# Envisaging Roles for Access Security

**C. Annapoorani & S. Sankar**

Department of Computer Science & Engineering, KCG College of Technology, Chennai
E-mail : poorani.kumar12@gmail.com, sankar@kcgcollege.com

***Abstract* – Access Control is the principle process of enterprise risk and Security Management. Role Based Access Control (RBAC) is the model for effective management of risk arises in safeguarding three important features of security Secrecy, Integrity and no Denial of Service (SoD). This paper addresses the new approach called "Role Visualization" - user permission pattern are managed as Visual Patterns. Basic concepts which essentially discover roles are by decomposing Binary Matrix Representation of User Permission Assignments, but it lacks the inclusion of several exceptions and SoD constraint. Implementing Extended Binary Matrix Factorization concept allow negative permissions to roles providing the way to include the exceptions also, which makes the roles meaningful. Traditionally administrators employ commands to interpret roles and permissions for a user, Visualization of Roles including the exceptions and constraints will pave the way for Visual Role Analytics and Knowledge Mining.**

***Keywords – role engineering, role mining, access security, permission set.***

## I. INTRODUCTION

Access Control Mechanism aim at mediating request to data and services. Among all models Role Based Access Control (RBAC) became the norm for managing permissions for Commercial Applications. Under RBAC, a role is a set of permission which users acquire the permissions to perform system functions only when they are assigned to specific roles.

### *A . Role Engineering*

The goal of Role Engineering [3] is to define set of roles that is complete, correct and efficient. Role Engineering requires defining roles and assigning permissions to them. Two basic approaches for Role Engineering, Top-down and bottom-up. Under top-down approach, roles are defined by carefully analyzing and decomposing business process into smaller units, defining particular job function and creating a role for this job function and associating needed permissions.

### *B . Role Mining*

Bottom-up approach of Role Engineering excels in the fact that much of Role Engineering process can be automated and that it utilizes the existing permission assignment to formulate roles. This bottom-up approach is referred to as Role Mining. Role Mining was defined as a method of generating roles from the Access Control Information (ACI) [4] of this collection of systems.

In this paper we propose a novel way of representing the users and roles i.e) the users and permissions in the pictorial way [5] . Mainly this represents the inclusion of exceptions and Separation of Duty constraints and present a meaningful role mining process

## II. BACKGROUND

The steps for representing the roles visually and the concept of EBMF in order to represent the roles to include the exceptions are explained in the following section.

Recently there has been an increasing interest in using automated role engineering technique. Despite much work dedicated to design of role mining algorithms, the proposed technique deal with three main practical issues Role Semantics, unwanted assignment of roles and correlations among roles [7].

### *A. Role Visualization*

The Visual Representation for user permission assignment allow for both an intutive role validation and a visual identification of relationship among roles. The natural representation for this information is Binary Matrix Representation where rows and columns correspond to users and permissions, and each cell is on when a certain user has a certain permission granted.

Traditional Boolean Matrix Decomposition [8] refers to decomposing of an input boolean matrix into a

product of two boolean matrix. The challenges in BMD are real world applications have varying constraints and exceptions and the real semantics cannot be revealed. The proposed technique implement Extended Boolean Matrix Factorization which aims to include both set union and set difference operations.

### *B . System Architecture*

The following architecture diagram depicts all the steps involved for the Visualization of roles will be done by algorithms ESSA and EBMF.

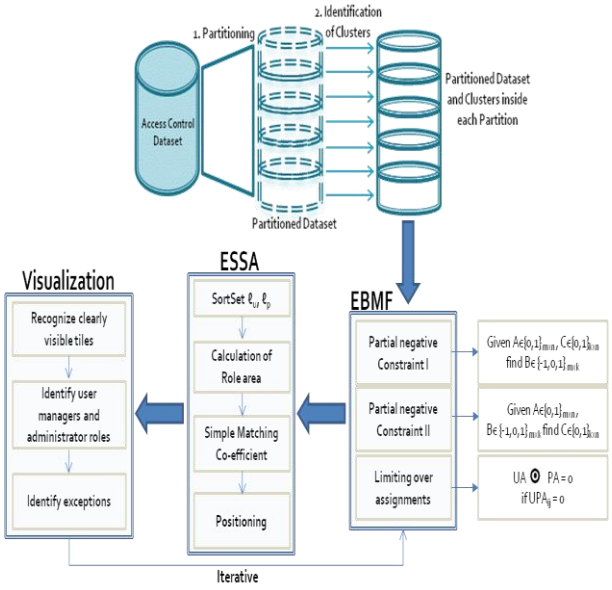



Fig. 1: Architecture Design For Role Visualization

#### *1) Partitioning*

In this step, the original dataset is decomposed into several disjoint subsets. However, in all prior approaches, each partition is analyzed Independently, without searching for the roles that extend across different partitions. On the contrary, in the proposed approach, the subsequent steps specifically focus on the identification of all roles. Formally, given U $\subseteq$ USERS, the partition of UP induced by U is defined as the set: {<u, p> $\in$ UP: u $\in$ U}

Similarly, the access control dataset can be partitioned using permission attributes as well. In this case, a subset P $\in$ PERMS induces the partition {<u, p> $\in$ UP: p $\in$ P}

#### *2) Identification of Clusters*

Analyzing each partition independently, trying to identify clusters of users, from which representative users will be picked. Since each partition is analyzed independently, the memory load is correspondingly reduced and it is also easily possible to parallelize this step.

#### *3) Extended Binary Matrix Factorization (EBMF)*

To address this ineffectiveness of the BMD model in capturing semantics, the proposed project introducing negative permissions or negative user-role assignment. As distinct from regular permissions, negative permissions mean that once permission is assigned to a user negatively, this user can never exercise that permission. Thus, negative permissions have higher priority than positive permissions indeed, if the user is already assigned the permission positively through another role or even through the hierarchy, this assignment is automatically revoked. If the user is assigned the permission positively in the future, it still does not become effective. Thus, negative permissions yield a great power and can effectively model both SoD constraints and exceptions.

#### *I) Partial Negative Constraint I*

Given original user-permission assignments A and regular roles C (or regular user-role assignments X), find semantic user-role assignments X (or semantic roles C), such that

$$||A - X \odot C||1 \text{ is minimized.}$$

The partial negative constraint I problem can arise on its own in a scenario as the following. Recall that semantic roles can be deployed to enforce some SoD polices by introducing negative permissions in roles. Suppose that SoD polices have been enforced and the reflective semantic roles are given. Now we need to assign those roles appropriately to users to match their existing user-permission assignments. This is a partial PNC I problem.

#### *II ) Partial Negative Constraint II*

Given original user-permission assignment A and semantic roles C (or semantic user-role assignments X), find regular user-role assignment X (or regular C), such that

$||A - X \odot C||1$ is minimized.

and $(X \odot C)_{ij} = 0$ if $A_{ij} = 0$.

The partial negative constraint II can also arise on its own. One possible scenario is that regular roles are given; the system administrator wants to assign fewer roles to each user by employing both positive role assignments and negative role assignments.

#### *4) Empirical Simple Sorting Algorithm (ESSA)*

Rows and columns are sorted independently. If some items are assigned to the same set of roles, they

are put together. Item set positions are decided one-by-one. The algorithm tries to avoid large gaps by putting item sets close to each other when they share large roles. Each item set is preferentially positioned at the beginning or at the end of already sorted item sets. Item set sorting is converted to item sorting.

*5) Visualization*

Visualization and sampling algorithms finalized to a visual role engineering activity. Then Inspection is made to identify potentially wrong or missing assignments. The proposed approach is an iterative method.

## III. SYSTEM IMPLEMENTATION

*1) Extended Binary Matrix Factorization*

As distinct from regular permissions, negative permissions mean that once permission is assigned to a user negatively, this user can never exercise that permission. Thus, negative permissions have higher priority than positive permissions indeed, if the user is already assigned the permission positively through another role or even through the hierarchy, this assignment is automatically revoked.

*2) Empirical Simple Sorting Algorithm*

a) The ESSA includes the following steps

b) Rows and columns are sorted independently.

c) If some items are assigned to the same set of roles, they are put together.

d) Item set positions are decided one-by-one. The algorithm tries to avoid large gaps by putting item sets close to each other when they share large roles.

e) Each item set is preferentially positioned at the beginning or at the end of already

f) Item set sorting is converted to item sorting.

Procedure essa(USER/PERMISSION,UPA,ROLES)

USER/PERMISSION ← { I C USER/PERMISSION |Vi, I' Є 1, roles(i) = roles(I')}

For all I Є USER/PERMISSION sorted by descending areas of roles(I) do

If |σ|<2 then

σ.append(I)

else

if SMC(I, σ.first) > SMC(I, σ.last) then

p ←1 s← SMC(I, σ.first)

else

p ← |σ| + 1 s← SMC(I, σ.last)

end if

for i = 2……|σ| do

sprec ← SMC(I, σ[i-1], ssucc ← SMC(I, σ[i])

scurr ← SMC( σ[i-1], σ[i])

if max{ sprec, ssucc } > s min { sprec, ssucc } then

p ←i , s ← max{ sprec, ssucc }

end if

end for

σ.insert(p,I)

end if

end for

end procedure

*3) Simple Matching Coefficient Calculation*

Simple Matching Coefficient is useful when both positive and negative values carried equal information

$S_{ij} = (p + s) / t$ where $t = p + q + r + s$

p – No. of variables of positive for both objects

q – No. of variables positive for i and negative for j

r – No. of variables negative for i and positive for j

s – No. of variables of negative for both objects

*4) Visualization – Knowledge Mining*

Graphical Visualization to be performed based on the matrix output

Knowledge can be mined are

a) Each Tile area is a Separate Roles

b) Number of Roles, Permissions, Number of users to corresponding roles can be derived

c) Elicit meaningful roles

d) Hierarchical constraint can be identified by the overlapping of matrix

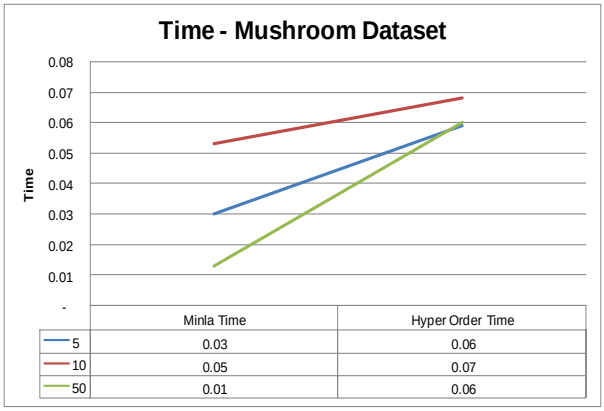
e) Cardinality constraint i.e. maximum roles to a user can be crosschecked

f) Implementing EBMF is used to visualize negative authorization add semantic role mining

g) Identification of noise such as Roles with only one user, User deviating cardinality constraint can be identified

## IV. RESULT & ANALYSIS

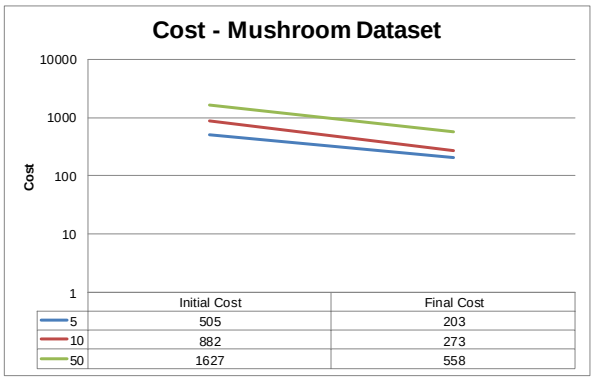
The Experiment was performed with standard datasets in C++ platform in Linux based system. Mushroom dataset was taken as a sample and the time and cost comparison was made between our algorithm and MinLa algorithm. The time taken for the execution of algorithm compared against the MinLa Algorithm. The result shows that the execution time is greater compared to MinLa program.




| | Minla Time | Hyper Order Time |
|---|---|---|
| 5 | 0.03 | 0.06 |
| 10 | 0.05 | 0.07 |
| 50 | 0.01 | 0.06 |

Fig. 2 : Time Comparison between ESSA and MinLA

When comparing the cost of execution our algorithm was very much efficient when compared to the MinLa and the experiment was repeated for several standard datasets. Our Algorithm is Cost Efficient but Time consumption greater than MinLA.




| | Initial Cost | Final Cost |
|---|---|---|
| 5 | 505 | 203 |
| 10 | 882 | 273 |
| 50 | 1627 | 558 |

Fig. 3 : Cost Comparison between ESSA and MinLA

The Unordered image of the Users and Permissions was given below at the end of the implementation of Algorithm

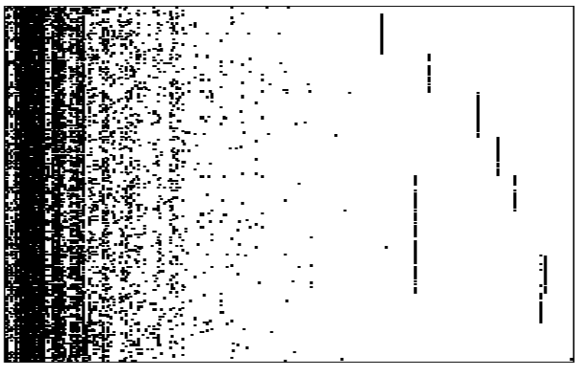


Fig 4 : Unordered Image

The Reordered image of the Users Permissions set data and the roles are highlighted in the following image

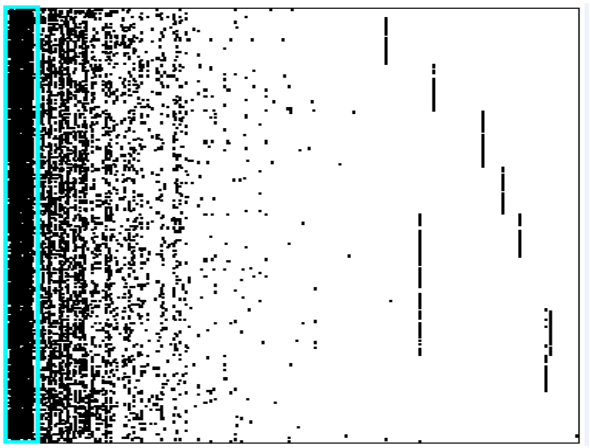


Fig 5 : Reordered Image

## V. CONCLUSION

Visualizing user-permission assignments is an intuitive graphical form that makes it possible to simplify the role engineering process. The proposed representation of data allows role designers to gain insight, draw conclusions, and ultimately design meaningful roles from both IT and business perspectives. The paper includes both positive and negative constraints and the corresponding unordered and reordered matrix and the visualization images were generated. The limitation in terms of visualization is as there is increase in the data. The Visualization gave the better understanding of roles than by tables.

## VI. REFERENCES

[1] A. Colantonio, R. Di Pietro, A. Ocello, and N.V. Verde, "A New Role Mining Framework to Elicit Business Roles and to Mitigate Enterprise Risk," Decision Support Systems, vol. 50, no. 4, pp. 715-731, 2010.

[2] A. Colantonio, R. Di Pietro, A. Ocello, and N.V. Verde, "Mining Business-Relevant RBAC States through Decomposition," Proc. Security and Privacy-Silver Linings in the Cloud, pp. 19-30, 2010.

[3] E.J. Coyne, "Role-Engineering," Proc. ACM Workshop Role-Based Access Control (RBAC '95), pp. 15-16, 1995.

[4] S. De Capitani Di Vimercati, S. Foresti, P. Samarati, and S. Jajodia, "Access Control Policies and Languages," Int'l J. Computational Science and Eng., vol. 3, no. 2, pp. 94-102, 2007.

[5] J.-D. Fekete, J.J. Wijk, J.T. Stasko, and C. North, "The Value of Information Visualization," Information Visualization: Human-Centered Issues and Perspectives, pp. 1-18, 2008.

[6] D. Ferraiolo, R.S. Sandhu, S. Gavrila, R. Kuhn, and R. Chandramouli, "Proposed NIST Standard for Role-Based Access Control," ACM Trans. Information and System Security, vol. 4, pp. 224-274, 2001.

[7] Q. Guo, J. Vaidya, and V. Atluri, "The Role Hierarchy Mining Problem: Discovery of Optimal Role Hierarchies," Proc. Ann. Computer Security Applications Conf. (ACSAC), pp. 237-246, 2008.

[8] R. Jin, Y. Xiang, D. Fuhry, and F.F. Dragan, "Overlapping Matrix Pattern Visualization: A Hypergraph Approach," Proc. IEEE Int'l Conf. Data Mining (ICDM '08), pp. 313-322, 2008.

[9] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role Mining—Revealing Business Roles for Security Administration Using Data Mining Technology," Proc. Eighth ACM Symp. Access Control Models and Technologies (SACMAT '03), pp. 179-186, 2003.

[10] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo, "Evaluating Role Mining Algorithms," Proc. 14th ACM Symp. Access Control Models and Technologies (SACMAT '09), pp. 95-104, 2009.

◈◈◈