in [6] can express the fact that a certain level of trust exists if there is at least one path to support it; however, it cannot express the idea that the level of trust increases if multiple such paths exist. Recently, however, new results on probabilistic pushdown systems [7] open up the opportunity for such an extension.

Our paper makes the following contributions:

- We describe a new framework, called SDSIrep, that can be used to build a reputation system suitable for modelling trust relationships in an open-world scenario. Moreover, SDSIrep allows to distinguish between the trust one has in a person and in their recommendations. We then show how these trust values can be aggregated to measure each participant's reputation.
- We expose the relationship between SDSIrep and probabilistic pushdown systems and extend the probabilistic approach to *alternating* pushdown systems. This solution allows to handle so-called intersection certificates, increasing the expressiveness of the SDSIrep framework at practically no extra computational cost.
- As a small case study, we design a system for measuring academic reputation. We implement the algorithms for computing reputations in this example and report on their performance.

We proceed as follows: In Section 2 we recall basic notions of SPKI/SDSI. We then present SDSIrep, our reputation system, in Section 3, and solve the associated trust and reputation problem. In Section 4 we extend SDSIrep with intersection certificates. We present some experimental results in Section 5 before discussing related work and concluding in Section 6.

2 Background

This section provides some background on SPKI/SDSI and pushdown systems.

2.1 A Brief Introduction to SPKI/SDSI

The SPKI/SDSI standard was proposed in [3] and formalised in first-order logic in [8]. We present a subset of SPKI/SDSI that has been considered in most of the work on this topic. The full SPKI/SDSI standard also provides for so-called threshold certificates, which we treat later in Section 4.

SPKI/SDSI was designed to denote authorization policies in a distributed environment. A central notion of SPKI/SDSI are *principals.* A principal can be a person or an organisation. Each principal defines his/her own namespace, which assigns *rôles* to (other) principals. For instance, principal *Fred* can define the rôle `friend` and associate principal *George* with this rôle. Such associations are made in SPKI/SDSI by issuing so-called *name certificates* (*name certs*, for short). A special feature is that principals may reference the namespace of other principals in their certificates. For instance, *Fred* may state that all of *George*'s friends are also his own friends. In this way, SPKI/SDSI allows to associate a rôle with a group of principals described in a symbolic and distributed manner. SPKI/SDSI then allows to assign permissions to rôles using so-called *authorisation certificates* (or *auth certs*).

The SPKI/SDSI standard also uses a public-key infrastructure that allows for certificates to be signed and verified for authenticity. Public-key infrastructure does not play a major rôle in our approach, but we shall re-use the ideas behind the naming scheme.

More formally, a SPKI/SDSI system can be seen as a tuple (P, A, C), where P is a set of *principals*, A is a set of *rôle identifiers* (or identifiers, for short) and $C = Na \uplus Au$ is a set of certificates. Certificates can be either *name certificates* (contained in Na), or *authorization certificates* (contained in Au).

A *term* is formed by a principal followed by zero or more identifiers, i.e., an element of the set PA^*. A term t is interpreted as denoting a set of principals, written $[\![t]\!]$, which are defined by the set of name certificates (see below).

A name certificate is of the form $p\ \mathtt{a} \rightarrow t$, where p is a principal, $\mathtt{a}$ is an identifier, and t is a term. Notice that $p\ \mathtt{a}$ itself is a term. The sets $[\![t]\!]$, for all terms t, are the smallest sets satisfying the following constraints:

- if $t = p$ for some principal p, then $[\![t]\!] = \{p\}$;
- if $t = t'\ \mathtt{a}$, then for all $p \in [\![t']\!]$ we have $[\![p\ \mathtt{a}]\!] \subseteq [\![t]\!]$;
- if $p\ \mathtt{a} \rightarrow t$ is a name certificate, then $[\![t]\!] \subseteq [\![p\ \mathtt{a}]\!]$.

For instance, if *Fred* and *George* are principals and `friend` is an identifier, then *Fred* `friend` → *George* expresses that George is a friend of Fred, and *Fred* `friend` → *George* `friend` means that all of George's friends are also Fred's friends, and *Fred* `friend` → *Fred* `friend` `friend` says that the friends of Fred's friends are also his friends.

An authorisation certificate has the form $p\ \Box \rightarrow t\ b$, where p is a principal, t is a term, and b is either $\Box$ or $\blacksquare$. Such a certificate denotes that p grants some authorisation to all principals in $[\![t]\!]$. If $b = \Box$, then the principals in $[\![t]\!]$ are allowed to delegate said authorisation to others; if $b = \blacksquare$, then they are not. (Auth certs in SPKI/SDSI contain more details about the authorisation that they confer; this detail is not important for our approach.)

More formally, authorisation certs define a smallest relation $aut\colon P \times P$ between principals such that $aut(p, p')$ holds iff p grants an authorisation to p':

- if there is an auth cert $p\ \Box \rightarrow t\ b$, for $b \in \{\Box, \blacksquare\}$, and $p' \in [\![t]\!]$, then $aut(p, p')$;
- if there is an auth cert $p\ \Box \rightarrow t\ \Box$, $p' \in [\![t]\!]$, and $aut(p', p'')$, then $aut(p, p'')$.

For instance, the certificate *Fred* $\Box \rightarrow$ *George* `friend` $\blacksquare$ means that Fred grants some right to all of George's friends, however, George's friends are not allowed to delegate that right to other principals.

The *authorisation problem* in SPKI/SDSI is to determine, given a system (P, A, C) and two principals p and p', whether p' is granted authorisation by p, i.e., whether $aut(p, p')$.

2.2 SPKI/SDSI and Pushdown Systems

Certificates in SPKI/SDSI can be interpreted as prefix rewrite systems. For instance, if $p\ \mathtt{a} \rightarrow p'\ \mathtt{b}\ \mathtt{c}$ and $p'\ \mathtt{b} \rightarrow p''\ \mathtt{d}\ \mathtt{e}$ are two certificates interpreted as rewrite rules, then their concatenation rewrites $p\ \mathtt{a}$ to $p''\ \mathtt{d}\ \mathtt{e}\ \mathtt{c}$. In SPKI/SDSI, a concatenation of two or more certificates is called a *certificate chain*. It is easy to see that the authorisation problem, given principals p and p', reduces to the problem of whether there exists a certificate chain that rewrites $p\ \Box$ into either

$p' \ \square$ or $p' \ \blacksquare$ (in the first case, p' also has the right to delegate the authorisation further, in the second case he has not).

Moreover, it is well-known that the type of rewrite systems induced by a set of SPKI/SDSI certificates is equivalent to that of a pushdown system (PDS), see, e.g. [5,6,9,10]. For example, a cert like $p \ \mathtt{a} \rightarrow p' \ \mathtt{b} \ \mathtt{c}$ is interpreted as a pushdown transition, where p, p' are states of the finite control and where the stack content $\mathtt{a}$ is replaced by $\mathtt{bc}$. Then, the SPKI/SDSI authorisation problem reduces to a pushdown reachability problem, i.e., whether from control location p with the symbol $\square$ on the stack (and nothing else) one can eventually reach control location p' with empty stack.

In the following, we present our system, SDSIrep, which extends SPKI/SDSI with weights on certificates and with so-called intersection certificates. In pushdown-automata theory, these extensions correspond to weighted pushdown systems [6] and alternating pushdown systems [9]. For brevity, we will not elaborate on these correspondences any further, and we simply apply the appropriate pushdown theory to SDSIrep. Notice, however, that the combination of weighted and alternating systems employed in this paper is novel.

3 A SDSI-Based Reputation System

We now explain the model of trust and reputation employed by SDSIrep, which motivates the design of our system, given in Section 3.2. We then proceed to show how to compute trust and reputation values in this system. In Section 4 we introduce an extension that further improves the expressiveness of the system.

3.1 A Numerical Model of Trust

Many reputation systems allow participants to express degrees of trust numerically. A common problem with this is that malicious participants may attempt to "spam" the system and boost each other's reputations with arbitrarily high values. The solution employed here is to normalise trust values. In SDSIrep, each principal has a total trust of 1 at his/her disposal, fractions of which can be allocated freely to other principals.

Like in EigenTrust [4], this approach lends itself to a probabilistic interpretation, similar to the "Random Surfer" model used in Google's PageRank [11]. We interpret a SDSIrep system as a Markov chain whose states are the participants, and where the trust that participant A has in B (expressed as a fraction between 0 and 1) serves as the probability of going from A to B. Then, one way to find reputable participants is to perform a random walk on this Markov chain: after a "long enough" period of time, one is more likely to be at a well-reputed participant than not. In particular, each party's reputation is taken as their value in the stationary vector of the Markov chain. Thus, even though all participants can distribute a total trust value of 1 to others, this does mean that the opinions of all participants have the same influence. Well-reputed participants will be visited more often in a random walk than less-reputed ones, giving more weight to their opinions.

What distinguishes SDSIrep from EigenTrust is the way peer-to-peer ratings are specified: principals can assign their trust to *groups of* principals that are defined indirectly, using name certificates like in SPKI/SDSI. Membership in a group is associated with a numeric value, in a kind of fuzzy logic. Suppose, for instance, that a researcher wants to recommend those researchers whose findings have been published in a certain journal. Then, somebody with 10 papers in that journal could be considered to belong more strongly to that group than somebody with just one paper. SDSIrep allows to make such distinctions.

In the terminology of [1], PageRank, EigenTrust, and SDSIrep are all examples of *flow models*. In a flow model, participants can only increase their reputation at the cost of others. This property is obviously satisfied by SDSIrep, because the sum of the reputation values over all participants is bounded by 1. Thus, the absolute reputation values computed within the SDSIrep framework have no meaning in themselves; they only indicate how well-reputed each participant is in comparison with others.

3.2 SDSIrep Certificates

Our system is based on the design of SPKI/SDSI, i.e. a SDSIrep system is again a triple (P, A, C) with (almost) the same meaning as in Section 2. However, in SDSIrep, we are not concerned with authorisation problems. Rather, we reinterpret authorisation certificates as *recommendations*, which express trust in certain groups of principals.

Another change is the addition of weights to certificates. Adding weights drawn from the set $[0, 1]$ to recommendation certs allows to express the degree of recommendations. Similarly, weights on name certs express the degree of membership to a set. We provide only simple examples in this section; a more elaborate example of a SDSIrep system is presented in Section 5.

Weighted recommendation certs allow to recommend all members of a group by issuing one single cert. This reflects common situations in which a principal recommends a group even though the members of the group change along time, or even though he or she does not know many of its members.

A weighted recommendation cert has the form $p \,\square \xrightarrow{x} t \,\blacksquare$, where $x \in [0, 1]$ is its weight. Such a cert states that the principal p recommends the principals of the set $[\![t]\!]$ with weight x. The cert $p \,\square \xrightarrow{x} t \,\square$ states that p recommends not the principals of $[\![t]\!]$ themselves, but *their recommendations* with weight x.

As an example, suppose that researcher A wants to give 50% of his "share" of recommendations to the authors of journal J. This could be stated by the cert $A \,\square \xrightarrow{0.5} J\ \mathtt{aut}\ \blacksquare$. To explain the semantic difference between $\square$ and $\blacksquare$, imagine a reputation system for film directors with directors and critics as principals. Film critics will not be recommended for their directing skills, only for their recommendations. A similar distinction exists in PGP, which separates the trust that principals have in the authenticity of some person's public key from the trust they have in the ability of that person to correctly judge the authenticity of other people's keys.

Notice that there is no certificate with $\blacksquare$ on the left-hand side. Thus, a chain starting with a recommendation cert of the form $p \;\Box \xrightarrow{x} t \;\blacksquare$ necessarily ends when t has been rewritten to an element of $\llbracket t \rrbracket$, whereas a chain starting with $p \;\Box \xrightarrow{x} t \;\Box$ allows to apply further recommendation certs at that point. This corresponds to the idea that $\Box$ expresses a recommendation of somebody's recommendations, whereas $\blacksquare$ expresses a recommendation of that person as such.

To normalise the trust values in the system, and in order to enable a probabilistic interpretation as discussed in Section 3.1, we additionally demand that the weights on each principal's recommendation certs add up to 1.

Weighted name certs have the form $p\;\mathtt{a} \xrightarrow{x} t$, where $x \in [0,1]$. Intuitively, such a cert states a fuzzy membership relation: the elements of $\llbracket t \rrbracket$ belong to the set $\llbracket p\;\mathtt{a} \rrbracket$ with membership degree x.

As an example, consider a journal J and an identifier `aut` such that $\llbracket J\;\mathtt{aut} \rrbracket$ are the authors that have published in J. Then, if the journal has published 100 papers and B has authored 10 of them, B might be considered to belong to $\llbracket J\;\mathtt{aut} \rrbracket$ with degree 10%, expressed as $J\;\mathtt{aut} \xrightarrow{0.1} B$. In order to uphold the fuzzy-set interpretation we demand that for all pairs $p\;\mathtt{a}$, the sum of the weights on all name certs with $p\;\mathtt{a}$ on the left-hand side is 1.

3.3 Certificate Chains and Markov Chains

Consider the certs $A \;\Box \xrightarrow{0.5} J\;\mathtt{aut}\;\blacksquare$ and $J\;\mathtt{aut} \xrightarrow{0.1} B$. If A gives 50% of his recommendations to the authors of J, and B has authored 10% of the papers in J, then a natural interpretation is that 5% of A's recommendations go to B. Thus, the weight of the certificate chain formed from the two certs is obtained by multiplying their individual weights.

To find out how much trust A puts into B, we are interested in the certificate chains going from $A \;\Box$ to $B \;\blacksquare$. In general, there could be more than one such chain. Thus, one needs to find all these chains in order to determine the degree of recommendation A gives to B. The following example shows that the number of such paths can in fact be infinite:

$$A \;\Box \xrightarrow{1} A\;\mathtt{friend}\;\blacksquare \quad (1) \qquad A\;\mathtt{friend} \xrightarrow{x} B \quad (3)$$

$$B \;\Box \xrightarrow{1} A \;\blacksquare \quad (2) \qquad B\;\mathtt{friend} \xrightarrow{1} A \quad (4)$$

$$A\;\mathtt{friend} \xrightarrow{1-x} A\;\mathtt{friend}\;\mathtt{friend} \quad (5)$$

Cert (5) is the crucial one. It states that the friends of A's friends also belong to A's friends, albeit with smaller weight. Notice that whenever this cert can be applied, it can be applied arbitrarily often. So A recommends B through many possible chains: for instance, we can apply the cert (1), then cert (5) $2n$ times, and then certs (3) and (4) alternatingly n times each.

We can now define the two algorithmic problems related to SDSIrep. The *trust problem* in SDSIrep is as follows: Given two principals p and p', compute the sum of the weights of all certificate chains that rewrite $p \;\Box$ into $p' \;\blacksquare$. The *reputation problem* is to compute, for each principal, their value in the stationary vector of the Markov chain in which the transition probabilities are given by the solutions to the pairwise trust problems. We discuss solutions for the trust and reputation problems in Section 3.4.

3.4 Solving the Trust and Reputation Problems

It is easy to see that a system of SDSIrep certificates corresponds to a probabilistic pushdown system (pPDS) [7]. The trust problem in SDSIrep then reduces to a pPDS reachability problem, i.e., given p and p', compute the probability of reaching control location p' with stack content $\blacksquare$ when starting from p and $\square$.

Following [7], the solution to this is given by an equation system (see also [12] for the same result using a different but equivalent model). Given a SDSIrep system (P, A, C), the variables are elements of the set $\{\, [p, \mathtt{a}, q] \mid p, q \in P,\ \mathtt{a} \in A \,\}$, where $[p, \mathtt{a}, q]$ denotes the probability of rewriting the term $p\ \mathtt{a}$ into q. To solve the trust problem, we also add an artificial certificate $p'\ \blacksquare \xrightarrow{1} \bar{p}'$, where $\bar{p}'$ is a fresh control location; since $p'\ \blacksquare$ does not appear on any other left-hand side, the solution of $[p, \square, \bar{p}']$ gives us the trust placed by p in p'.

Each variable $[p, \mathtt{a}, q]$ has the following equation:[1]

$$[p, \mathtt{a}, q] = \sum_{p\mathtt{a} \xrightarrow{x} p'\mathtt{bc}} x \cdot \sum_{r \in P} [p', \mathtt{b}, r] \cdot [r, \mathtt{c}, q] + \sum_{p\mathtt{a} \xrightarrow{x} p'\mathtt{b}} x \cdot [p', \mathtt{b}, q] + \sum_{p\mathtt{a} \xrightarrow{x} q} x \quad (6)$$

Intuitively, equation (6) sums up the probabilities for all the possible ways of reaching q from $p\ \mathtt{a}$. We just explain the first half of the expression; the other cases are simpler and analogous: if $p\ \mathtt{a}$ is replaced by $p'\ \mathtt{b}\ \mathtt{c}$ (with probability x), then one first needs to rewrite this term to $r\ \mathtt{c}$ for some $r \in P$, which happens with the probability computed by $[p', \mathtt{b}, r]$, and then $r\ \mathtt{c}$ needs to be rewritten into q, which is expressed by the variable $[r, \mathtt{c}, q]$.

For instance, consider the system consisting of rules (1) to (5) in Section 3.3. Some of the resulting equations are (abbreviating `f` for `friend`):

$$\begin{array}{ll} [B, \mathtt{f}, A] = 1 & [B, \square, B] = 1 \cdot [A, \blacksquare, B] \\ \multicolumn{2}{l}{[A, \mathtt{f}, B] = x + (1 - x) \cdot ([A, \mathtt{f}, A] \cdot [A, \mathtt{f}, B] + [A, \mathtt{f}, B] \cdot [B, \mathtt{f}, B])} \end{array}$$

This equation system has a least solution, and the elements of this least solution correspond to the aforementioned probabilities. Notice that the equation system is non-linear in general. We discuss the resulting algorithmic problems in more detail in Section 5.3. The following theorem now follows from the definitions and the results of [7,12].

Theorem 1. *The solution to the trust problem for principals* p *and* p' *is equal to the value of variable* $[p, \square, \bar{p}']$ *in the least solution of the equation system (6).*

[1] We show the equation system for the case where the terms on the right-hand side of each cert consist of at most two identifiers; however, this is not a restriction as any system can be converted into a system observing this rule with linear overhead [5].

In general, the least solution cannot be computed exactly, but can be approximated to an arbitrary degree of precision using standard fix-point computation methods [7]. We give more details on this computation when discussing our experiments in Section 5. Notice that the equation system actually gives the probabilities (and hence the trust values) for all pairs of principals, therefore all values in the Markov chain used for solving the reputation problem can be obtained from just one fixpoint computation.

As discussed in Section 3.1, a measure of the "reputation" of principals in the system can be obtained by computing the stationary vector of the Markov chain whose states are the principals and whose transition probabilities are given by the solutions of the trust problems. Computing the stationary vector amounts to solving a linear equation system, using well-known techniques.

However, for the stationary vector to exist, the Markov chain needs to be irreducible and aperiodic. This is not guaranteed in general: e.g., if there is a clique of participants who trust only each other, the Markov chain contains a "sink", i.e., it is not irreducible. This type of problem is also encountered in other models based on random walks, e.g. EigenTrust or PageRank, and the solutions employed there also apply to SDSIrep. For instance, the irreducibility and aperiodicity constraint can be enforced by allowing the random walk to jump to random states at any move with small probability. Notice that the example in Section 5 does not exhibit this kind of problem; therefore, we did not use this trick in our experiments.

4 Intersection Certificates

The SPKI/SDSI standard provides for so-called *threshold certificates*, which consist of, say, an auth cert of the form $p \,\square \rightarrow \{t_1 b_1, \ldots, t_n b_n\}$, where $b_1, \ldots, b_n \in \{\square, \blacksquare\}$, and an integer $k \leq n$. The meaning of such a cert is that p grants authorisation to principal p' if there is a certificate chain to p' from at least k out of $t_1 b_1, \ldots, t_n b_n$. Threshold certificates for name certs could be defined analogously. We restrict ourselves to the case where $k = n$ and use the more suggestive name *intersection certificate* instead.[2]

In this section we show how intersection certificates can be added to SDSIrep and define the corresponding trust problem and a probabilistic interpretation for SDSIrep with intersection certificates. We then show that the equation system from Section 3.4 can be modified to accomodate this extension.

Algorithms for authorisation in SPKI/SDSI with intersection were studied in [5] and [9]. In the latter, the problem was reduced to reachability in *alternating pushdown systems* (APDS). It turns out that the authorisation problem with

[2] In the case *without* weights, any certificate where $k < n$ can be replaced by a set of certificates, one for each k-sized subset of the right-hand side. In the case *with* weights, this can also be done, but the degree to which a participant belongs to the right-hand side can be interpreted in different ways. This question of interpretation is beyond the scope of the paper.

intersection is EXPTIME-complete in general, but remains polynomial when intersection is restricted to authorisation certs. This distinction translates directly to SDSIrep; therefore, we restrict intersection to recommendation certs.

4.1 Intersection Certs in SDSIrep

Sometimes one wishes to recommend principals belonging to the intersection of two or more groups. For instance, researcher A may wish to recommend those of his co-authors that have published in journal J. In SDSIrep, we model this by a certificate such as $A \,\square \xrightarrow{x} \{A\ \texttt{coaut}\ \blacksquare,\ J\ \texttt{aut}\ \blacksquare\}$. In general, intersection certificates have the form $p \,\square \xrightarrow{x} \{t_1\ b_1, \ldots t_n\ b_n\}$, where $b_1, \ldots, b_n \in \{\square, \blacksquare\}$, and express that p recommends the set $\bigcap_{i=1}^{n} [\![t_i]\!]$ with weight x.

The trust problem for the case without intersection certs consists of computing the values of certificate chains. When intersection certs come into play, we need to think of certificate trees instead, where each node is labelled by a term, and a node labelled by term t has a set of children labelled by T if T is the result of applying a rewrite rule to t. For instance, if in addition to the previous intersection certificate we have $A\ \texttt{coaut} \xrightarrow{y} B$ and $J\ \texttt{aut} \xrightarrow{z} B$, then we have the following certificate tree:

$$A \,\square \xrightarrow{x} \left\{ \begin{array}{l} A\ \texttt{coaut}\ \blacksquare \xrightarrow{y} B\ \blacksquare \\ \;\;J\ \texttt{aut}\ \blacksquare \;\;\;\xrightarrow{z} B\ \blacksquare \end{array} \right.$$

In the probabilistic interpretation, the probability for this tree is $x \cdot y \cdot z$. Thus, the *trust problem for SDSIrep with intersection* is as follows: Given principals p and p', compute the sum of the probabilities of all trees whose root is labelled by $p \,\square$ and all of whose children are labelled by $p' \,\blacksquare$. Notice that the solution for the associated *reputation problem* remains essentially unchanged, as the addition of intersection certs merely changes the way peer-to-peer trust is assigned.

4.2 Solving the Trust Problem with Intersection Certs

We now extend the equation system from Section 3.4 to the case of intersection certificates. (In terms of [9], we extend the solution to probabilistic APDSs.)

Let $\varXi := \{\blacksquare, \square\}$. Since intersection is restricted to recommendation certs, the following important properties hold: (1) if $p \,\square$ is the root of a certificate tree, then all nodes are of the form $t\, b$, where $b \in \varXi$ and t does not contain any symbol from $\varXi$; (2) if a term t of a certificate tree has more than one child, $t = p \,\square$ for some p. It follows that if a term pw is the root of a tree and w does not contain any occurrence of $\blacksquare$ or $\square$, then every term of the tree has at most one child, and so the tree has a unique leaf. We exploit this fact in our solution.

Let (P, A, C) be a SDSIrep system with intersection certificates. The variables of the new equation system are of the form $[p, \bot, q]$ or $[p, w, q]$, where $p, q \in P$, $\bot \in \varXi$, $w \in A^*$, and w must be a suffix of the right-hand side of a cert. Notice that, by definition, w contains no occurrence of $\blacksquare$ or $\square$. The variable $[p, \bot, q]$ represents the probability of, starting at $p\bot$, eventually reaching a tree where all leaves are labelled with q. The variable $[p, w, q]$ represents the probability of, starting at pw, reaching a tree whose unique leaf (here we use the fact above) is labelled with q. We add (as in Section 3.4) an artificial rule $p' \blacksquare \xrightarrow{1} \overline{p}'$, which is the only rule consuming the $\blacksquare$ symbol.

For $p, q \in P$ and $\gamma \in A \cup \Xi$, we have:

$$
\begin{aligned}
[p,\gamma,q] = & \sum_{p\gamma \xrightarrow{x} p'w} x \cdot [p',w,q] \quad + \\
& \sum_{p\gamma \xrightarrow{x} \{p_1 w_1 \bot_1, \ldots, p_n, w_n \bot_n\}} x \cdot \sum_{q_1,\ldots,q_n \in P} \prod_{i=1}^{n} [p_i, w_i, q_i] \cdot [q_i, \bot_i, q]
\end{aligned}
\tag{7}
$$

(Notice that if $\gamma \in A$ then the second sum is equal to 0 by property (2) above.) Moreover, we set $[p,\varepsilon,q] = 1$ if $p = q$ and 0 otherwise, and $[p,\gamma w,q] = \sum_{q' \in P} [p,\gamma,q'] \cdot [q',w,q]$ for every two $p, q \in P$, $\gamma w \in (A \cup \Xi)^+$. Notice that γw is a suffix of the right-hand side of some cert, and therefore so is w.

The intuition for these equations is the same as in the case without alternation, see Section 3.4. The corresponding equation system also has the same properties and can be solved in the same way.

Theorem 2. *The solution to the trust problem for principals p and p' in a SDSIrep system with intersection certificates is equal to the solution of variable $[p, \Box, \overline{p'}]$ in the least solution of the equation system (7).*

5 Experiments

For demonstration purposes, we have used SDSIrep to model a simple reputation system for the PC members of TACAS 2008. We have chosen this example because the reader is likely to be familiar with the sources of reputation in academia, in particular in computer science. We do not claim that our experiments say anything really relevant about the actual reputation of the PC members, in particular, because part of the required data (the personal preferences of the PC members, see below) was not available to us.

In this section, we give some details on this system, and report on the performance of our solver for the equation systems given in Sections 3.4 and 4.2.

5.1 A Small System for Academic Reputation

Principals and identifiers. The set of principals contains the 28 members of the TACAS programme committee, 6 of the main conferences on automated verification (CAV, ICALP, LICS, POPL, VMCAI, TACAS), and 3 rankings: the CiteSeer list of 10,000 top authors in computer science (year 2006) [13], denoted `CiteSeer`, the CiteSeer list of conferences and journals with the highest impact factors [14], denoted `Impact`, and the list of h-indices for computer scientists [15], denoted `H-index`. The identifiers are `aut`, `publ`, `coaut`, and `circ`, with the following fuzzy sets as intended meaning:

- $[\![c\ \mathtt{aut}]\!]$: researchers that publish in conference c;
- $[\![r\ \mathtt{publ}]\!]$: conferences in which researcher r has published;
- $[\![r\ \mathtt{coaut}]\!]$: r's co-authors;
- $[\![r\ \mathtt{circ}]\!]$: r's "circle", defined as r's coauthors, plus the coauthors of r's coauthors, and so on (the degree of membership to the circle will decrease with the "distance" to r).

Name certs. Some illustrative examples of the certs in our system are shown in Figure 1. For the sake of readability, we present them without having normalised the weights (normalized values are more difficult to read and compare). So, to set up the equation system, one has to take all the certs with the same tuple $p\ \mathtt{a}$ on the left-hand side, say $p\ \mathtt{a} \xrightarrow{x_1} t_1, \ldots, p\ a \xrightarrow{x_n} t_n$, and then replace each x_i by $x_i / \sum_{i=1}^{n} x_i$. In this way, all weights are normalised.

Two certs describe to which degree a PC member is an author of a conference and which share each conference has in the PC member's publication list. In both cases, the weight (before normalisation) is the number of papers the author has published in the conference, obtained from DBLP [16]. For instance, for TACAS and Kim Larsen (KL), we have certs (8) and (9).

Another set of certs describes which PC members are coauthors of each other. The weight is the number of jointly written papers, obtained again from DBLP. For instance, cert (10) denotes that KL has written 22 papers with PP.

Finally, each PC member has a circle of fellow researchers, composed of the member's coauthors, the coauthors of the member's coauthors, and so on. We define KL's circle by means of certs (11) and (12).

$$\mathtt{TACAS\ aut} \xrightarrow{10} \mathtt{KL} \tag{8}$$
$$\mathtt{KL\ publ} \xrightarrow{10} \mathtt{TACAS} \tag{9}$$
$$\mathtt{KL\ coaut} \xrightarrow{22} \mathtt{PP} \tag{10}$$
$$\mathtt{KL\ circ} \xrightarrow{0.8} \mathtt{KL\ coaut} \tag{11}$$
$$\mathtt{KL\ circ} \xrightarrow{0.2} \mathtt{KL\ circ\ circ} \tag{12}$$
$$\mathtt{Impact}\ \Box \xrightarrow{1.24} \mathtt{TACAS\ aut}\ \blacksquare \tag{13}$$
$$\mathtt{H\text{-}index}\ \Box \xrightarrow{34} \mathtt{KL}\ \blacksquare \tag{14}$$
$$\mathtt{CiteSeer}\ \Box \xrightarrow{2023} \mathtt{KL}\ \blacksquare \tag{15}$$
$$\mathtt{KL}\ \Box \xrightarrow{4} \mathtt{KL\ publ\ aut}\ \blacksquare \tag{16}$$
$$\mathtt{KL}\ \Box \xrightarrow{3} \mathtt{KL\ circ}\ \blacksquare \tag{17}$$
$$\mathtt{KL}\ \Box \xrightarrow{2} \mathtt{Impact}\ \Box \tag{18}$$
$$\mathtt{KL}\ \Box \xrightarrow{3} \mathtt{CiteSeer}\ \Box \wedge \mathtt{H\text{-}index}\ \Box \tag{19}$$

Fig. 1. Name and recommendation certificates for the example

Recommendation certs. The system contains one recommendation cert for each conference, in which Impact recommends the authors of the conference with the weight given by its impact factor. For TACAS we have cert (13).

The next two certs, (14) and (15) express that the h-index and CiteSeer lists recommend a PC member (in this case KL) with a weight proportional to his h-index and to his number of citations in the list, respectively.

Finally, each PC member issues four more certs. The certs for KL are given in (16)–(19). Intuitively, they determine the weight with which KL wishes to recommend his circle, the authors of the conferences he publishes in, and how much trust he puts in the Citeseer and h-index rankings. In a real system, each PC member would allocate the weights for his/her own certs; in our example we have assumed that all PC members give the same weights. In order to illustrate the use of intersection certs we have assumed that `KL` only recommends researchers on the basis of their ranking values if they appear in *both* CiteSeer's list and in the h-index list (19). Moreover, observe that in certs (18) and (19), `KL` places trust in the *recommendations* given by the rule targets (signified by $\Box$), whereas in the other rules he expresses trust in the principals themselves.

In the following two sections we describe the running times and some interesting aspects of solving the equation systems computing the reputation of each researcher. All experiments were performed on a Pentium 4 3.2 GHz machine with 3 GB memory.

5.2 Experiment 1

We have written a program which takes as input the set of SDSIrep certificates described above, generates the equation system of Section 4.2, and computes its solution. We can then compute the degree to which researchers recommend one another. From the result we build a Markov chain as described in Section 3.3. The stationary distribution of the Markov chain, given at the top of Table 1, can be interpreted as the (relative) reputation of each researcher when compared to the others in the system. The lower part of Table 1 shows how the running time scales when the number of researchers is increased. For this experiment we have put together the PCs of TACAS, FOSSACS, and ESOP, with a total of 76 members, adding FOSSACS and ESOP to the list of conferences. We have computed the stationary distribution for subsets of 10, 20, ..., 76 PC members. The first line of the table shows the number of variables in the system (which is also the number of equations), and the second shows the time required to solve it and compute the stationary distribution.

Table 1. Stationary distribution for TACAS PC members (values multiplied by 1000) and statistics for different numbers of researchers

PB	EB	TB	RC	BC	BD	PG	OG	AG	FH	MH	JJ	KJ	JK	BK	MK	KL	NL	KN	PP	SR	CR	JR	AR	SS	SS	BS	LZ
26	18	19	78	45	6	56	60	30	19	45	19	5	23	10	30	88	26	37	33	64	22	45	6	54	15	80	41

scientists	10	20	30	40	50	60	70	76
variables	627	1653	3089	4907	7126	9752	12777	14779
time (s)	0.47	2.07	6.85	12.55	23.90	44.89	78.35	106.55

Notice that the equation system used here is non-linear (see Section 4.2). Following [12], we solve it using Newton's iterative method, stopping when an iteration does not change any component of the solution by more than 0.0001.

5.3 Experiment 2

In contrast to other trust systems, in which trust is assigned from one individual to another, our choice of SDSI allows to assign trust measures to sets of principals using multiple levels of indirection. For instance, A can transfer trust to

B because B is a coauthor of C, and C publishes in the same conference as A. This added expressiveness comes at a price. Certs like (12) or (16), with more than one identifier on the right-hand side, cause the resulting equation system to become non-linear (see Section 3.4). Likewise, intersection certs also cause non-linear equations (see Section 4.2).

On the other hand, if the system does not contain these two types of certs, the resulting equation system is linear, and instead of Newton's method more efficient techniques can be applied, e.g. the Gauß-Seidel method.

In the following, let us assume that intersection certs are absent. Consider cert (12). The certificate is "recursive" in the sense that it can be applied arbitrarily often in a certificate chain, rewriting `KL circ` to `KL` $\texttt{circ}^n$, for any $n \geq 1$. Thus, the length of terms to which `KL circ` can be rewritten is unbounded. (In pushdown terms, the "stack" can grow to an unbounded size.) If the set of certs is such that this effect cannot happen, then each term can be rewritten into only *finitely many* different other terms. Therefore, we can apply a process similar to that of "flattening" a PDS into a finite-state machine and derive a *larger*, but *linear*, equivalent equation system. If there are recursive certs, we can still choose an arbitrary bound on the length of terms and ignore the contributions of larger terms. In this case, the "unflattened" and "flattened" systems do not have the same solution, but the solution of the "flattened" system converges to the solution of the "unflattened" one when the bound increases.

This provokes the question of whether the performance of the equation solver can be improved by bounding the maximal term length, "flattening" the non-linear system into a linear one, and solving the linear system. In order to experimentally address this question, we again took the system introduced in Section 5.1, but without cert (19). We fixed the maximal term depth to various numbers, computed the corresponding linear flattened systems, and solved them using the Gauß-Seidel method. (We omit the details, which are standard.)

We found that in this example flattening works very well. Even with stack depth 2 we obtained a solution that differed from the one given by Newton's method by less than 1% and can be computed in 1.23 seconds instead of 5.83. Table 2 shows the results for stack depths up to 8, i.e. the size of the equation system obtained for each stack depth and the time required to solve it. Notice that in this case, the growth of the equation system as the stack depth grows is benign (only linear); in general, the growth could be exponential.

This result might suggest that using Newton's method could always be replaced by flattening in the absence of intersection certs. However, some caution is required. When we tried to repeat the experiment for the case with 76 researchers, our solver was able to solve the unflattened system within two minutes, but ran out of memory even for a flattened stack depth of 2.

Table 2. Size of equation system and running times for flattened systems

	Unflattened	Depth 2	Depth 3	Depth 4	Depth 5	Depth 6	Depth 7	Depth 8
vars	2545	5320	7059	8798	10537	12276	14015	15754
time	5.83	1.23	3.32	6.39	10.34	18.78	32.18	42.97

6 Discussion and Related Work

There is a large and growing body of literature on trust and reputation systems, see e.g. [1,17]. In this paper, we have proposed a new framework, SDSIrep, that is novel (to the best of our knowledge) in the way it expresses transitive trust relations in an open-world scenario. More specifically, trust can be assigned to principals based on their memberships in a group described by specific attributes, e.g. co-authors of a researcher or employees of a certain university. We believe that this mimics an important facet of how reputation is usually perceived.

Most trust and reputation systems collect peer-to-peer trust ratings and aggregate a global reputation from these ratings. EigenTrust [4] is an example of a system that also takes transitive trust into account, and it shares some similarities with SDSIrep. Both EigenTrust and SDSIrep allow individual users to express and quantify their personal trust relationships. In EigenTrust, principals express how much they trust their peers, and trust in a peer automatically translates into trusting the peer's recommendations, and so on. In the terminology of [1], EigenTrust is an example of a *flow model*. SDSIrep falls into the same category, but differs in the means in which trust between principals is defined. In SDSIrep, trust can be assigned to groups of principals (see above), and we allow to distinguish between how much we trust a person and how much we trust their recommendations.

Both SDSIrep and EigenTrust make use of a probabilistic interpretation by which these recommendations are aggregated into a measure of reputation. In both cases, this measure is obtained from a Markov chain whose entries are given by the peer-to-peer recommendations. In EigenTrust, the values of this Markov chain are supplied directly by the users, whereas in SDSIrep they are obtained by evaluating the certificates. Thus, roughly speaking, every SDSIrep system has an equivalent EigenTrust system. However, the translation from SDSIrep to EigenTrust is not completely straightforward, it requires to solve the equation systems from Sections 3.4 and 4.2. In fact, providing these equation systems is one of the contributions of this paper.

EigenTrust was designed for *distributed* computation of global trust values in a peer-to-peer network with minimal overhead. We have not investigated this aspect. For the purposes of this paper, we have assumed that some central authority can collect relevant certificates and carry out the computation. In [10], it was shown how authorization questions in SPKI/SDSI can be solved when the relevant certificates are distributed among multiple sites. Our system is also based on SPKI/SDSI, so it is conceivable that ideas from [10] could be lifted to SDSIrep.

We assume that the certificates used in the computations represent the current preferences of the users, and therefore the results of our algorithms reflect the current situation. It is conceivable that users' preferences change over time, and that they will eventually want to redistribute their trust values to reflect their new preferences. (Analogous effects occur, e.g., in PageRank or EigenTrust.) Such dynamics are beyond the scope of this paper. For our purposes, we simply assume that there exists some mechanism that allows the users to manage their certificates and make their current certificates available to the computation engine.

References

1. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. In: Decision Support Systems (2005)
2. Adler, T., de Alfaro, L.: A content-driven reputation system for the Wikipedia. In: Proc. 16th WWW Conference, ACM, pp. 261–270 (2007)
3. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylönen, T.: RFC 2693: SPKI Certificate Theory. In: The Internet Society (1999)
4. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust algorithm for reputation management in P2P networks. In: Proc. 12th WWW Conference (2003)
5. Jha, S., Reps, T.: Model checking SPKI/SDSI. JCS 12(3–4), 317–353 (2004)
6. Schwoon, S., Jha, S., Reps, T., Stubblebine, S.: On generalized authorization problems. In: Proc. CSFW, pp. 202–218. IEEE, Los Alamitos (2003)
7. Esparza, J., Kučera, A., Mayr, R.: Model checking probabilistic pushdown automata. In: LICS 2004, IEEE, Los Alamitos (2004)
8. Li, N., Mitchell, J.C.: Understanding SPKI/SDSI using first-order logic. In: Proc. CSFW, pp. 89–103. IEEE, Los Alamitos (2003)
9. Suwimonteerabuth, D., Schwoon, S., Esparza, J.: Efficient algorithms for alternating pushdown systems with an application to the computation of certificate chains. In: Graf, S., Zhang, W. (eds.) ATVA 2006. LNCS, vol. 4218, pp. 141–153. Springer, Heidelberg (2006)
10. Jha, S., Schwoon, S., Wang, H., Reps, T.: Weighted pushdown systems and trust-management systems. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 1–26. Springer, Heidelberg (2006)
11. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project (1998)
12. Etessami, K., Yannakakis, M.: Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In: Diekert, V., Durand, B. (eds.) STACS 2005. LNCS, vol. 3404, Springer, Heidelberg (2005)
13. CiteSeer: Top 10,000 cited authors in computer science `http://citeseer.ist.psu.edu/allcited.html`
14. CiteSeer: Estimated impact of publication venues in computer science `http://citeseer.ist.psu.edu/impact.html`
15. Hirsch, J.E.: An index to quantify an individual's scientific research output. Proceedings of the National Academy of Sciences 102, 165–169 (2005)
16. Ley, M.: DBLP bibliography, `http://www.informatik.uni-trier.de/~ley/db/`
17. Jøsang, A., Marsh, S., Pope, S.: Exploring different types of trust propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)