Decentralized Trust Management; Exam 06/12/2007

Questions (each answer is worth 3.5 points)

1. Recall the definition of the RT_0 grammar.

   - *Simple Member*: $A.r \longleftarrow D$ With this statement A asserts that D is a member of $A.r$.
   - *Simple Inclusion*: $A.r \longleftarrow B.r_1$: With this statement A asserts that $A.r$ includes (all members of) $B.r_1$. This represents a delegation from A to B, as B may add principals to become members of the role $A.r$ by issuing statements defining $B.r_1$.
   - *Linking Inclusion*: $A.r \longleftarrow A.r_1.r_2$. With this statement A asserts that $A.r$ includes $B.r_2$ for every B that is a member of $A.r_1$.
   - *Intersection Inclusion*: $A.r \longleftarrow B_1.r_1 \cap B_2.r_2$ With this statement A asserts that $A.r$ includes every principal who is a member of both $B_1.r_1$ and $B_2.r_2$.

   Write an RT_0 model of the following situation: Jan runs a virtual community site devoted to exchanging information related to music in which he has a database of friends (defined by $Jan.friends \longleftarrow Alice$, $Jan.friends \longleftarrow Bob$, etcetera). Jan's friends each keep a database of nice songs (e.g. $Alice.nicesong \longleftarrow \ldots$).

   Some of these friends are student of the local conservatory (e.g. $ConservatoryEnschede.student \longleftarrow Alice$).

   Jan keeps a list of good conservatories ($Jan.goodconservatory \longleftarrow \ldots$). Some other friends of Jan are not conservatory students, but Jan keeps them in high consideration w.r.t. their music taste ($Jan.goodtaste \longleftarrow \ldots$). Now Jan wants to define a new group of songs which get "high priority". These are the songs are considered "nice" both by a friend of Jan who studies in a conservatory AND (b) by a friend of Jan who has "good taste" AND (c) by a teacher at a good conservatory.

2. A consortium of webshops offers a promotions program. When a user a registers for the program at company C, her email address, email(a), and interests, interests(a), are stored and the company C and 'selected partners' may send up to three promotions a month. Company C implements this by issuing upto three tokens a month. Below is the policy syntax of the permission logic ALFA with some user defined predicates and actions for use in this setting.

$$\begin{aligned}\phi \quad ::= \quad & \bot \mid a \text{ says } \phi \text{ to } b \mid \phi \wedge \phi \mid \phi \rightarrow \phi \mid \texttt{?Act} \mid \texttt{!Act} \mid \forall x : \phi \mid \\ & \text{isPromotion}(d) \mid \text{isSP}(a) \mid \text{mayMail}(a, d) \mid \text{mayRead}(d)\end{aligned}$$

   The expressions `?Act` and `!Act` respectively denote actions that need to be done once and every time a permission is used, the predicates mayMail(a, d) and mayRead(d) respectively describe that a mail containing d may be sent to user a and the user data d may be read while isPromotion(d) describes that d is a promotional offer complying to the consortium's promotion rules and isSP(a) expresses that d is a selected partner.

   The data items (d) in this setting are promotions, addresses and interests. The actions Act include: `obtainToken`(x): obtain a token for emailing customer x, with empty ($\bot$) proof obligation, `mail`($address, d$): send promotion d to $address$ with proof obligation mayMail($address, d$), `read`(d): reading user data d with proof obligation mayRead(d) and `notify`(x): provide a notification to the user, with empty proof obligation.

   (a) Describe the following ALFA policy in words:

$$\begin{aligned}\text{isSP}(x) \rightarrow C \text{ says } \texttt{!obtainToken}(y) \rightarrow \text{isPromotion}(d) \rightarrow \\ \text{mayMail}(email(y), d) \text{ to } x\end{aligned}$$

   (b) Give an ALFA formula for the following policy: "The company C may allow selected partners to read the interests of users provided the partner notifies the user once."

(c) Give a scenario in which a partner company P, correctly following the policies, sends a promotion to user *Alice* after finding out her interests. Give the policies involved, the actions that happen and the additional information that is logged with the actions.

(d) In the setting of ALFA there are two notions of decidability that play a role: that of proof finding and that of proofs checking. For both cases give an example of a problem that could occur (in the setting of this question) if we do not have a decision algorithm and conclude which is more important in this setting.

3. The success of the Google search engine can be traced back to the application of reputation-based trust management.

(a) How did Google improve on previous search engines like Altavista, and how does reputation-based trust management fit into this picture?

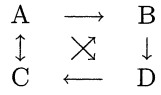
(b) Google's search engine is based on the PageRank algorithm, which can be described by iterating the following formula:

$$r(u) = ds(u) + (1-d) \sum_{v \in N^-(u)} \frac{r(v)}{|N^+(v)|}.$$

Explain the meaning of each symbol in the above formula.

(c) Explain the random surfer model, and what it has to do with the PageRank algorithm.

(d) Assume we have a (very) small intraweb, consisting of the webpages A,B,C and D. They have links to each other as follows:



Using $d = 0.5$, and $s(u) = (1, 0, 0, 0, 0)$ (A is the start page and gets the complete initial score), calculate three iterations of the rank of each page.

(e) As presented above, PageRank is a centralized algorithm. How could you make it distributed? (Hint: think of Eigentrust).