Risk-centred role engineering in identity management audits – An approach for continuous improvement of the access control model and possible risk accumulations

Sebastian Kurowski[1]

Abstract: Success and costs of audits in identity management largely depend on the structure of the underlying access control model. Auditing access rights includes the determination of actuality and adequacy of provided access rights. In order to ease audit and administration of access rights, role mining approaches have provided several solutions for identifying a minimal set of roles based upon either existing usage data, or business data. However, these approaches have focused on homogeneous, static environments. When facing dynamic, heterogeneous environments, such as infrastructure administration or smart systems, the accompanied noise of access rights provisioning hinder the determination of adequacy and actuality of access rights. With application of static approaches, accumulation of access risks at users may arise due to inadequate access rights, or aggregation of access roles. These issues are however mostly neglected by current approaches. Within this contribution we propose a method based upon the design structure matrix approach, which enables the identification of role aggregations, and examination of access risk accumulation within aggregated roles, and their assigned users throughout continuous audits of the access control model.

Keywords: identity management, access control, role engineering, design structure matrix, smart systems, cloud computing, RBAC, role mining

1 Introduction

Within the identity data lifecycle [MR08] auditing of identity data, including access rights enables identification of old, or falsely provided access rights. If this process is carried out regularly, it enables the identification of mistakes and, thus mitigation of potential access risks, due to the provisioning of too many privileges, or privileges which are no longer required. The success and costs of such audits are largely depending on the underlying access control model. For instance, [OL10] find that audit and policy maintenance costs are nearly doubled when using a classical access control list (ACL) paradigm over role-based access control (RBAC), stating that employing roles makes "it easier to accomplish [the policy review and attestation process]". Yet, the same study finds that half of all companies still used ACL as a secondary access control alternative in a hybrid solution along with RBAC, in 62% of all cases for their directory services [OL10]. Also multiple issues, especially with hierarchical RBAC have been known

[1] University of Stuttgart, Competence Team Identity Management, Allmandring 35, 70565 Stuttgart, sebastian.kurowski@iat.uni-stuttgart.de

resulting out of the complexity of the reality projected by the access control model. For instance, across multiple circles-of-trust, the application of hierarchical RBAC may lead to unintended privilege escalations for some roles [JGZ11]. However, these issues do not arise out of formal issues with RBAC itself, but out of the aggregating characteristics of these models. The classic role of access control, as regulating interactions between subjects and objects leads to the technologies itself having to deal with potentially endless possibilities in the respective real world scenario. As Luhmann states "technologies [are usually] conceived as relations between cause and effect, confirmed by scientific knowledge or practical experience" [Lu90]. Yet, technology and as such also access control is more an encapsulating of causal dependencies between subjects and objects, which itself may be numerous and even potentially infinite in their variations. This mismatch yields dire consequences in the application of technology: "Paradoxically, we lose control of causalities, as they become too complex" [Lu90].

Paradigms such as RBAC, which aim at decreasing administrative complexity, by encapsulating causality between multiple subjects and objects, rather than encapsulating the causality of existence of one object (as in the case of ACL), transform this paradox to the following: While on the one hand RBAC results in significantly lower audit and administrative efforts, imperfections of the access control model may lead to conflicts with the reality it is applied to. This paradox may especially become visible, when dealing with environments characterized by highly heterogeneous and dynamic workflows, and objects, e.g. in privileged identity management for cloud based platform- and infrastructure-as-a-service environments. In current research this paradox has mostly been accounted for as noise, which results in an open research issue [Fr09] [KSS03] [Mo09] [VAG07] [ZRE03].

The noise created by applying of role mining approaches in dynamic and heterogeneous workflows (e.g. privileged identity management), further translates to noise in access risk distribution at the subjects. Aggregation of subject – object relationships, as in RBAC in such scenarios may lead to subjects inadequately gaining access rights and thus access risks to objects. Additionally, hierarchical role mining may lead to access risk accumulation, which, especially in heterogeneous environments with dynamic component introduction and access rights modifications, may be hard to examine.

In this contribution we therefore propose an approach for risk centred role mining by using the user assignment matrix UA, which can be obtained from the identity directory. In order to account for the noise, associated with the underlying access control model in dynamic environments, we transfer an approach from process and engineering planning [YB03]. This approach tries to determine the best order in inter-dependent engineering processes. By transferring the approach we are able to determine both, intra-dependent clusters of subject-object associations (e.g. possible candidates for role aggregations), along with their inter-dependencies upon other roles. Therefore, we aim to use it both for simplifying the access control models in order to ease the administration, and for identifying possible risk accumulations which may arise as a consequence of aggregating roles.

In the following we provide an overview on current role mining approaches (Section 2). Our used scenario along with our assumptions of access control administration in dynamic environments is introduced within Section 3. This scenario includes a definition of the risk consolidation problem which may arise within aggregated role systems (e.g. hierarchical RBAC). Finally, our approach is presented within Section 4, along with its application for clustering of possible role aggregations, and analysis of intra- and inter-dependencies of the role clusters (Section 5). The presented approach is implemented within an Excel spreadsheet, and will be provided to organizations for usage within this year. As our approach only requires access to data acquirable from the directory services, easier integration into existing identity and access management (IAM) audit commodities and tools is expected. It is therefore intended for use in the audit phase of the identity lifecycle [MR08].

2 State-of-the-Art

By using the Scopus[2] literature database, we found a total of 132 publications[3] regarding role mining and access control. Hereby the first publications are dated back to 2002. Since 2008 the academic interest in the topic has increased to about 15 publications, and since then shows only a small decline in interest, except for the years 2011, and 2014. However, this indicates that the topic of Role Mining has so far not been highlighted in academic research on access control. Interestingly, the decrease in interest aligns well, with the overall decrease in academic interest regarding Role-Based Access Control (RBAC)[4], and the slight decrease in academic interest regarding Access Control itself[5]. The reasons for this decrease in interest may not be subject to this paper. However, it is worthwhile noting that the decrease of interest in Role Mining might well be rooted within the mentioned paradox resulting out of causal encapsulation within role-based policies and the collision of the necessarily imperfect model with the real-world scenarios they are applied in. The contribution of Vaidya et. al. [VAG07] define various role mining problems, including the problem of finding a minimized set of roles which is consistent with user-permission assignments of the scenario, while the amount of roles is smaller than a defined threshold (DECISION RMP), consistent to a certain threshold with user-permission assignments of the scenario, while the amount is smaller than a defined threshold (δ-DECISION RMP), and consistent to a certain threshold, without any upper boundary on the amount of roles (DECISION MinNoise RMP). Hereby, the

[2] http://www.scopus.com: The search using the literature database was conducted in April 2016.

[3] The used search term was: TITLE-ABS-KEY (role mining) AND (LIMIT-TO (EXACTKEYWORD , "Access control"))

[4] The search term TITLE-ABS-KEY (rbac) AND (LIMIT-TO (EXACTKEYWORD , "Access control")) provides a sum of 1656 publications. However, analysis shows, that the amount of publications have decreased by nearly 50% in 2015 (104 publications), compared to the peak-of-interest in 2008 (189 publications).

[5] The search term TITLE-ABS-KEY (access control) AND (LIMIT-TO (SUBJAREA , "COMP")) provides a sum of 53148 publications. However, the amount of publications has slightly decreased from its peak-of-interest in 2008 (4711 publications), to 3996 publications in 2015.

DECISION RMP is of little interest for our contribution, as it mostly applies to static environments. In dynamic environments a perfect consistency with the user-permission assignments is very likely to be impossible [VAG07]. All problems are shown to be NP complete [VAG07]. One possible problem solving proposition is found in the application of the minimum tiling problem [GGM04]. But, since this problem aims at finding the minimum set of tiles consistent with the original Boolean dataset of user and role assignments, it is not applicable to hierarchical RBAC. Furthermore, the MinNoise RMP, which aims at finding an imperfect set of roles, and thus is more likely to be applicable in dynamic scenarios [VAG07], is shown to be transferrable to the NP-hard discrete basis problem, considering segmentation of Boolean matrices [Mi08]. The NP-completeness and even NP-hard character of non-hierarchical Role Mining Problemes indicate the appropriateness for meta-heuristic, over exact searching algorithms. Furthermore, it favours user-centric, organizational approaches, where Role Mining is considered within a process during regular audit. Kuhlmann et. al. [KSS03] present a Role Mining approach, which enables a bottom-up approach by mining data from different Access Control Systems. The approach being presented mines access control data, and therefore presents a similar approach to this contribution. Their approach uses the IBM Intelligent Miner for Data, for segmentation and mining of User-Role assignments. However, the approach in [KSS03] considers flat RBAC, whereas this contribution aims at administration and continuous improvement of hierarchical RBAC. Finally, [Fr09] provide a probabilistic approach for role mining, and demonstrate this approach on large enterprises. Their approach is able to extract understandable roles, showing that most relevant information for mining business roles arises out of the organization unit, rather than the job description itself. By applying an objective function, which penalizes the assignment of roles to a user which does not share the same business information as other users assigned to that role, business information can be incorporated into the role mining process. However, the application of this approach for our scenario is highly unlikely. The proposed introduction of business information requires the set of information implying authorization rights to be at least partially disjunctive. However, in our scenario this is not necessarily the case.

Our considerations show, that the NP-completeness of the Role Mining problem [VAG07], requires further consideration of non-exact search and organizational integration of such techniques. Approaches, such as [Fr09] or [KSS03] are merely applicable for hierarchical RBAC. However, hierarchical roles may provide the necessary decrease in complexity, in order to administer highly dynamic scenarios.

3 Scenario

Our scenario considers hierarchical RBAC in a dynamic, and heterogeneous environment, such as in privileged access of IT service operations. If we consider, for instance cloud service providers, multiple different systems, such as VM Hypervisors, Logging components, Privileged Identity and Access Management (IAM), firewalls, VPN gateways, FTP servers, Fileshares, and Workflow engines are possible objects, accessed by privileged users. Classically, privileged IAM focuses on accountability of actions [Ja11], rather than minimization of provided accesses. However, as the infrastructural, and thus the amount of involved administrators increases, formalization of workflows and responsibilities may yield groups, which are neither disjunctive nor equal in terms of access rights. This means, that while e.g. administrative teams for managing the security infrastructure, and administrative teams for managing the VM hypervisors may exist, both may be required to access similar systems, e.g. the logging system.

In such a scenario, we use the definition of the hierarchical RBAC, as in [Sa96]. Hereby we omit the concepts of sessions and separation of duties (SOD). We argue, that sessions can be omitted for the sake of simplicity, as we do not aim on optimizing the currently used roles, but ease administration of assigned roles for the respective subject. The concept of SOD is omitted since it does not fully extend to the issue of risk consolidation, requiring duties to be formalized and thus disjunctive in nature. However, this is very likely not the case in our scenario. Therefore we yield the following definition of the RBAC [Sa96]:

- Users, Roles, and Permissions: U,R, and P
- A many-to-many permission to role assignment relation: $PA \subseteq P \times R$
- A many-to-many user to role assignment relation: $UA \subseteq U \times R$
- A partial order on R called the role hierarchy or role dominance relation : $RH \subseteq R \times R$

In our scenario we aim at improving the role structure throughout regular audit. Improving the role structure includes both minimization of the roles involved within the role structure [KSS03] [VAG07], and providing understandable roles, which match with the semantics of the regarded scenario [Fr09].

3.1 Role Structure Optimization and the risk accumulation problem

Additionally, improving the role structure requires consideration of possible access risks. We consider the access risks which occur with each permission as:

- A many-to-many permission to access risk relation: $AR \subseteq P \times Ri$

Especially, in our privileged access scenario, not all possible access risks combinations may be determinable, nor formalized. This is especially the case, if access risks materialize according to the accumulation principle which involves the combination of different lower risks leading to a high access risks, rather than the maximum principle which allows the formal application of the highest risk. If we denote access risks as a binary matrix, whereas a 1 indicates, that the access risks should be strictly separated, we are still only able to depict access risk accumulations between two risks. Accumulations between three risks, would require an additional matrix, for up to four possible access risks two additional matrices, and so on. The possible solution space therefore grows rapidly, and may quickly become hard to administer.

As the problem of finding a minimal set of roles itself is NP-complete [VAG07], and the access risks being many to many regarding the respective permission, which itself are in a many-to-many relation with the respective roles, the issue of detecting access risk accumulations on a full role list is hardly expectable.

Additionally, access risks accumulate at the user itself, due to multiple access rights assignment to the same set of users. This means, that information on the user, the roles, and the assignment is required to identify accumulated access risks.

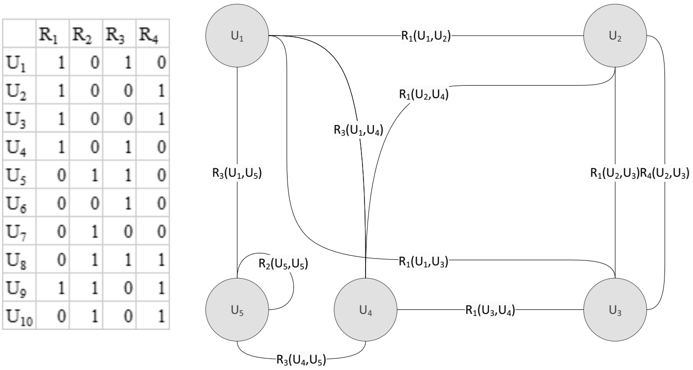
Following the definition by [Sa96], hierarchical RBAC enables easier maintenance by providing a superset of permissions. This means, that roles accumulate the permissions along their hierarchy. For the detection of accumulated access risks, this means that only a subset of all roles, a subset of permissions, and thus a subset of access risks must be determined, enabling the accumulation of access risks to become manageable.

Current approaches for mining RBAC roles however, either neglect the mining of hierarchical RBAC, such as [Fr09] [KSS03] [VAG07], or abstract information on the user by edge minimization between user and role nodes, such as [ZRE03].

Our approach therefore uses a different graph construction approach as [ZRE03], by providing a Graph with only one type of nodes (the users), wherein each role assignment becomes an edge. This graph is being clustered and analysed using the design structure matrix method, enabling detection of interdependencies between clusters and in clusters.

4 Approach for risk centred role mining

In our scenario, business data such as the organisational unit is very likely to not provide sufficient mutual information on roles, as in [Fr09] [KSS03]. Therefore we largely focus on role mining approaches, which use the UA as a source for role mining, such as [VAG07] [ZRE03]. Both approaches consider the UA as input, and either aim at clustering the UA [VAG07], or try to minimize the edges in a graph built from different types of nodes (roles, and users) [ZRE03]. One key issue with these approaches for risk centred mining of hierarchical roles, is that neither provides sufficient information on possible risk consolidations, and conflicts with user-role assignments of the mined roles.



	R_1	R_2	R_3	R_4
U_1	1	0	1	0
U_2	1	0	0	1
U_3	1	0	0	1
U_4	1	0	1	0
U_5	0	1	1	0
U_6	0	0	1	0
U_7	0	1	0	0
U_8	0	1	1	1
U_9	1	1	0	1
U_{10}	0	1	0	1

Figure 1 Overview on the graph based approach. The edges depict the mutual group assignments of roles (Right). The corresponding UA is shown on the left

By using graph optimization as in [ZRE03], the followed approach of minimizing the edges [Mo09] most likely aggregates user assignments, and does not allow further follow up on possible risk consolidations at users due to a role aggregation. The same holds for [VAG07], where the NP-completeness of the underlying non negative matrix factorization issue [Va09] results in various possible solutions, and may thus make it impossible to follow-up on risk and access rights consolidations occurring out of role aggregation.

We therefore follow a slightly different, yet graph-based approach, in order to enable the evaluation of access risk consolidations in our current role structure, and in the aggregated role structure. This means, that we do neither aggregate roles, users, nor assignments in our analysis. This is achieved by defining the UA as a Graph, in which each user is depicted as a node, and each mutual role assignment between users results in an edge. Figure 1 visualizes this approach. The matrix indicated on the left, shows an example UA, which corresponds to a binary matrix, indicating 1, if a role is assigned to a user, and 0 otherwise.

Based on the UA, we extract the corresponding graph of mutual role assignments, by creating an edge for each shared role. In our example, shown in Figure 1, this means that role R1 results in 8 edges (4 edges between U1, U2, U3, and U4, 3 edges between U2, U3, and U4, and 1 edge between U3, and U4). The present data now enables us to follow three different strategies: (1) clustering of the resulting incidence matrix of the graph I(G); (2) clustering of the resulting adjacency matrix of the graph, and (3) clustering of the adjacency matrix of the line graph. As we want to preserve information on possible role assignments, and possible risk consolidations, Strategy (2) would only provide information on which users share mutual roles, lacking information on which roles are shared mutually by which users, and thus not enabling further risk analysis. Strategies (1) and (3) preserve the required information. However, since each edge can be assigned to up to two user nodes, and one role, strategy (3) promises the most efficient handling of our issue.

Therefore, we obtain the line graph in the next step, indicating which edges correspond together, and thus which roles are assigned mutually at which users. The line graph can be obtained by:

$$Adj(G)^L = I(G)' * I(G) - 2 * I_{|V|}$$

Where $Adj(G)^L$ indicates the adjacency matrix of the line graph, I(G) the incidence matrix of the graph, and $I_{|V|}$ indicates the identity matrix in the size of the resulting number of edges in our graph.

5 Clustering of mutual roles and analysis

As $Adj(G)^L$ is a symmetric matrix, indicating mutual roles, and in a broader term interactions between roles, we can apply the methodologies around design structure matrices [YB03], This approach is mostly used within process and product planning, and knowledge management, and is applied in scenarios where multiple interactions between components occur. Hereby the resulting interaction matrices, which indicate whether two components interact, are being clustered, and analysed. Clustering hereby takes both internal and external dependencies into account. As our approach seeks to find mutually shared roles and risk consolidations, using an approach that identifies both internal and external dependencies throughout the mining of hierarchical RBAC roles seems reasonable.

We therefore follow this approach, by clustering $Adj(G)^L$, using the algorithm described in Section 5.1, followed by an analysis of the resulting internal, and external dependencies of the resulting role clusters (see Section 5.2).

5.1 Clustering of the Role Assignment Adjacencies

For clustering of the role assignment adjacencies in $Adj(G)^L$, we follow the algorithm proposed by [Id95] along with the refinements by [Th01]. The following pseudo-code describes the process of clustering within the role assignment adjacencies:

```
Clusters[] <- Initialize each element in role assignment
adjacencies as own Cluster;
Do While (iteration < abortIteration)
                  And (deltaCosts < abortDelta)
    clusterCurrent <- Pick a cluster from Clusters[]
    u.a.r.;
    totalCosts <- Calculate Costs according to equation
 2, 3 and 4;
    highestBiddingCluster <- calculate Bids from all
    Clusters in Clusters[] according to equation 1 to
```

```
      clusterCurrent and pick the highestBiddingCluster;
      If (highestBid = random_Bet)
                     highestBiddingCluster <- Pick second
                     highest Bidding Cluster;
      fi
      clusterJoined <- Join clusterCurrent and
      highestBiddingCluster;
      newCosts <- Calculate Costs;
      r <- Pick integer r u.a.r.;
      If (newCosts < totalCosts) Or (r < acceptThreshold)
                 Clusters[] <- Remove clusterCurrent and
                 highestBiddingCluster and Add
                 clusterJoined;
      Else
                 Clusters[] <- Restore old Cluster List;
      fi
      iteration <- iteration + 1;
   Od
```

The algorithm by [Ja11] [Th01] requires two basic input parameters. random_Bet specifies a random guess by the user, and enables the algorithm to take the second best option in search for an optimum, rather than following the best option as in greedy strategies. The second algorithm, with the same intention, enables the algorithm to randomly accept solutions which result in a worse solution than the one of the previous iteration. As such, the algorithm roughly follows the path of simulated annealing [Hw88], which partially accepts worse solutions, rather than following a greedy strategy by only accepting the best solution during an iteration. As the role mining problem, and the problem of finding clusters in non-binary matrices are both NP-complete [VAG07] [Va09], reliance on local optima in finding optimal solutions will most likely result in the algorithm becoming stuck in a local optima. Therefore, following the approach of simulated annealing within the clustering, as done by [Id95] [Th01] provides a promising approach.

For determining the elements which should be joined towards a new cluster, the algorithm is using bids. These bids of a cluster j to a cluster k, depend on the internal dependencies of the newly joined cluster (j,k) and punish the size of the bidding cluster j. The bids are defined in [Th01] as:

$$\mathrm{ClusterBid}_{j,k} = \frac{\mathrm{DSM}_{j,k}^{\mathrm{powDep}}}{\mathrm{ClusterSize}_{j}^{\mathrm{powBid}}}$$

Equation 1 Function for determining the bid of a cluster j to a cluster k

Where the bid is considered as a bid from cluster j to cluster k. DSM depicts the internal dependency of the newly arranged cluster. In our case, DSM is the sum of adjacencies within the cluster in $\mathrm{Adj}(G)^{L}$. ClusterSize indicates the size of the bidding cluster j. Both the internal dependency DSM and the size of the bidding cluster are additionally adjusted by using the input variables powDep, and powBid. powDep determines the weight of the internal dependency of the newly arranged cluster over the bidding cluster's size. powBid on the other hand emphasizes the role of the bidding clusters size. If larger cluster should be obtained, this value should be lower than the input value powDep, and vice versa.

The target function of the clustering algorithm aims at minimizing the overall costs. These are defined as:

$$TotalCosts = \sum IntraClusterCost + \sum ExtraClusterCost$$

Equation 2 Function for determining the overall costs

The IntraClusterCost hereby indicate the cost function within each cluster, which considers both the internal dependencies DSM of the cluster, and the cluster size (ClusterSize).

$$IntraClusterCost_j = DSM_j * ClusterSize^{powCC}$$

Equation 3 Function for determining the internal costs of a cluster j

Hereby the size of each cluster is being punished by using the exponent powCC, which is being defined by the user. In order to calculate the Extra Cluster Costs, the algorithm originally determines the dependencies between each cluster [Id95]. However, as the DSM value in our case is being defined, as the sum of adjacencies, and as our goal is to minimize the adjacencies outside of our clusters, we define the Extra Cluster Costs as:

$$ExtraClusterCost_j = DSM_{i,j} * DSMSize^{powCC}$$

Equation 4 Function for determining the external costs of a cluster j

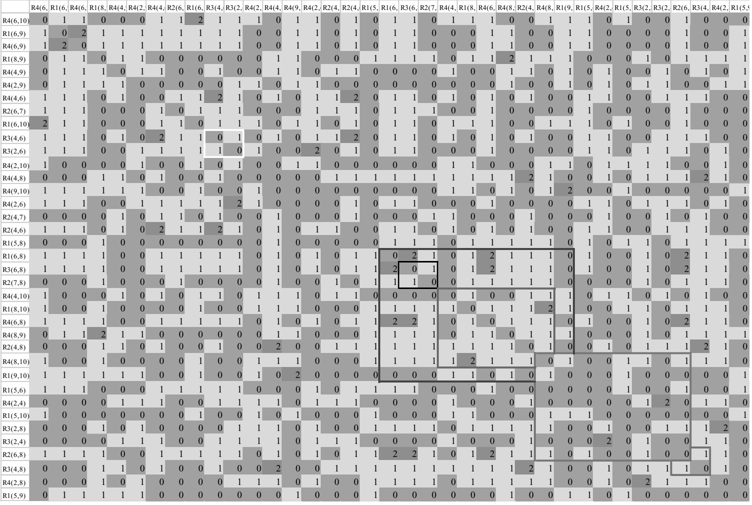
Where $DSM_{i,j}$ defines the dependability of all elements outside of the cluster, but within the rows i, or the columns j of the cluster. DSMSize considers the amount of external elements which depend upon the cluster j. These elements reside within the rows i, or the colums j of the cluster. As with the IntraClusterCosts the weight of the amount of external dependencies is emphasized by the value powCC.

This cluster algorithm can be executed for a maximum amount of iterations, or until the cost function converges.

5.2 Analysing the Clustered Role Assignment Adjacencies

After clustering of the role assignment adjacencies a matrix, such as in Figure 2 is obtained. The clusters are indicated within the squares, showing which user role assignment could be aggregated, and which risk accumulations should be further examined. If a dependency is indicated with the value 0 no association between the role assignments is given (meaning, that these roles are not associated with the same subjects). A value of 1 indicates, that one of the roles is assigned to both subjects (e.g. in Figure 2, role R1 is associated both with subjects 6, and 9. Subject 6 is additionally associated with role R4). Finally, a value of 2 indicates that both roles are assigned with both subjects are associated with both roles (e.g. in Figure 2, subjects 6 and 8 are both associated with roles R1 and R3). The algorithm flips rows and columns according to the algorithm introduced in Section 5.2, and tries to establish clusters along the diagonal line of the adjacency matrix.

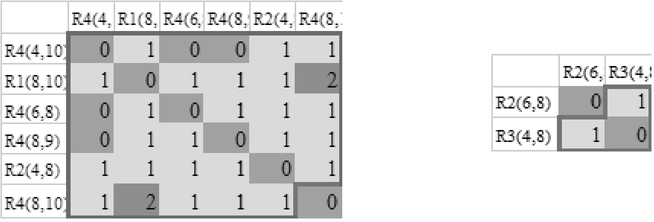
For further analysis of the obtained role assignment adjacencies, we must consider both the internal structure of the cluster, and the external structure of the elements within the rows and columns, but outside of the cluster. In this section, we will therefore discuss different observations that can occur within the clustered role adjacency matrix, and their implications.



	R4(6,	R1(6,	R4(6,	R1(8,	R4(4,	R4(2,	R4(4,	R2(6,	R1(6,	R3(4,	R3(2,	R4(2,	R4(4,	R4(9,	R4(2,	R2(4,	R2(4,	R1(5,	R1(6,	R3(6,	R2(7,	R4(4,	R1(8,	R4(6,	R4(8,	R2(4,	R4(8,	R1(9,	R1(5,	R4(2,	R1(5,	R3(2,	R3(2,	R2(6,	R3(4,	R4(2,	R1(5,9
R4(6,10)	0	1	1	0	0	0	1	1	2	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	1	0	0	0
R1(6,9)	1	0	2	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1
R4(6,9)	1	2	0	1	1	1	1	1	1	1	1	0	0	1	1	0	1	0	1	1	0	0	0	1	1	0	0	1	1	0	0	0	0	1	0	0	1
R1(8,9)	0	1	1	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	2	1	1	1	0	0	0	1	0	1	1	1	1
R4(4,9)	0	1	1	1	0	1	1	0	0	1	0	0	1	1	0	1	1	0	0	0	0	1	0	0	1	1	0	1	0	1	0	0	1	0	1	0	1
R4(2,9)	0	1	1	1	1	0	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	1	0	0	1	1
R4(4,6)	1	1	1	0	1	0	0	1	1	2	1	0	1	0	1	1	2	0	1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0
R2(6,7)	1	1	1	0	0	0	1	0	1	1	1	0	0	0	1	1	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0
R1(6,10)	2	1	1	0	0	0	1	1	0	1	1	1	0	1	1	0	1	0	1	1	0	1	1	1	0	0	1	1	1	0	1	0	0	1	0	0	0
R3(4,6)	1	1	1	0	1	0	2	1	1	0	1	0	1	0	1	1	2	0	1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0
R3(2,6)	1	1	1	0	0	1	1	1	1	1	0	1	0	0	2	0	1	0	1	1	0	0	0	1	0	0	0	0	1	1	0	1	1	1	0	1	0
R4(2,10)	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	1	1	1	1	0	0	1	0
R4(4,8)	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	2	1	0	0	1	0	1	1	1	2	1	0
R4(9,10)	1	1	1	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	2	0	0	1	0	0	0	0	0	1
R4(2,6)	1	1	1	0	0	1	1	1	1	1	2	1	0	0	0	0	1	0	1	1	0	0	0	1	0	0	0	0	1	1	0	1	1	1	0	1	0
R2(4,7)	0	0	0	0	1	0	1	1	0	1	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	0	0	1	0	0	1	0	1	0	0
R2(4,6)	1	1	1	0	1	0	2	1	1	2	1	0	1	0	1	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	0	1	1	1	0	0
R1(5,8)	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0	1	0	1	1	0	1	1	1	1
R1(6,8)	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	1	1	0	2	1	0	1	2	1	1	1	0	1	0	0	1	0	2	1	1	0
R3(6,8)	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	1	1	2	0	1	0	1	2	1	1	1	0	1	0	0	1	0	2	1	1	0
R2(7,8)	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	1	0	1	1	1	0	0	1	1	1	1	1	0	0	0	0	1	0	1	1	1	0
R4(4,10)	1	0	0	0	1	0	1	0	1	1	0	1	1	1	0	1	1	0	0	0	0	0	1	0	0	1	1	1	0	1	1	0	1	0	1	0	0
R1(8,10)	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	1	1	1	1	0	1	1	1	2	1	0	0	1	1	0	1	1	1	0
R4(6,8)	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	1	1	2	2	1	0	1	0	1	1	1	0	1	0	0	1	0	2	1	1	0
R4(8,9)	0	1	1	2	1	1	0	0	0	0	0	0	1	1	0	0	0	1	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	1	1	1	1
R2(4,8)	0	0	0	1	1	0	1	0	0	1	0	0	2	0	0	1	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	1	1	2	1	0
R4(8,10)	1	0	0	1	0	0	0	0	1	0	0	1	1	1	0	0	0	1	1	1	1	1	2	1	1	1	0	1	0	0	1	1	0	1	1	1	0
R1(9,10)	1	1	1	1	1	1	0	0	1	0	0	1	0	2	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	1
R1(5,6)	1	1	1	0	0	0	1	1	1	1	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1
R4(2,4)	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	2	0	1	1	0
R1(5,10)	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1
R3(2,8)	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	2	0
R3(2,4)	0	0	0	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	0	0	0	1	0	0	0	1	0	0	0	2	0	1	0	0	1	1	0
R2(6,8)	1	1	1	1	0	0	1	1	1	1	1	0	1	0	1	0	1	1	2	2	1	0	1	2	1	1	1	0	1	0	0	1	0	0	1	1	0
R3(4,8)	0	0	0	1	1	0	1	0	0	1	0	0	2	0	0	1	1	1	1	1	1	1	1	1	1	2	1	0	0	1	0	1	1	1	0	1	0
R4(2,8)	0	0	0	1	0	1	0	0	0	0	1	1	1	0	1	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	0	2	1	1	1	0	0
R1(5,9)	0	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	1	0	0	0	0	0	0

Figure 2 Example for Role Assignment Adjacencies after Clustering. Clusters are indicated in squares. The clustering used the values (powDep = 1; powBid = 1; powCC = 2), maximum Iterations were 1500, and convergence criteria was set to 0.5.

5.3 Internal structure of the clusters

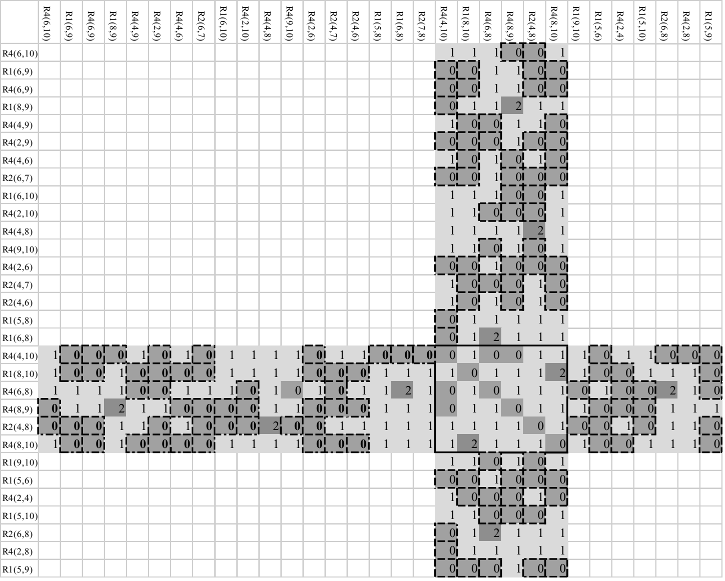


	R4(4,	R1(8,	R4(6,	R4(8,	R2(4,	R4(8,
R4(4,10)	0	1	0	0	1	1
R1(8,10)	1	0	1	1	1	2
R4(6,8)	0	1	0	1	1	1
R4(8,9)	0	1	1	0	1	1
R2(4,8)	1	1	1	1	0	1
R4(8,10)	1	2	1	1	1	0

	R2(6,	R3(4,
R2(6,8)	0	1
R3(4,8)	1	0

Figure 3 Examples of the internal cluster dependencies. The left shows a cluster with incomplete dependencies, indicating that users may gain additional access rights, and thus risks if the role is aggregated. The right shows a cluster with full internal dependency, indicating that all user assignments within this cluster would be achieved by aggregating the roles (in this case role R2, and R3).

The internal structure of the clusters provides insights into how many user assignments are actually affected by a possible role aggregation. Additionally, it may provide insights into which roles are currently assigned together, at which users. Figure 3 shows two possible cases of internal dependencies. The left case shows a cluster with incomplete internal dependency. Here the roles R1, R2, and R4 are being proposed for aggregation. However, since (except for the diagonal line), 0-elements are within the cluster, we can conclude that the aggregation of roles will result in users gaining additional access rights, and thus access risks.



	R4(6,10)	R1(6,9)	R4(6,9)	R1(8,9)	R4(4,9)	R4(2,9)	R4(4,6)	R2(6,7)	R1(6,10)	R4(2,10)	R4(4,8)	R4(9,10)	R4(2,6)	R2(4,7)	R2(4,6)	R1(5,8)	R1(6,8)	R2(7,8)	R4(4,10)	R1(8,10)	R4(6,8)	R4(8,9)	R2(4,8)	R4(8,10)	R1(9,10)	R1(5,6)	R4(2,4)	R1(5,10)	R2(6,8)	R4(2,8)	R1(5,9)
R4(6,10)																			1	1	1	0	0	1							
R1(6,9)																			0	0	1	1	0	0							
R4(6,9)																			0	0	1	1	0	0							
R1(8,9)																			0	1	1	2	1	1							
R4(4,9)																			1	0	0	1	1	0							
R4(2,9)																			0	0	0	1	0	0							
R4(4,6)																			1	0	1	0	1	0							
R2(6,7)																			0	0	1	0	0	0							
R1(6,10)																			1	1	1	0	0	1							
R4(2,10)																			1	1	0	0	0	1							
R4(4,8)																			1	1	1	1	2	1							
R4(9,10)																			1	1	0	1	0	1							
R4(2,6)																			0	0	1	0	0	0							
R2(4,7)																			1	0	0	0	1	0							
R2(4,6)																			1	0	1	0	1	0							
R1(5,8)																			0	1	1	1	1	1							
R1(6,8)																			0	1	2	1	1	1							
R4(4,10)	1	0	0	0	1	0	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	1	1	1	0	1	1	0	0	0
R1(8,10)	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	0	1	1	1	2	1	0	0	1	1	1	0
R4(6,8)	1	1	1	1	0	0	1	1	1	0	1	0	1	0	1	1	2	1	0	1	0	1	1	1	0	1	0	0	2	1	0
R4(8,9)	0	1	1	2	1	1	0	0	0	0	1	1	0	0	0	1	1	1	0	1	1	0	1	1	1	0	0	0	1	1	1
R2(4,8)	0	0	0	1	1	0	1	0	0	0	2	0	0	1	1	1	1	1	1	1	1	1	0	1	0	0	1	0	1	1	0
R4(8,10)	1	0	0	1	0	0	0	0	1	1	1	1	0	0	0	1	1	1	1	2	1	1	1	0	1	0	0	1	1	1	0
R1(9,10)																			1	1	0	1	0	1							
R1(5,6)																			0	0	1	0	0	0							
R4(2,4)																			1	0	0	0	1	0							
R1(5,10)																			1	1	0	0	0	1							
R2(6,8)																			0	1	2	1	1	1							
R4(2,8)																			0	1	1	1	1	1							
R1(5,9)																			0	0	0	1	0	0							

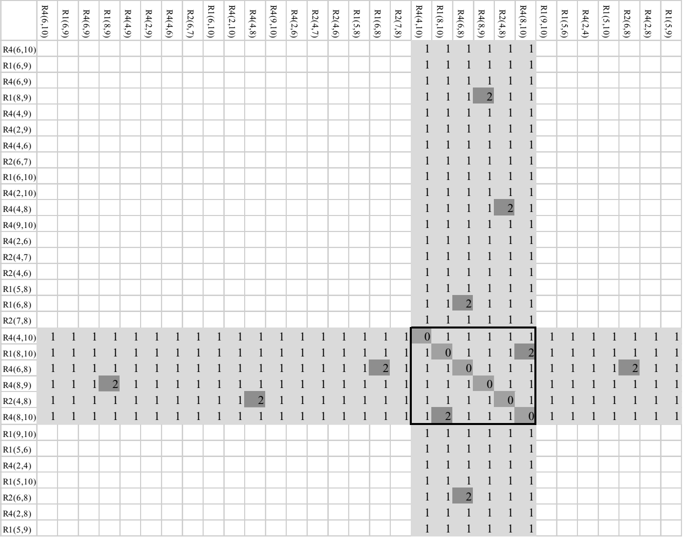
Figure 4 Cluster with incomplete dependency outside the cluster, indicating the increase of access

rights for users (marked with dashed lines) in the case of role aggregation

On the contrary, the right side of Figure 3 shows an example for a cluster with full internal dependency, in this case with the roles R2, and R3. This means that if the roles R2, and R3 would be aggregated no additional access rights, compared to the status quo would be provided to the users within this cluster.

5.4 Dependency outside the role cluster

The dependency outside the role cluster provides additional insights into possible access rights, and thus access risk increases in the case of a role aggregation.



	R4(6,10)	R1(6,9)	R4(6,9)	R1(8,9)	R4(4,9)	R4(2,9)	R4(4,6)	R2(6,7)	R1(6,10)	R4(2,10)	R4(4,8)	R4(9,10)	R4(2,6)	R2(4,7)	R2(4,6)	R1(5,8)	R1(6,8)	R2(7,8)	R4(4,10)	R1(8,10)	R4(6,8)	R4(8,9)	R2(4,8)	R4(8,10)	R1(9,10)	R1(5,6)	R4(2,4)	R1(5,10)	R2(6,8)	R4(2,8)	R1(5,9)
R4(6,10)																			1	1	1	1	1	1							
R1(6,9)																			1	1	1	1	1	1							
R4(6,9)																			1	1	1	1	1	1							
R1(8,9)																			1	1	1	2	1	1							
R4(4,9)																			1	1	1	1	1	1							
R4(2,9)																			1	1	1	1	1	1							
R4(4,6)																			1	1	1	1	1	1							
R2(6,7)																			1	1	1	1	1	1							
R1(6,10)																			1	1	1	1	1	1							
R4(2,10)																			1	1	1	1	1	1							
R4(4,8)																			1	1	1	1	2	1							
R4(9,10)																			1	1	1	1	1	1							
R4(2,6)																			1	1	1	1	1	1							
R2(4,7)																			1	1	1	1	1	1							
R2(4,6)																			1	1	1	1	1	1							
R1(5,8)																			1	1	1	1	1	1							
R1(6,8)																			1	1	2	1	1	1							
R2(7,8)																			1	1	1	1	1	1							
R4(4,10)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
R1(8,10)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	2	1	1	1	1	1	1	1
R4(6,8)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	0	1	1	1	1	1	1	1	2	1	1
R4(8,9)	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
R2(4,8)	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
R4(8,10)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1	0	1	1	1	1	1	1	1
R1(9,10)																			1	1	1	1	1	1							
R1(5,6)																			1	1	1	1	1	1							
R4(2,4)																			1	1	1	1	1	1							
R1(5,10)																			1	1	1	1	1	1							
R2(6,8)																			1	1	2	1	1	1							
R4(2,8)																			1	1	1	1	1	1							
R1(5,9)																			1	1	1	1	1	1							

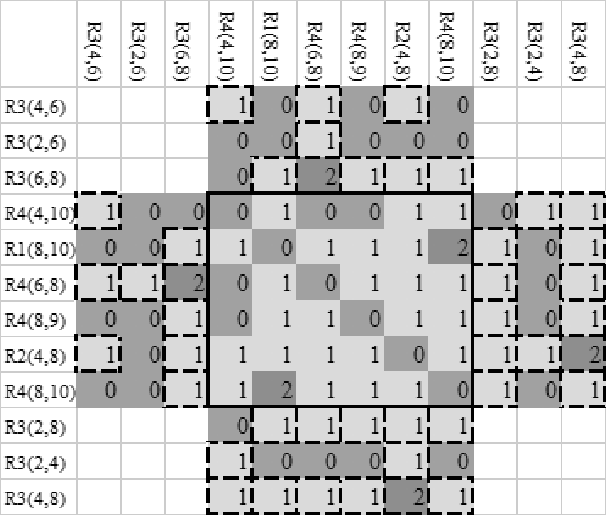
Figure 5 Example of a cluster within the role assignment adjacency matrix that indicates full dependency outside the cluster

By masking the edges, which are not element of a role, whose edges are within the cluster, we can highlight the accumulation of access rights outside the cluster (see Figure 4). Hereby, each 0-element indicates that a user does not participate in a role assignment that is proposed as aggregated by the cluster. These elements have been highlighted with dashed lines for better visibility.

Aggregation of the roles in the cluster could in this case lead to users gaining additional access rights. Therefore, it should be examined, why these users do not participate in the role assignments of the cluster, and whether the risk of providing the additional access rights within the cluster is feasible.

Another example is given in Figure 5. Here all assignments of the same roles, as in the cluster, are fully given. All affected users have been assigned to the same set of roles, which is indicated by the 1-elements outside of the cluster, but within the cluster rows and columns. This indicates that the access risks of the roles participating in the cluster must be examined. If the examination concludes that the combination of access risks is feasible, then aggregation of the roles enables easier maintenance of the role structure. Therefore these roles should be aggregated into a single role.

In the next step, the role assignments of roles participating in the cluster, which are outside the cluster, are being masked.



	R3(4,6)	R3(2,6)	R3(6,8)	R4(4,10)	R1(8,10)	R4(6,8)	R4(8,9)	R2(4,8)	R4(8,10)	R3(2,8)	R3(2,4)	R3(4,8)
R3(4,6)				1	0	1	0	1	0			
R3(2,6)				0	0	1	0	0	0			
R3(6,8)				0	1	2	1	1	1			
R4(4,10)	1	0	0	0	1	0	0	1	1	0	1	1
R1(8,10)	0	0	1	1	0	1	1	1	2	1	0	1
R4(6,8)	1	1	2	0	1	0	1	1	1	1	0	1
R4(8,9)	0	0	1	0	1	1	0	1	1	1	0	1
R2(4,8)	1	0	1	1	1	1	1	0	1	1	1	2
R4(8,10)	0	0	1	1	2	1	1	1	0	1	0	1
R3(2,8)				0	1	1	1	1	1			
R3(2,4)				1	0	0	0	1	0			
R3(4,8)				1	1	1	1	2	1			

Figure 6 Dependency of roles not participating in the cluster on the role cluster

This leads to a matrix indicating the role assignments participating in the cluster. Additionally we can see the role assignment of users, who would be affected by the cluster, whereas the role is not included in the cluster. This way, we can see which access risks accumulate at the user in the case of role aggregation within the cluster. Figure 6 provides an example for such an overview. When examining the dependencies with roles not affected by the cluster, the elements of the adjacency matrix, which are not set to zero are of most interest for us. These elements have been highlighted with a dashed bordering. In this case we can see, that from those users that have been assigned to the roles R4, R1, and R2 which are contained in the cluster, some have been assigned additional roles. For instance, some users that have obtained one of the roles R1, R4, or R2 have also obtained R3.

This means, that in order to successfully aggregate the roles, we must in this case not only consider the access risk accumulation occurring out of the set R1, R2, and R4, but also the access risk accumulation occurring out of the set R1, R2, R3, and R4. Furthermore, the access risk of the combinations (R1, R3), (R2, R3), and (R4,R3) should be examined if the role is not further aggregated.

6 Conclusion

Improving the role hierarchy and operative structure of RBAC is important to allow feasible administration of the access control data infrastructure [Fr09] [KSS03] [Mo09] [ZRE03]. However, as we have showed simply finding a minimal set of descriptive roles [VAG07] is merely enough, when it comes to dynamic and heterogeneous environments, such as infrastructure administration, smart systems. We have showed that entangling possible access risks within the role mining process is complex, due to the lack of formalization of access risks, and their combinations. While e.g. the maximum principle allows to use the highest risk, risk accumulations which may result in a more severe access risk than its' risk components, cannot be faced within classic role mining. We have therefore proposed to introduce a role mining approach within regular audits of the identity data infrastructure, and provided an approach which can be used to find role aggregations, while enabling examination of access risks. Hereby all access risks associated with the objects can be examined regarding their possible accumulation at subjects. The introduction of approaches surrounding design structure matrices seems feasible and provides a promising approach for identifying access risks. The ability to identify aggregation of subjects and objects within dynamic environments promises application during audit in dynamic environments. Further application of this approach for document exchanges in engineering will be considered within future research.

A central issue with such approaches however, as with most role mining approaches, remains the amount of noise incorporated within the UA. If the UA contains short-term role assignment, adjustment of the role structure within regular audits may cause to role data infrastructure to become instable between audits. In this case, the resulting structure may undergo several unnecessary adjustments, increasing workload for the administrators, while diverging from the regular work of the users. A technological solution could for instance incorporate the usage statistics. However, in most identity data infrastructure, such statistic is merely available for separate roles, but rather for the whole user (e.g. last account activity, or last log on). Therefore a solution of this issue may be found within case studies of concrete role data infrastructure within the given environments.

Still, the approach enables administrators to deal with issues of access risk consolidation, and to examine a role aggregation on a per user basis, without being required to examine the whole UA itself.

Acknowledgement

The research leading to these results was supported by the "German Ministry of research and Education (BMBF)" as part of the VERTRAG research project.

References

[Fr09] Frank, M.; Steich A. P.; Basin D.; Buhmann J. M.: A probabilistic approach to hybrid role mining. In: Proceedings of the 16th ACM conference on Computer and communications security. pp. 101–111 ACM (2009).

[GGM04] Geerts, F.; Goethals, B.; Mielkäinen T.: Tiling databases. In: Discovery science. pp. 278–289 Springer (2004).

[Hw88] Hwang, C.-R.: Simulated annealing: theory and applications. Acta Appl. Math. 12, 1, 108–111 (1988).

[Id95] Idicula, J.: Planning for concurrent engineering. Gintic Inst. Singap. (1995).

[Ja11] Jahchan, G.J.: Privileged User Management. In: Information Security Management Handbook. (2011).

[JGZ11] Jianyong, C.; Guiha, W; Zhen, J.: Secure interoperation of identity managements among different circles of trust. Comput. Stand. Interfaces. 33, 6, 533–540 (2011).

[KSS03] Kuhlmann, M.; Shohat, D.; Schimpf, G.: Role mining-revealing business roles for security administration using data mining technology. In: Proceedings of the eighth ACM symposium on Access control models and technologies. pp. 179–186 ACM (2003).

[Lu90] Luhmann, N.: Technology, environment and social risk: a systems perspective. Organ. Environ. 4, 3, 223–231 (1990).

[MR08] Meints, M., Royer, D.: Der Lebenszyklus von Identitäten. Datenschutz Datensicherheit DuD. 32, 3, 201 (2008).

[Mi08] Miettinen, P. et al.: The discrete basis problem. Knowl. Data Eng. IEEE Trans. On. 20, 10, 1348–1362 (2008).

[Mo09] Molloy, I. et al.: Evaluating Role Mining Algorithms. In: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies. pp. 95–104 ACM, New York, NY, USA (2009).

[OL10] O’Connor, R.C., Loomis, R.J.: 2010 Economic Analysis of Role-Based Access Control. NIST, Gaithersburg, MD, USA (2010).

[Sa96] Sandhu, R.S.; Coyne, E. J.; Feinstein, H. L.; Youman, C. E.: Role Based Access Control Models. IEEE Comput. 29, 2, 38–47 (1996).

[Th01] Thebeau, R.E.: Knowledge management of system interfaces and interactions from product development processes. Massachusetts Institute of Technology (2001).

[VAG07] Vaidya, J.; Atluri, V.; Guo, Q.: The role mining problem: finding a minimal descriptive set of roles. In: Proceedings of the 12th ACM symposium on Access control models and technologies. pp. 175–184 ACM (2007).

[Va09] Vavasis, S.A.: On the complexity of nonnegative matrix factorization. SIAM J. Optim. 20, 3, 1364–1377 (2009).

[YB03] Yassine, A., Braha, D.: Complex Concurrent Engineering and the Design Structure

Matrix Method. Concurr. Eng. Res. Appl. 11, 3, 165–176 (2003).

[ZRE03] Zhang, D.; Ramamohanarao, K.; Ebringer, T.: Role engineering using graph optimisation. In: Proceedings of the 12th ACM symposium on Access control models and technologies. ACM (2007).