Online Appendix for "Role Refinement in Access Control: Model and Analysis"

Hao Xia, Milind Dawande, Vijay Mookerjee

Naveen Jindal School of Management, The University of Texas at Dallas, Richardson, Texas 75080

hao.xia@student.utdallas.edu, milind@utdallas.edu, vijaym@utdallas.edu

Proofs of Technical Results

LEMMA 1. *If a feasible solution exists for RRP, then Algorithm* GREEDY *generates one such solution in a finite number of iterations.*

Proof of Lemma 1. Initially, each existing role $R_i \in \mathscr{R}$ is exactly the union of some subset of its neighbors (i.e., sets $C_j \in \mathscr{C}$ such that edge $e(R_i, C_j)$ exists) in the bipartite graph B. Otherwise, clearly no feasible solution exists. In each subsequent iteration, only the permissions covered by the chosen candidate C are removed from each set $R \in \mathscr{R}$. Thus, in any iteration, an updated set R_i is always a subset of the union of its neighbors in B. Consequently, as long as there exists a $R_i \neq \emptyset$ in $\mathscr{R}$, we are guaranteed to find at least one candidate C_j such that $w_{ij} \geq 1$ and, thus, $w_j \geq 1$. This, in turn, guarantees that the greedy algorithm can pick one candidate and remove at least one permission from one of the roles in $\mathscr{R}$ in each iteration, until the updated system $\mathscr{R} = \emptyset$. The number of iterations required is finite, since both the number of roles and the number of permissions in each role are finite. ■

THEOREM 1. *Algorithm* GREEDY *is a $O(\ln M)$-factor, polynomial-time algorithm for RRP.*

Proof of Theorem 1. Consider Iteration t, when candidate C_j is picked. The remaining $M(t)$ permissions in (the updated system) $\mathscr{R}$ can always be fully covered by the optimal solution OPT. Therefore, the cost to cover these $M(t)$ permissions is at most OPT^c. The chosen candidate C_j has the lowest cost per unit coverage $c(C_j)/w_j$ over all the sets in $\mathscr{C}$. Thus,

$$\frac{c(C_j)}{w_j} \leq \frac{OPT^c}{M(t)}$$

Note that the price of each of the w_j permissions covered in Iteration t is $c(C_j)/w_j$. In our assumed ordering (defined above), let e_ℓ be the first permission to be covered in Iteration t. Then, the number of permissions remaining (to be covered) at the beginning of Iteration t is at least $M - \ell + 1$. That is, $M(t) \geq M - \ell + 1$. Since $p_\ell = c(C_j)/w_j$, it follows that

$$p_\ell \leq \frac{OPT^c}{M - \ell + 1}$$

Summing up the prices of all the M permissions, we obtain a bound on the cost GRD^c of Algorithm GREEDY:

$$GRD^c = \sum_{i=1}^{M} p_i \leq (1 + \frac{1}{2} + \frac{1}{3} + \ldots + \frac{1}{M})OPT^c \leq (1 + \ln M)OPT^c$$

To see that Algorithm GREEDY is a polynomial-time procedure, we recall that $|\mathscr{C}| = n$ and $|\mathscr{R}| = k$. The number of iterations is clearly no more than n, since each iteration picks one candidate from the set $\mathscr{C}$. Let $u = \max_{1 \leq j \leq n; 1 \leq i \leq k} (|C_j||R_i|)$. The time complexity of checking whether $C_j \subseteq R_i$ in the initial construction of the bipartite graph $B(\mathscr{R}, \mathscr{C}, \mathbf{E})$ is $O(u)$. Thus, the time complexity of building the initial graph B is $O(nku)$.

Similarly, the time required for updating the graph in each iteration is also $O(nku)$. Thus, the algorithm runs in time $O(n^2ku)$, which is polynomial in the size of the input. ∎

THEOREM 2. *Algorithm* RANDOMIZED ROUNDING *is a polynomial-time algorithm for RRP which achieves the following two properties: (i) its output* $\mathscr{S} = \cup_{\theta=1}^{\lceil 2\ln M\rceil}\bar{\mathscr{S}}_\theta$ *is a feasible solution to RRP with high probability for sufficiently large* M*. In particular, the probability that* $\mathscr{S}$ *is feasible approaches 1 as* $M \to \infty$*, (ii) the expected cost of* $\mathscr{S}$ *is* $O(\ln M)$ *times the cost of the optimal solution.*

Proof of Theorem 2. The output of Algorithm RANDOMIZED ROUNDING is the union of $\lceil 2\ln M\rceil$ potential solutions to RRP $\bar{\mathscr{S}}_\theta, \theta = 1, 2, \ldots, \lceil 2\ln M\rceil$. The probability that any specific permission, say e_ℓ, is not covered by any one potential solution to RRP is no more than $1/e$. Therefore,

$$\textbf{Prob}[e_\ell \text{ is not covered by } \mathscr{S}] \le \left(\frac{1}{e}\right)^{\lceil 2\ln M\rceil} \le \frac{1}{M^2}$$

$\mathscr{S}$ is a feasible solution to RRP if it covers all $e_\ell, \ell = 1, 2, \ldots, M$. Thus,

$$\textbf{Prob}[\mathscr{S} \text{ is a feasible solution to RRP}] \ge 1 - M\frac{1}{M^2} = 1 - \frac{1}{M}$$

Consequently, as $M \to \infty$, the probability that $\mathscr{S}$ is feasible for RRP tends to 1. Furthermore, $\mathbf{E}[\text{cost}(\mathscr{S})] \le \mathbf{E}[\lceil 2\ln M\rceil \cdot \text{cost}(\bar{\mathscr{S}})] \le \lceil 2\ln M\rceil \cdot OPT^c$.

It is easy to see that Algorithm RANDOMIZED ROUNDING is a polynomial-time procedure. The time required to obtain the LP formulation and the time to solve it are both polynomial in the size of the input, with the latter being a well-known result in the literature; see, e.g., Martin (1999), Nemhauser and Wolsey (1988). The cardinality of each potential solution is at most n and, hence, the time required to generate $\lceil 2\ln M\rceil$ of them and compute their union is $O(n\ln M)$, which is also polynomial in the input size. ∎