

Master Thesis
Computer Science
Thesis no: MCS-2006:16
January 2007



Separation of Duty in Role Based Access Control System: A Case Study

Authors:
Francis M. Kugblenu
Memon Asim

Department of
Interaction and System Design
School of Engineering
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Interaction and System Design, School of Engineering at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Author(s):

Francis M. Kugblenu

E-mail: kugblenu@gmail.com

Memon Asim

E-mail: assimmemon@hotmail.com

University advisor(s):

Peng Zhang

Department of
Interaction and System Design
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457 102 45

ABSTRACT

In today's business world, many organizations use Information Systems to manage their sensitive and business critical information. The need to protect such a key component of the organization cannot be over emphasized. Access control has been found to be one of the effective ways of insuring that only authorized users have access to the information resources to perform their job function. Role Based Access Control has been found to be the access control mechanism that fits naturally with the organizational structure of businesses. Separation of duties is a security principle that has been used extensively to prevent conflict of interest, fraud and error control in organizations. In this thesis, we identify the various forms of separation of duties in role based access control systems. We also do a case study of the role based access control system in the banking application of a financial institution.

Keywords: Role Based Access Control System, Separation of duty, Case Study.

CONTENTS

ABSTRACT	2
List of Figures	4
ACKNOWLEDGMENT	5
1. Introduction	6
Background	6
Aims, Research Question, Approach	7
2. Separation of Duty.....	9
2.1. Introduction	9
2.2. Background	9
2.3. Types of Separation of Duty	11
2.3.1. Static Separation of duty	11
2.3.2. Dynamic separation of Duty.....	11
2.3.3. Simple Dynamic Separation of Duty.....	12
2.3.4. Object-based Separation of Duty	12
2.3.5. Operational Separation of Duty.....	12
2.3.6. History-based Separation of Duty	13
2.4. Kinds of Separation Duty	13
2.5. Separation of duty in Real system.....	13
2.5.1 Separation of Duty in role hierarchies.....	14
2.6 Mutual Exclusion	15
3. Role Based Access Control	16
3.1. Introduction	16
3.2. What is Role?	17
3.2.1. Type of Roles	18
3.3. Access Control	18
3.3.1. Types of Access Control	19
3.2.2. Discretionary access control.....	19
3.3.3. Mandatory Access Control (MAC)	20
3.3.4. Detailed Description of Role base access control	21
3.3.5. Why RBAC is Useful?	22
3.3.7 RBAC Model.....	23
3.3.7. Roles Hierarchies in Role Based Access Control	27
3.3.8. User and permission assignment	29
3.3.9. Comparing RBAC to MAC and DAC.....	29
4. Research Review of Separation of Duty in Role Based Access Control	31
5. Case Study.....	35
5.1. Introduction	35
5.2. The Institution	35
5.3. The Banking Application	35
5.4. Access Control System:.....	36
5.4.1. Roles.....	36
5.4.2. Templates	36
5.2.4. Classes	38
5.3.5. User Id	38
5.2.6. Separation of Duties:	39
5.2.7. Assessment	39
6. Discussion	41
7. Conclusion.....	42

References 43

List of Figures & Tables

Figure 1: Role Hierarchy [5] 14
Figure 2: RBAC Model [35] 24
Figure 3: Relationship between Class, user and templates [57]..... 38
Table 1: Branch Process Separation of Duties Matrix [56].....40

ACKNOWLEDGMENT

We would like to say a big thank you to our supervisor Peng Zhang for his guidance and support during this work. To Rune Gustavsson and Martin Fredricksson we say a very big thank you for their comments and suggestions that have shaped this work. To all our lecturers and friends thanks for all your useful comments.

To our families, we say big thanks for their love and support throughout the duration of our course. And to all those who have in their various ways contributed to the successful completion of our course we say thanks.

1. Introduction

Background

Information is a valuable asset for most businesses. In today's highly competitive business environment, information assets need to be protected from unauthenticated access. The disclosure of sensitive information about a company's customers, strategic plans or products a competitor could not only lead to a huge financial loss, loss of competitive edge, loss of reputation and legal liability but also gives the competitor the opportunity to leapfrog the company [3]. The competitor does not need to incur the financial burden as well as the time involved in research and development. They also have the opportunity to evolve counter strategies to a company's plans before they even implement the strategy. Such disclosure of critical information is almost impossible to recover from [3].

The arrangement of similar development dependences for these critical systems faces development scope of threats like Denial of Service (DOS), Hacking, Viruses, Worms, Espionage, Computer Assisted Fraud, sabotage etc [2]. In order to protect their systems from such attacks, organizations have evolved various procedures and systems aimed at protecting information assets from both internal and external threats [1]. These procedures and systems are called information security. The main aim of information security is to ensure Confidentiality, Integrity and Availability of Information assets.

- Confidentiality: Prevents access to or the disclosure of information to unauthorized entities. Many methods can be used to achieve this including Cryptography and Access control
- Integrity: Ensures the information that is available to authorized users is both accurate and properly processed.
- Availability: This is the ability of authorized users to have access to information resources when they need or request for it. It is concerned with the prevention of denial of service attacks.

In summary one can say that the objective of information security is to deny access to information resources to unauthorized users whilst making it available to the authorized users. This should be done in such a way that it does not adversely affect the business of the organization [4]. The availability of information resources is one that organisations have invested heavily in. One of the technologies that organisations have used to achieve this is access control. Access control provides a means to control which entities in an information system have access to which resources and what the nature of such an access [5, 6]. By making information resources available to only authorised users, the mechanism ensures that only information is always available to those permitted to access it. There are many access control mechanisms that can be deployed by organizations to meet their information security needs such as Mandatory access control (MAC), Discretionary Access Control (DAC), Role Based Access Control (RBAC) etc. Depending on the nature of the organisation a particular access control mechanism is usually more appropriate. In the business world, organisations are usually divided into various departments and units. Each department/units has a specific function within the organisation. Staffs of the department/units have responsibilities assigned

to them based on the department/unit they belong to. For example a clerk in the accounts department has different responsibilities from a clerk in the administration department. As a result of the differences in the responsibilities assigned to staff, their access to information resources varies. Access control mechanisms ensures that these staff have access to only those information resources that is required to facilitate their work. With the increasing sizes of organisations and the complexities of job functions within organisations managing this large number of access rights becomes a major problem. There are many examples where the failure or user access rights management has resulted in huge losses to organisations. Role Based Access Control (RBAC) is one of the access control mechanisms that have been found to be very useful in the managing of access rights within organisations. It uses the concept of roles to manage user rights. Since in most organisations responsibilities are already assigned based on roles, it makes a perfect fit to assign access rights along that line. Since the roles in organizations are less likely to change than the users performing the role, this makes the administration of users easier whilst reducing the possibility of errors in the administration of access rights [5]. In addition to restricting who has access to what information and what the operations a user can perform, organizations also make use of conflict of interest mechanisms to prevent attack, fraud and also detect errors. One of such mechanisms is separation of duties. Separation of Duties (SOD) is a fundamental security principle used to prevent fraud and detect errors [5]. Role Based Access Control (RBAC) provides organisations with a platform to implement this security principle. Whilst this principle has been implemented in manual systems, it has always been difficult to make an implementation in information systems. According to [56] preventive controls are considered to be more effective than detection after it had occurred. The prevention of fraud is more beneficial to organisations than its detection after it had occurred. The challenge for most organisations is how to make an assessment of their separation of duties policies.

Aims, Research Question, Approach

From our background we come with the following research question:

- Can we make an assessment of the final result of implementing separation of duties in role based access control systems?

We also have some sub questions to be answered:

- How can organisations implement SOD in RBAC Systems without any adverse effects on business?

These questions are important enough to warrant the study because Role-Based Access Control is used in many organizations across all sectors. Because its ability to implement access control in a way that fits in naturally with the roles within organizations [5]. It is important such an access control policy fully implements the Separation of Duty principles especially in organizations like Banks and the Military where a single user must not complete certain operations.

In this research we commenced by conducting a detailed review of previous and current research into Access control, Role-Based Access Control and Separation of Duty. From our review we generated a list of questions that formed the basis of the case study we conducted. We used the approach suggested by [56] to make an assessment of the Separation of Duties.

Chapter 2 introduces Separation of Duties (SOD); we discuss the various types of this security principle and the different kinds of separation. Chapter 3 starts with an introduction to roles, and then we discuss access control and the various access control mechanisms. We then go into a detailed discussion of role based access control and the Role Based Access Control Model. In chapter 4 we discuss previous works done. On the basis of the literature review and previous work done, we generated a list of questions and conducted a case study of the access control in banking application software of a financial institution in chapter 4. We discuss our findings in chapter 5 and conclusion in chapter 7.

2. Separation of Duty

2.1. Introduction

Separation of duty is the security principle used in multi-person control policies that shows two or more different persons responsible to complete the task and set of related task. The purpose of this principle is to oppose fraud by dispersal the responsibility of an authority for an action or task over multiple persons.

The problem of separation of duty is important for several reasons because it reserves for some tasks that are essential for the security. Second is that the single person making them unique authorized issue. Third its specialized mechanism and in last it is generally used in real world for example the case history ascendancy big alignment transaction, military application to curb the nuclear weapon, and credit the medical for non - advent surgeon.

There are two types of Separation of duty, Static or strong exclusion and Dynamic or weak exclusion. Static Separation of duty defines the partitioned ascendancy groups of role and altered roles assigned to the altered user influence different action. This means two roles have no shared principle. For example if order approval and creator are exclusive role than no one who may assume to be the approver role would be allowed in the creator role and no one who may assume to be creator role would be allowed in approval role. Dynamic separations of duty shows that task have different actions that are performed by different individual even if both are governed by same role. This means member of any two exclusive roles they should not be activates both in same time. In this chapter we discuss about the separation of duty with its types.

2.2. Background

Separation of duty has long history in the computer security research. It is the foundational polices of the computer security. In 1975 Seltzer and Schroeder [12] defined the privileges of separation of duty, which is one of the eight-designed principles of the information of computer security. The R. Needham is the making of following observation in 1973 defines protection mechanism to secure the system and also shows that two keys to unlock the system, is better than require the key with single one. No single accident, dishonesty or break the trust of insufficient to assist the system[11].

There are numbers of computer systems supporting the basic of this principle. But frequently this support is unreliable with the way of the principle that is used in the computing environment. It is multi support people's policies that mean two or more persons have performed the task or set of related task [11]. RBAC mechanisms used by a system administrator insist the policy of separation of duties. It is valuable to prevent fraud since fraud can happen if situation or occasion exists in teamwork between various jobs related capabilities [9]. Separation of duty is a fundamental principle of computer security. The principle states that task to be performed by two different users, which acts in any corporation. The idea of separation of duty is long existed before the information age and it extensively used like in the bank and in the military. The policy of separation of duty is like as K different users can perform the task. Some time needs k-1 together with the all permission to perform that task [8].

The main idea of the separation of duty is that user cannot start the action until some action granted to that one. This means that separation of duty is number of responsibility to various steps with different individuals. Some tasks are to be decomposed which has to be assigned to the different users. Every user are assigned a subtask which is been performing and it is to be restricted that no one cannot perform more than one task[7].

When talking about the separation of duty constraints. We need more actions and actions can perform by more than one person. This reduces the fraud with at least two reasons. First is, More than two persons can fail to act the role in the organization that is low probability then the failure of a single person can act the job. Second, collusion requires that one party propose impostor to more. The second party may report the first party to authorities; alternatively, if the second party does not report the antecedent party he runs the risk that the party was testing him at the order of authorities (and therefore the anterior party's suggestion was not authentic). Consequently, real is not guarded for either party to cooperate or even discuss an offer of collusion[16].

Let takes the example that we have taken from [13] about the buying and purchasing the goods. Following steps are given below

Step 1:

Order and record the order detail of goods.

Step 2:

Record the influx and check the detail of influx with matching the order detail.

Step 3:

Check that the goods have been received and quality of the detail match the detail of influx.

Step 4:

Approve the payment to the dealer against the influx. Which would want to show that payment is not released on that order that was never places and given goods match those in the order and those in the invoice.

The policy is too restrictive where different user has performed some step. It may be allowable, for example that user, which orders the place, is also the record the invoice of the arrival.

1. It should be three users that perform these four steps.
2. Two users can perform step 1 and step 4 tasks.

We can say that a single user cannot order the good and allow for payment.

If a single user creates and approves order by a phone, or gives the money by pocket, but the number of users create and approve the order than it creates fraud require with other people which is bad or illegal and increase the risk of revelation and capture the things significantly.

The problem of separation of duty is important for some reasons. The first reason is that it reserves for the same task which is good for the computer security. Second, according to the definition it is so important that single person authorize the issue, which is uniquely identified. The third is that it should be explained in particular mechanism. And last it is generally used[15].

2.3. Types of Separation of Duty

The separation of duty has two types, Static or strong exclusive and Dynamic or weak exclusive.

2.3.1. Static Separation of duty

The static separation of duty is also called strong exclusive which states that “A principal may not be member of any two exclusive roles” [10]. It means that the user is authorized of one role may not be authorized of another role or two roles have no any shared principle. If the roles are assigned in the system and the users has been assigned the roles with the given task. So static policies will carefully assign the tasks to the users without keeping the history of every task. And if the static separation of duty is satisfies the policy then the separation of duty is also satisfied. Lets takes an example that when the creator and approval are restricted then no one who may assume as a create role is assume to be the approval role and vice versa. It can be implemented only in the member of the roles. The main advantages are simplicity because it is not a practical or pragmatic version of the variation [11]. And it is not the actual function of human organization [14]. Users frequently have lawful reasons for inadequate or needing to access in two strongly exclusive roles, and prudent construction of a security policy can ensure that these violations are secure [11].

❖ *Statically Mutually Exclusive Role (SMER)*

Static separation of duty imposed using constraint, which restricts the role membership of the user. Let’s take an example the one constraint doing two roles that is mutually exclusive, means that no user can be a member of both roles. In other explanation constraint that no more roles for the one user and no more roles is that part of on set roles called Statically Mutually Exclusive Role (SMER).

The static separation of policy is not to be considered the same as SMER constraints. Every SSOD policy describes the number of users, which allow the users to work together with all permissions to the given task. Such type of policy may be independent whether users should manage the permission or not.

Each constraint limits the role memberships a single user is allowed to keep. Whether a set of SMER constraints is adequate to enforce an accustomed SSoD policy depends upon how permissions are assigned to roles. For example, if all permissions that are needed to essential a sensitive task are assigned to a different lower - conspicuously role, apart cannot advantage SMER constraints to ensure that no single user possesses all the permissions, as no SMER constraint responsibility dissuade a user from being assigned to that single role and thereby gaining all permissions needed for the job[8].

2.3.2. Dynamic separation of Duty

The Dynamic separation of duty is also called weak exclusive which states that “A principal may be a member of any two exclusive roles but must not activate them both at the same time” [10]. The above definition shows that user is authorized of both roles but both roles cannot be active at the same time. It means system will keep the record of each task. In this record all the information that is used is to be performed. Before doing any task, the system will check the separation of policy should not to be broken[14]. Weak exclusion, or Dynamic Separation of Duty, provides the larger set of possible policies, which control the commencements, and use of roles. Dynamic separation of role allow the users to Perform the roles that would be strongly exclusive in static systems, as long as constraints are satisfied that get rid of or decrease the risk of fraud. Because it convenient and replicates the function of human organizations. It has several variations. To define the weak exclusive we use the term called restricted role that refer the role has the limitation of other membership activation or use[11].

SSOD is good to analyze and specify the system while DSOD does not. But DSOD is more flexible because it enables the protection to betoken tuned. So that for the identical character of persons, either added separation responsibility represent obtained or authentic is easier to identify someone who can perform an action (or some combination of these two).

In other authorization mechanisms, authentic suffices to allot one sufficiently trusted user to perform each action. But it is not good in the DSOD. If the user is highly trusted, he cannot perform the given action because he has to do other contradictory action earlier.

Influence this sense; DSoD actions consume users since, because a consequence of performing an activity a user is unable to perform future actions to which he would antithetic appear as permitted. Therefore, an organization’s test sequence for DSoD depends on both organizational size and structure.

If N is the people of organization, then N is the number of people that is involved in the task. Smaller organizations will commitment to determine, for case history; between lower SoD and having executives perform actions which larger organization would perform by clerks. So, organizations commitment master and analyse their own provability sequences bury the desired degrees of SoD[16].

2.3.3. Simple Dynamic Separation of Duty

It also called the Dynamic separation of duty where roles are restricted for the common member but the user may not assume to be performed same role at the same time [11, 20].

2.3.4. Object-based Separation of Duty

Roles have restricted for the common member, and those members may assume both roles in the same time. But no user may close upon a target that user has previously acted upon. This is called the object-based separation of duty. [11, 14]

2.3.5. Operational Separation of Duty

In this duty that all the roles action can not be contain in the complete business task and its may restricted for the common member as long as the union of all groups. This prevents

apportionment one person from performing all of the actions ascendancy the business task called as operation separation of duty [11, 20].

2.3.6. History-based Separation of Duty

The lack of respond to both object based separation of duty and operational separation of duty in the simple alternatives. They have still act not to allow some of the policies of separation of duties. Object based Separation of Duty does not concede a user to perform a second activity on an object when this makes sense and is allowed by human policy, and in the operational Separation of duty policy that a user cannot perform all the action in the task to different object when this makes sense and allowed by human policy. To allow complete suppleness, describing the separation of duty policy, these two variations should act as combined. Two and more limited roles may have common members and the union of the action granted by those roles may distance the action in the business task, but no role member is allowed to perform all the actions ascendancy the business task on the equivalent target or collection of target called as history based separation of duty[11].

2.4. Kinds of Separation Duty

There are two kinds of separation duty. One is the Dual control another is function separation.

❖ Dual Control

In the Dual control the two different members can work together to access the transaction. For example two different people turn the keys at the same time as compare to work on single person to operate. The example is taken from [17] when a bank issues a letter of guarantee, which will typically undertake to bring the losses should a loan fabricated by added bank get-up-and-go sour. If a single manager could issue like an antecedent, therefore an accomplice could steal the guaranteed loan report at the other bank, and the alarm might not represent raised for months.

❖ Functional Separation

Functional separation of duties, two or also altered staff members act on a transaction at altered points influence its path. The classic exemplar is corporate purchasing. A manager makes a purchase accord and tells the purchasing department; a clerk adept writes a purchase order; the store clerk records the arrival of goods; an invoice arrives at accounts; the accounts clerk correlates real cache the purchase adjustment and the stores receiving, and cuts a check; the accounts manager sign the check[17].

2.5. Separation of duty in Real system

Market provides many complicated feature of RBAC system, which includes complex role hierarchies. Some have strong support to SOD. Some have used to implement of Separation of duty constraint with limited degree. Developers and system integrators check the feature in the operating system and DBMS, which supports the RBAC. There are some tradeoff and SOD rules, which use in the real system design. In general we use some automated engineering tools that will be needed to check and create the SOD rule in the RBAC system [19].

2.5.1 Separation of Duty in role hierarchies

More RBAC system supports Role hierarchies and SOD constraints have some implication for role hierarchies that impact the administration of SOD. When the role can join with another role, the system must confirm that the joining structure cannot result to break the separation of duty. Some properties define in Kuhn [18] that supports the interaction between SOD rule and role hierarchies. There are some properties which defines as

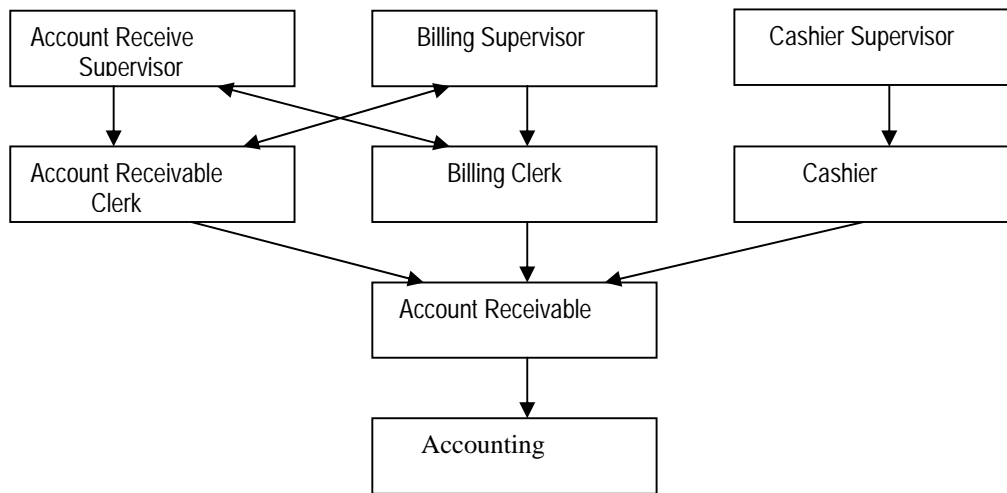


Figure 1: Role Hierarchy [5]

❖ *Property1*

It shows that when Role1 and Role2 are mutually exclusive except it cannot be join to one another either directly or indirectly. The separation of duty cannot upload if the user uses the other role. This rule affected the role administration. So that role management will check the administration violation that created by joining [19].

❖ *Property 2*

It shows that if the two roles are mutually exclusive that third one cannot be join both of them. This rule is better than first one because it may manage the role management that allows the administration to maintain the role hierarchies without breaking the Separation of duty constraint.

❖ *Property3*

When Static SOD is control than Dynamic SOD is access for two roles but can never be act in same time. The main things of this rule organization have some resources which fix the limit of roles that user may access. The main example of this property is that if one user has the role of receiving account. It will never use to access the billing account. The static separation of duty can be created for this purpose. However the SOD is created when the roles are accessible. System will not check these constraints in user session. It reduced the risk of attack by virus and improves the performance.

❖ *Property 4*

If there are two roles, which are mutually exclusive, then there cannot be any root or super privileges role active in the system. It means that role cannot be joining two other roles that are mutually exclusive but root role can join any other role [19]. It can be occur when the role in Dynamic SOD, one user can access all the roles but the roles cannot be active simultaneously. Because root role are join to other roles. But root cannot be activated with all joining roles.

Constraint with the SOD is joining in the opposite direction of the role membership. If the role hierarchy as a tree in the representation and the root is the most general one, then role membership is joining down. For example in the fig 1 the accounting department hierarchy the membership chain is account Receivable supervisor to account receivable clerk to account receivable to accounting. An employee selected for the account receivable clerk is join membership that down to the tree that include account receiving and accounting.

Constraints are joining in the opposite direction. For example in the fig 1.the account receivable account clerk role have static SOD relationship and account receivable supervisor also have static relationship. Other way of thinking is that the account receivable supervisor can be work as the account receivable clerk because it's an instance of accounts. Also account receivable clerk must also use account receivable supervisor.

2.6 Mutual Exclusion

Those systems that have existed and will come in near future supports the mutual exclusion of roles that effect of SOD policies. However the mutual exclusive rule is supplied more than one way and design will distress ease of use considerably. .

3. Role Based Access Control

3.1. Introduction

With the great use of Internet, low cost of the technology and need to share and access the data is now being driven in the new technology market. In the competitive edge with security and productivity the system user and developer are looking the properly administration to increase costly infrastructure. Moreover the number of users and delay in the account creation losses the data access. So that government and other organization are working on the security and also the privacy of information. This security is not only in the internal level but also in the external level. The Role based access control is the security in administrative level with large and complex environment without give up the need the security and access the data.

Access control technology has evolved from research also adding to efforts supported by the Department of Defence (DoD). This research has resulted influence two fundamental types of access called Discretionary Access Control (DAC) and Mandatory Access Control (MAC), while initial explore and applications addressed preventing the unauthorized access to classified break.

DAC allows to right access and permission of the users under the access control privileges, however organizations have not been given access to the end users to allow the own information. . For these organizations, the concern or agency is the actual owner of system objects seeing bright-eyed because the programs that advances them. Access priorities are controlled by the alignment and are recurrently based on employee functions quite than data hold.

The Department of defense is the Trusted Computer Security Evaluation Criteria (TCSEC), defines the Mandatory Access Control (MAC), which shows to restrict the access object information and authorizations, based on the security label. These types of the policies are not good for the requirement of government and the industry organizations. For these polices the security objective support the high level of organization which resulting the Law of ethics and general accepted practices. These policies have ability to control the access information shows that how information are labeled based on its sensitivity [27].

The role based access control system is based on the roles where single user is the part of an organization. It means to control access of system resources and protects them with unauthenticated users. It is based of individual user role and their responsibilities. Roles are some kinds of task where user can perform some operation. User takes as assign role like as a doctor, a bank accountant, bank manager etc. The case of defining roles should appear as based on an in-depth analysis of how an alignment operates and should build in input from a wide spectrum of users in an organization.

RBAC works in the organization, where the user can check the mails having given role in the system. And the system administrator checks how user can interact with the system. Each role has own view and process. It's the alternative approach of Mac and DAC. In this chapter we discuss about the Role, Access Control and its types.

3.2. What is Role?

Role is certain collection of rights; duties and position describe the status within organization. It specifies the management action, behavior of dynamic aspect of the place, which is essentially as static concept. The role is identified the authority, responsibility and interaction associated with the position. We model rights considering authorization policies which specifies what activates a subject is permitted or forbidden to perform on a set of target thing. Duties are model being obligation policies which specifies what actualize a subject need or committal not perform on agree of target object. A role is target of centre object further compulsion policies, which keep a particular director position being a subject [38]. The advantage is if using a position being the subject of polices is that human's responsibility act as assigned to or withdrawn from their positions lost having to re-specify policies [32].

“It is the set of authorization and commitment policies which have the particular position as subject. The benefit of position as a subject can be assign or can be withdraw without having any respectively policies” [32].

The main purpose of the above definition is that every organization has some user, and Organization has some tasks, which has to assign to the particular user's means he or she can do the task of given related role without disturbing the other user. The benefit for this role that company can change any time the role to one person to another without any commitment or any other policies.

Role gives you some task to assign particular user or more users. Role can represent the authority and responsibility like project manager has some responsibility and authority that is different from their abilities. The person has responsibility to mange the department but have some other responsibility only those companies that have managed.

Roles can be assigned to turn in circle with different users. Role based model implementation has good accommodate to show all of role concept. Many commercially auspicious access control systems for mainframes apparatus use roles for security administration. For example, in the security office role can changes the permission but not to access the resources. In the same way the access user role can use the resources that are given but not to change the access permission. System administrators use the roles system administrator in network operating system called Novell networkNT[35].

The role provides the clear group of policies of the position, which shows to examine the consent and errands assigned to those positions. In a usual access control based on control list, which formative the permission assigned to a subject may require in depth search of all target object in the system. Assigned the role to people don't work in separation interrelate and cooperate with other roles. We have comprehensive role structures that allow the specification of connection between roles reflecting. The relation between dissimilar roles in a computer system provides the people to simplify the specification for the system and also clearly specification of the right and duties, which connected to the organization structure [32].

The combination of roles and permission by the role is change over time. The permission connected with the role is the other hand, which cares to change, is less often the people who is doing the job function that role are present. Therefore the security administrator on role, as compare to on permission is simple. User can reassign the different role, as they needed. So

that some time company needs new application and system roles that have permission access granted and existing permission cancel [35].

3.2.1. Type of Roles

There are two types of role. One is the basic role and other is function role.

❖ *Basic Role*

It is an authenticated independent that determines the function role and operation on protected information object. The resources information determines which application is workflow are permitted to a user in a first place. Basic roles support service-based architecture where the centrally manager protects the access resources. It advantages for the place and definition of function roles in the context, which includes basic roles. It is a user authorization that occurs before the roles can be activated [33].

❖ *Function roles*

“Function roles cannot occur until the session is established and authorization to establish the session may occur outside of the application authorization function”[33]. The definition for the above example is that in every organization has some task for the particular project and its role cannot be performed until the session is start means that user has already assigned the roles before starting any task or any project.

3.3. Access Control

The main objectives of access control are to regulate users to access the resources in the system. In other words access control ensures that only authorized users have access to resources. It determines which resources a user has access to, and if the user has access such a resource what the nature of the access is. In addition to that access control is also concerned with when a user has access to a resource any attempt by a user to access are source is subject to the access control policy [5,26]. For example in a bank a policy could take the form of preventing tellers from initiating transactions on weekends or allowing access only upon the permission from a manager. In an Information system, access control is implemented at different levels namely. At the Application level, access control is what's visible to the user. This level implements complex and rich access control mechanism. At the Middleware level, DBMS access control mechanisms regulate the access to tables and views by users. At Operating system level, access control is used to control to resources such as files and communication ports. The Hardware level is concerned with regulating the access to memory addresses by processes [26]. Access control preserves the confidentiality of information resources by ensuring that information is only disclosed to users authorized to access it. Integrity of information resources is achieved by restricting modification to resources only to those with the permission to do so. Availability is preserved by preventing unauthorized users from gaining access to the system and thus performs a denial of service attack [5].

Access control when properly implemented can enhance the operation of an organization through the exchange and sharing of information resources. However, when it not properly implemented can lead frustrate users, lead to unauthorized disclosure and impose a high financial burden on organizations [5].

3.3.1. Types of Access Control

There are several types of access control; for example Mandatory Access Control (MAC), and Discretionary Access Control (DAC), Role Based Access Control (RBAC).

3.2.2. Discretionary access control

Discretionary access control determines by the owner of the file and other resources. In this policy the owner will decide which user has to allow accessing the file and what privileges to access them. The access control depends on carefulness of objects owners or anyone who is the responsible to control the information to access the object. The advantage of the DAC is that users have great suppleness but the problem of DAC to assurance integrity like as least privileges and also the separation of duty. It is more important to sharing information rather than protection of information [29].

DAC is working in the centralized level and also in the distributed level. The Centralized levels are those when the administrator can access the user Data and other information. The changes required accessing the data through the department. The disadvantages of centralized level are in the large organization because it is very overwhelming especially the administrators are outsourced. But in the distributed level it is good in the large organization because it has distributed levels, those that allow to known person to access the data and other information service. It may be any single member like manger or team leader. The main advantage of the distributed level is that the delay can be avoided when the administrator of account is away from the large area [28]. Following are the concepts of Discretionary access control.

❖ *Files and Data Ownership*

In a system every object has an owner. The owner of the resources including files, data, system resources and devices determines the access policy. So we can say that an object without an owner is unprotected. This means that owner of the resources is the person who creates the resources. Like file or directories etc.

❖ *Access rights and permissions*

In this policy control that an owner can assign access right and permission to the individual user or group for specific resources. Discretionary access controls can apply through following techniques.

- In the Access control list (ACLs) name specifies rights and permission that assigned to the subject for a given object. ACL provide a flexible method for applying discretionary access controls.
- In the Role based access control assigns group membership that is based on organizational or functional roles. This strategy greatly simplifies the management of access rights and permission.

3.3.3. Mandatory Access Control (MAC)

This policy determines by the system, not the owner. It is used in multilevel system, which process highly sensitive data like classified government and military information. It means access control policy decisions are away from the control of individual owner of the object. This is central authority that determines which information is to access the system and by whom. User change, cannot access the change system. The main thing of MAC is every object is label by security level and user is restricted for this security level. The MAC security conditions that are typically label in to the application and other system and it applies in the entire objects like application and other resources. It uses to protect the data that is secret and confidential against information revelation. In the enterprise level it is very difficult to classify. So that MAC is not proper in Enterprise level [29].

Let takes the example of Doctor, patient and Nurse. The secret data may consist of patient name with the appointment time. So that secretary of doctor can update or change the existing time. The nurse may not able to see the appointment information but it can change the secret information like patient blood pressure and weight because she is the requestor of secret and sensitive data. The doctor may update or change the Weight and blood pressure information because his classification is confidential and secret. So that he can view the patient level. It is not necessary for the doctors, which have access the sensitive level like view the patient appointment or not to care which patient is the next one. The above example shows the concept of data and grant access information with classified access trying to use it. It is good to hierarchy's structure in the MAC implementation that shows to access the secret data; means access the secret and confidential data. And access confidential means, to access the secret and confidential data. Another advantage is that the data can be level in the security label. So no unauthorized cannot be access it and it is good application for the group of user rather that similar needs [28]. Following are the concept of Mandatory access control

❖ *Sensitive labels*

In the Mac system, labels are assigned to all subjects and object. A subject sensitivity label specifies its level of trust object, which is required for the access. So in order to access a given object than the subject must have the sensitivity level equal to higher than the requested object [21].

❖ *Data Import and Export*

It controls information from one system to another. Which is the critical function of MAC based system and it must ensure that sensitivity labels are properly maintained and implemented. So that sensitive information is protected at all times.

There are two methods that are commonly used for applying mandatory access control.

❖ *Rule based access control*

In this type of control defines the specific condition for access to a requested object. All Mac based system implements a simple form of rule based access control, which determines whether access should be granted or denied by subject or object sensitive label.

❖ *Lattice-based access controls*

It is complex access control decision involving multiple objects or subjects. It is mathematical structure that defines lower bound least upper bound value for a pair of element like object and subject [21].

3.3.4. Detailed Description of Role base access control

The National Institute of standard and technology has started the project called RBAC project. The purpose of RBAC project is to design the access control model that could be standardized, good in design in security level, and self-system dependent. That has positive result in implementation. David Ferraiolo and Rick khun introduced which attempt to meet the requirements and scope of RBAC solution in 1992 [28].

The approach of role - based access control (RBAC) began in multi - user and multi - application on - line systems pioneered in the 1970s. The central idea of RBAC is that permissions are associated with roles, and users are assigned to befitting roles. This simplifies management of permissions. Roles are created for the different undertaking functions influence an alignment and users are assigned roles based on their responsibilities and qualifications. User's authority act as tender reassigned from one role to more. Roles responsibility represents when brand-new permissions being brand-new applications and systems are incorporated, and permissions responsibility be revoked from roles as needed [39].

The Role based access control is based on an individual roles and responsibilities in any organization. The roles are defined is to analyze the goal and structure of an organization and it usually for the security policy. Like in medical organization roles are nurses, doctors, patients and in the banks the roles are manger, director etc. These types of person require different roles to perform their function [23].

It is the approach to restrict the system to unauthorized user. It is the newer and alternative approach to Mandatory Access Control (MAC) and Discretionary Access Control (DAC) that we have already discussed above in this chapter.

RBAC differs from access control lists (ACLs) used in traditional discretionary access control systems in that it assigns permissions to specific operations with meaning in the organization, rather than to low level data objects. For example, an access control list could be used to grant or deny write access to a particular system file, but it would not say in what ways that file could be changed [22].

The main purpose of RBAC is to provide security administration and analysis. Recent revival of concern credit RBAC has focussed on general backing of RBAC at the application level. Influence the past, and today, specific applications hold been built with RBAC encoded within the application itself. Existing operating systems and environments accommodate support for application level adoption of RBAC. Such supports are basis to emerge agency wares. The challenge is to ascertain application – independent facilities that are sufficiently extensible, after all easy to appliance and adoption, to abutment a wide span of applications with little customisation [39].

3.3.5. Why RBAC is Useful?

The role is fundamental semantics that bases of access control policy. Using RBAC the system administrator will give the job to user that performs in enterprise level with access permission and authorization. Assigning roles to the user based on user responsibilities and their qualification. The benefit for administrator in enterprise level specifies the security policies that cannot be getting other access control and to dramatically make more efficient the typically onerous process of access management. The role security in the enterprise level lies in its perseverance with in enterprise computing example. The permission linked with role will change as the function with in the enterprise changes over time. Membership within the role may be clearly defined and created with user roles and assigning position. But the idea of the role is relatively constant. For example the bank may have constant turnover between checker and duties of checker that is changed over time. But the idea of the checker is remains same. Roles can be used to specify capability, dependability, and power task within the enterprise level [34].

As we know the RBAC supports the application level. Traditionally the application levels have used RBAC at the internal level within existing operating system and environment offers little application level with RBAC support. The RBAC has ability to relation between role, between roles and permission, and also between roles and user. For example two roles can be established in the same time but two users cannot do both roles at the same time. Roles required inheritance relation where one role inheritance assigned permission in the different role. The access control policy is the quality and idea component of user-role, role-role and role-permission. These components show that which particular user has assigned the role for the particular systems. The role-role relation defines the security like separation of duty and group of authority [35].

The role permission is predefined which means that assigning the role to different user with some permission. It is less technical to assigning the user roles as compare to assigning the permission role. Without RBAC it is very difficult to assign what permission is to access on which users.

The RBAC has directly assigned the role whether they are directly or indirectly by the system administrator. The policy obeys the given system result from the particular configuration of RBAC component that directly by the system administrator, because the access control system policy is change over the system life cycle [35].

The RBAC supports three-security level.

❖ *Least privileges:*

Only those permissions are required to the user that is to be performed within organization. This means no more privileges to the user that is necessary to perform his and her job function. It is the principal that controls the individual user having the ability to take unnecessary actions, which is harmful to the granting ability to perform desired functions. The question is that how can we assign the set of permission in the combination of function and duties to correspond user or as a role of user subject acting on behalf of user [26]. Least privilege provides the clear thought to separation of boundaries that is provided by access control.

❖ *Separation of Duty:*

Mutually exclusive role to complete the task like in the bank the clerk and account manger participate to issuing the account.

❖ *Data Abstraction:*

Read, write, execute operations which providing by the operating system. Abstract permission like credit and debit account in bank can be created.

3.3.7 RBAC Model

A role based access control policy is the control decision of a user to allow the action within organization. It is the form of mandatory access control, seeing the security supervisor is answerable to enforcing policies and users cannot pass access permissions on to other users. A role answerability act as best by acknowledge of operations (privileges) that a user or agree of users answerability bring about within an organization. An ambition director allocates authorized operations to objects the roles. Besides, membership of users influence a role is again amen and revoked by a security executive on the basis of user's specific task responsibilities and qualifications.

There are following a part, which shows RBAC access control Model.

1. Roles are allocated based on the organizational structure with importance on the organizational security policy.

“2.Each role is assigned its profile, which includes all authorized command, transaction and allowable information access” [23]. It means that every organization assigned the role to the user, which includes all the access command and the information resources. So that user can perform the task with given information resources.

3. The administrator is based on comparative relationship with the organization and its assign roles like bank offices has some authorized transaction in bank and they can do some roles to employee.

4. Permission based granted is based on principal of privileges.

5. Roles can be Determined with separation of duties.

6. Roles can be transferred and assigned using sign off procedure.

7. Roles can be activated statically and dynamically as certain relational level.

8. Security and program manager managed the roles centrally [23].

The main conventions are useful in RBAC model

- U = User = A person which perform the role.
- R = Role = roles which defines a job level.
- P = Permissions = an approval to use and access to a resource.

- S = Session = A session for the particular user
- UA = Assigning user role.
- PA = Assign permission.
- RH = role Hierarchy.
- A user can do the multiple roles and role can do the multiple users.
- A role may have more permissions and permission can be assigned to multiple roles [22].

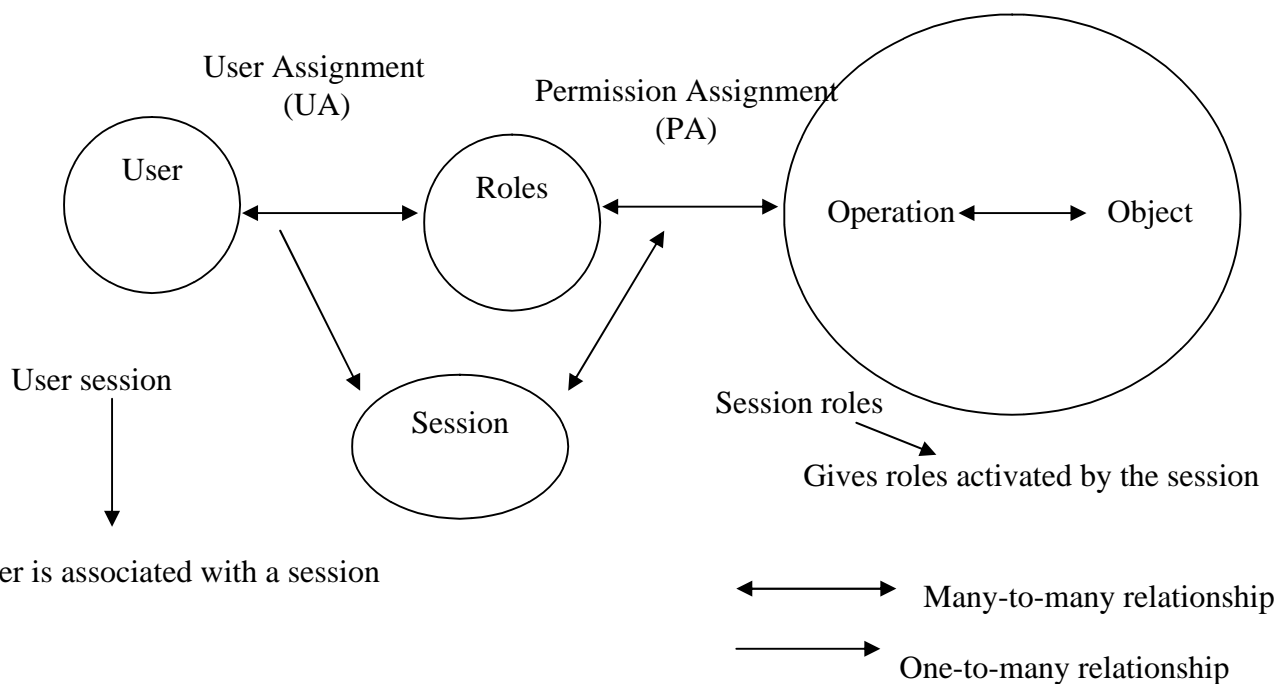


Figure 2: RBAC Model [35]

The basic model of RBAC defines the role, user, permission and Constraint.

❖ *User*

User is the main object, which access to the computer system. The role of single user corresponded to one person. The role is the main structure in the access control. The administrator establishes the role and assigns the role to the particular user according to the RBAC system. [30].

❖ *Permission*

The approval of access to one and more objects called permission. We are usually use the terms called authorization, access rights and privileges as permission. Permission is the action that is positive to exchange their information and ability to perform the action in the computer system. It is the special access modes like read write and update for one or more object in the system. The permission nature depends on system type and implementation. We use permission as a single or more objects like in the one file to access or more files in the particular department. The manner in which that individual permission is combined to the generic permission that can be assigned in the single unit and it is highly dependent implementation [35]. In RBAC user is assigned some roles and every role have some permission. The permission shows type of information where user can access the role with given permission. In RBAC is the meaning of authorization, access right execute, and privileges [30].

❖ *Role*

Role is the job function to the organization given by assigned user with some authority and responsibility for the given task. In the basic level roles can equal to the group which shows collection of user and user can be responsible of one role and multiple roles.

It is the semantic control of RBAC model. With the RBAC the role can create job function in any organization. Some permission access to the given roles and in the given roles user has some responsibility and some job function.

❖ *Roles and User*

In the RBAC user has assigned the roles based on the competencies and their responsibilities in the organization. The main task of user is to perform the role according to his given operation. User membership into role may be cancel and create again as job an assignment dictates. Roles can be created when new operation is introduced. And operation can be removed as organizational function changed and removed. This simplifies the administration and management of privileges. Roles can update the role without updating the role for every user on individual basis [27].

When assigning the role to the user. He has no more privileges that are necessary to perform. This concept shows the least privileges that determine the user job function. The minimum set of privileges requires performing the function and restricting the user to a domain with those privileges.

❖ *Roles and the Operation in Role Based Access Control*

Some companies can create the rule of operation with the roles. Like in the hospital provide the roles of nurse that responsibility to check the particular patient and not to disturb the other patient. The operation can particular in manner, which can be used in the expression and cause of rules and regulation.

“Operation gives you the unit of control that can be reference as individual role. It captures the complex security relevant detail or constraint that can be determined by a simple mode of access“[27]. Which means Operation is the single role that is to be performed by individual user. It gives you the safety that is determined by different modes. And different user has to

perform given access role. Let's take the example of two different roles in the bank, teller and accounting supervisor. The teller roles to perform the saving deposit operation. He will read and write the field to saving account paper. But accounting supervisor may not perform withdraw and deposit operation but only check the correction of the given saving account and in the other way the teller cannot do the correction when the transaction may complete [27].

❖ *Session*

The session is created when the user activates the given role that they are belonging to. Each session user can perform many roles. Each session is linked with one user and linked is constant for session duration. A user can open multiple sessions at same time and each have different window on workstation screen.

It is the part when the user sets as a part of the role to login the system. It can be one or multiple rights in one session. Session may combine different active roles. A user belong the several roles, which can use any subset of them to complete in single session [35].

❖ *Constraint of Role based access control*

It can be used to reproduce the policy of organization in RBAC operation. In order to stop the illegal action, One user cannot do the elite roles at the similar time called static separation of duty. Another one is user can have elite roles but not to allow the same role in the same time called dynamic separation of duty. And last the number of user assigned particular roles is limited called cardinality and the precondition role for obtaining role is called prerequisite [30].

The idea of constraint in role based access control is to layout of the organization policy example equally disjointing role. Once particular roles are declared to be equally exclusive, we don't need to concern with the individual user roles.

It can be specified in application or system level with or without being event happen. Which means particular cause constraint will be applied. The general approaches will not common to all cases.

For using constraint we need some suitable language with some function. Language should be simple and untreatable. So it can be used without any training and also it can be suppleness and fertility for the security reason. Because some time the language needs some basic so it is to be clear and exact [24].

We can specify the constraint in many ways. One is the invariant that will be grasped all time, and another is the precondition of the function like add the role of the user. Constraints in RBAC gives the restriction called invariant and change states call precondition. It is good advantage in theoretical way but it is not good for the real system. In RBAC Constraint it's include roles, permission, user and session

❖ *Roles Constraint*

1. Role ID defines the role.
2. Permission defines the permission of all roles.

3. User defines role of the user object.
4. Parent roles define the senior role.
5. Child role defines the junior role.

❖ *Permission Constraint*

1. Permission ID define the permission
2. The Constrain operation shows the action of the permission.
3. The target list constraint shows in the parameter as object where operations apply. [24].
4. Roles sets, which contain the all roles, object to support this permission.

❖ *User Constraint*

1. User Id defines the user.
2. Role set defines the reference to all role objects, which support by the user.
3. The Session set shows as a reference to active session object user [24].

❖ *Session Constraint*

1. Session id defines the ID session
2. User constraint, which submits this session
3. The Roles sets take the reference to all objects given by this session [24].

3.3.7. Roles Hierarchies in Role Based Access Control

In Role based access control, the Roles have some jobs and rights. And users have different roles to perform operation. Some of the roles have performed by all users. For this situation it would be unproductive and administratively unwieldy to specify repeatedly these general operations for each role that has been created [27]. The hierarchies of the roles provide the natural structural for the particular enterprises. A hierarchy of the role defines the single attributes and they have some roles. So that one role may unreservedly include the operation that can be linked with another role. It is the major component in the RBAC model. Role position in the role hierarchy shows the role that assigned is known and to be contained by other role in the role hierarchy. For example if Role A is inheritance to role B that means all the permission can use in the role A. The role hierarchies are established through the role inheritance among those roles [5]. “Role hierarchies are natural way of organizing of roles; reflect authority, responsibility, and competency” [27]. We can say that the role hierarchies are use for the organization that effects the role permission, role job and role capabilities. The concept of role hierarchies in the RBAC allows the one role to another. It means that one role includes the privileges and constraints that are linked with another role. The role hierarchs has following properties.

❖ *Property 1*

If user has authorized to perform the role and that role contains the other role. So the user can perform other role called role hierarchy. The association of user with the principle of roles has least privileges and separation of duty. There is the constraint requires with some role can be access to the particular user with certain given time. Operation is granted only those users who has least privileges in the role with respect to the RBAC needed to perform this role [38]

❖ *Property 2*

A user has access as a member of the role and that role is not mutually exclusive for that the user has already member. We can say that it's a static separation of duty. In RABC model we can restrict the number of user that allows the role in certain time. For example in the organization can give access the role to the different user as a manger for the certain time.

❖ *Property 3*

The role cannot increase through additional role member. The RBAC defines the activation of the role for the user and the role can be activated for the user.

❖ *Property 4*

It is also called the role authorization that shows that user has not an active role and not access for the user.

❖ *Property 5*

The user executes the operation only if the user is active in the active role called role execution. It means that active role cannot be mutually exclusive with other active role of the user. It provides the administration to effect on dynamic separation of duty. The difference between the static separations of duty places constraint is role authorization and dynamic separation of duty places constraint is role changing. For example the user can be access both the payment initiator and payment authorized but dynamically only one role is assumed in the same time.

❖ *Property 6*

A user is active in a new role only if the given role is not mutually exclusive with another role for that the user is currently active called dynamic separation of duty.

❖ *Property 7*

A user can execute the operation only the operation is authorized for the role in which the user is currently active called operation authorization.

❖ *Property 8*

The RBAC model defines the operational separation of duty that shows all the operation for the particular business function. No individual user allows accessing all the operation.

❖ *Property 9*

It is called object access authorization that shows that the user can access the object only role is the part of the active role. The role is permissible to access the operation and operation is access the object that is approved [38].

3.3.8. User and permission assignment

User and permission assignment are like as the multiple verses multiple relationship and it is the main part of the RBAC model. The main characteristics of the RBAC are to assigning operation to the essential role that completes the task (permission task). But it is place of assigning operation to the direct user. User is the become member of the corresponding role and can be complete the supported operation for the information object. This provides the acceptance of managing rights with many users and many information objects [30].

3.3.9. Comparing RBAC to MAC and DAC

RBAC is the ability to specify the access control policies and gives the shape to cover the authorization management. It gives the flexibility and control for existing MAC and DAC.

The definition according to the TSCE that DAC is an access control policy to access the user to allow and disallow the system and other access user object under their control.

DAC is to permit the access grant and permission to the right choice of individual user. It is the mechanism to access and granting policy to any of the object under their control without the interception to the system administrator.

In the enterprise level the end user has not allow to access there own object with the policy of DAC. For these organizations the computation is the actual owner of the system object and it is not allow to the user and may be away from the user. But in the role base access control the access decision is based in the role individual user that is the part of organization [36]. In the example of hospital Doctor, nurse, and in the bank includes accountant, checker, and manager. This policy is based on the action or performance that action which user has to allow acting within organization either is permission level or privileges level and user can not decline the permission to the other one based on the given judgment.

The Security level is used in the big organization to secure the organization objective. Like in hospital administration to secure information to the other users, the bank manager is to secure information to the accountant, and the teller. For these things needs some access rights required to control the system. The security administrator not the user that applies these policies must carefully represent the organization that specifies the access policy under organization resources.

RBAC is described in the MAC in sense that user has some policies and has no power over the action of organization protection policies. But RBAC is different from TCSE MAC.

MAC is defined as to restrict the access object that based on the label on that object with the formal authorization [37]. In the TCSE the Mac supports DOD requirement that protects an unauthorized access to the secret information and protect confidentially (definition in chapter 1). Systems that support MAC policies use unlawful information form up to down. It means its support only read and writes information. But control over writes to stop the unlawful sensitive information not with the integrity (define in the first chapter) information.

But in the RBAC is not only supports the confidential information but also controlling with confidently or integrity information or both that means which user access to what action. The difference between RBAC and MAC, RBAC is the non-optional access control and non

optional obey the law of protection policies that can be modified by enterprise-to-enterprise basis [37].

4. Research Review of Separation of Duty in Role Based Access Control

In this section we review previous work done on of Separation of duties and Role Based Access Control.

In his work titled “Separation of duty in computerized environments”, Sandhu [40] uses the payment of a check voucher as an example to illustrate the implementation of separation of duty in information systems. He identified three requirements for a system to implement the check voucher example, namely:

- Dynamic Separation of duties
- Hierarchical Roles
- Substitution of Attribution

He classified the objects within an information system into transient and persistent objects. The Transient objects have a finite sequence of steps, which disappears from the system. They have a complete history. Persistent objects have an unbounded existence in the system and an infinite sequence of steps is applied to them. He stated that the first level of defense was to limit modifications to the database to only users executing correct transactions. He proposed a second line of defense to effectively enforce SOD by partitioning the objects in the database into transient and persistent objects and then enforcing controls on the transient objects. Persistent objects get modified as a result of the modifications done on transient objects. In the example the transient objects are the vouchers and persistent objects are the accounts. He showed that this mechanism is able to enforce dynamic SOD, Hierarchical Roles and Substitution of Attribution.

Simon and Zurko [41] identified the various variations in separation of duty and categorized them. In role-based environments assigning users to roles controls access. These roles are defined by first grouping according to users who may act in a role, operations or actions comprising what may be done in the role, the object or target to be acted upon. They identified three types of constraints on the roles: Role membership, role activation and role use constraints. They categorized separation of duty into Static Separation (strong exclusion) and Dynamic separation of duty (weak exclusion). Static Separation Duty none of the users is allowed more than one role. It implements the role membership constraint. Static separation of duty is too rigid to meet the separation of duty requirements of organizations. Dynamic separation of duty makes use of role activation and role use constraints to make it possible to implement a richer set of separation of duty requirements of organizations. They identified four variations in this policy namely:

- Simple Dynamic Separation of duty
- Object-based separation of duty
- Operational separation of duty
- History based separation of duty
-

They also identified that History based separation of duty as a combination of both the object based and the operational separation of duty. Each step may consist of order-dependent or order-independent actions.

Let takes the case study of Role Based access control system in a major European bank was conducted by Schaad et al [42]. They discussed the overall structure of the system and compared it to the RBAC96 model. In addition to that they were also able to answer the question of the relationship between the number of users and the number of roles in the system. The system used was not specific to a single operating system or application. The FUB system is a RBAC system the roles are defined by the combination of the official position and job function. One of the weaknesses they identified in the FUB system was that a user could be assigned to more than one role and upon logging in had access to all those all the access rights in the roles. This was at variance with the RBAC96 model which only one role can be active per session. They also discussed inheritance and were able to make a distinction between inheritances as it occurs along official positions and inheritance between job functions.

Clark and Wilson's defined the commercial security policy for integrity [44] identified Separation of Duty because one of the two better mechanisms to opposite fraud and error time ensuring the correspondence between data objects within a system and the real world objects they appear as. At the policy aligned, processes were divided into steps; with each step as performed by a different person. Accordingly Separation of Duty is tightly responsible to application semantics or commands. Clark and Wilson suggested further safeguarding rail collusion by random selection of the sets of humans to perform some operation, consequently that bit proposed collusion is alone safe by chance[11].

Clark and Wilson defined the Separation of duty in security policy for integrity along with well-rounded transaction of two major mechanisms that control fraud and error. The use of well-rounded transaction shows that the computer information which are in the system internally reliable and separation of duty confirmed that object of the physical world are reliable with that information of object in the computer system. They also explained that computer does not access to monitor in the real world, which means that computer cannot check the system directly to the external constancy [44]. Whenever the association is ensure indirectly separate subparts in all operation and requiring the subparts perform by different users.

The information security literature shows the idea of SoD that comes into view in Seltzer and Schroeder with the name of "Separation of privileges" [45].

The Roger Needham is the making of surveillance in 1973 that shows a protection mechanism that requires two keys to unlock it is more robust and flexible than one that requires only a single key. There is no single mishap, dishonesty or break of trust is sufficient to assist the protection information [11].

Sandhu's work on Transaction control Expressions [46] introduced symbols for Dynamic Separation of Duty. Roles were used to describe who burden problem which transaction steps. However, influence Sandhu's model each user executing a step agency a transaction had to represent altered. To enforce this, the history of the implementation of each corporation was maintained. The constraints specifying the roles that could accomplish each move were associated with an object. These constraints turned into the history specifying which user executed each step on that object. Hierarchical roles were binding either on object or an extensive code. A weighted voting syntax allowed the specification of complicated person authorizations on a particular step on a particular entity

Nash and Poland's study of portable security device used ascendancy the asking apple raised a figure of different issues around Separation of Duty. The system determinate two disjoint groups of authorizing officers, and each day one or two officers from each of those groups were chosen as officer of the day. This was the aboriginal copy of the utility. Utility of specifying cardinality because a differentiating role and of age - based roles [11, 47]. Nash and Poland proposed the opinion of "object based Separation of Duty," which forced every transaction censure an object to act as by a contra distinct user. They suggested using Sandhu's Transaction Direction Expressions [46] to prolong the history of an object's transactions.

Ferraiolo and kuhn proposed the static and dynamic as a safety condition which shows that user as a member of the role is not mutually exclusive with any other role for which user have already gotten that one. That means no user cannot access the entire step that is needed to complete the task [49]. It is not the policy of Separation of duty but it is the requirement of Static mutually exclusive role.

Crapmton defined the set based approach in separation of duty. The access control system defined the state that as a set of sets and constraint is as a set, which would be forbidden in the system state. The system state will satisfied that if no element of the system state is the superset of constraint then the restrictive between different constraints is discussed [51]. In RBAC each session has only one user. And the task cannot be completed in one task. Several sessions are needed. Let takes as an example that permission place the order and issue the payment are two different roles. Which shows that these mutually exclusive in a DMER constraint. One can start the session, actives the role with the order permission, creates the order end the session. Start another session activates the role with payment permission and the access the payment against the order. This breaks the SoD policy.

Liner explain the use of Static separation of duty for those, which are from program development side. Its means that installed the software should be separated from those, which develop the software. The purpose for doing the programmer is making it harder to insert secret code without any one knowing or seeing [52].

Jarger and tidswell used the constraint and inheritance for Dynamically types access control model to implement the separation of duty [53,54]

Ferraiolo, Cugini, and Kuhn's paper on RBAC [11,31] presented the beginnings of a formal model of RBAC. They have three kinds of Separation of Duty. The antecedent two were Static Separation of Duty and Dynamic Separation of Duty. These variants were presented influence previous assignment. The third was Operative Separation of Duty, which introduced the impression of a "business function", and the set of operations required for that function; a career function resembles the opinion of job and duty any agency [11,55]. The formal bearing of Operational separation of duty stated that no role answerability admit the permissions to actualise all of the operations essential to a single calling function. These forces all business functions to desire at first two roles to represent used for their aftermath. The average description of Operational Separation of Duty assumes the roles involved keep disjoint memberships (Static Separation of Duty), forasmuch as that no single person has access to all the operations access a calling function [11].

There existence of wealth of literature on constraints other than SMER constraint in RBAC. It may be proposed and classified new kinds of constraint and new languages for specifying the complicated constraint which shows that two permission that are mutually exclusive with declaration of the one. It means no role can be access both permission. And two roles are dynamically mutually exclusive which they cannot be activated in the same session [50].

5. Case Study

5.1. Introduction

Our case study was carried out with the cooperation of a financial institution in Ghana. In this case study we used a questionnaire to elicit information from the Security Manager of the Banking application. These questions were arrived at based on our literature review. We start the chapter by giving a brief description of the bank, we then discuss its banking application. We then examine the access control system within the banking application, how it implements roles and separation of duties and what controls are in place to ensure that it does not adversely affect the business of the bank.

5.2. The Institution

The financial institution used for our case study has a mission to provide financial intermediation and related services for a sustained and diversified agricultural and rural development. As a result of its mission the institution has branches in both rural and urban areas. The nature of its clientele requires a lot of transactions to be carried out by visiting the branches. Other delivery channels and payment systems are not well developed. The bank has 49 branches spread across all the regions of the country with over 500,000 customers across all sectors. Our case study focussed on the retail (Domestic Banking) division of the bank. The range of services offered within this division was:

- Development Banking
- Corporate (Commercial) Banking
- Domestic (Commercial) Banking
- International Banking
- Treasury Management
- Money Transfer
- Mondex

5.3. The Banking Application

To provide its range of services there are a number of applications used by the bank. The main application used by the Retail group is FLEXCUBE[®] RETAIL. The bank started using this application for its Retail banking operations in 2004. Prior to migrating to this system the bank was using another system from the same vendor. The previous system had many challenges from the view point of the security manager. The security system of that application had the following issues:

- In the previous system access rights had to be assigned individually. With branches spread across a wide geographical area and a large number of users, one of the challenges faced by the security manager was managing access rights of users. IT Audits frequently detected some users who had acquired more access rights than they were entitled to. This happened as a result of the users temporarily being assigned to certain responsibilities.
- The other challenge they faced was with the implementation of changes in access policies bank wide.

- Another challenge was with users retaining their access rights when they login from another branch or department. Within the branch users from different departments could collaborate to complete a transaction provided they have those access rights.
- The process of creating user profiles was also not subject to separation of duties. The security manager alone could initiate and complete the entire process alone.

The FLEXCUBE[®] RETAIL system is based on Open Systems Technology and uses a Service Oriented Architecture [25]. According to [25] at the heart of its architecture are systems for business support, operational support, and information management. These support systems offer services that are used by all services and products by provided FLEXCUBE[®] RETAIL. Examples of some the FLEXCUBE[®] products are: Current Accounts, Savings Accounts, Term Deposits, Demand Deposits, Loans and Overdrafts etc [25]. The system has a very detailed Security Management System (SMS) that provides Application Security, Role-Based Access Control and User Profile Definition [25].

5.4. Access Control System:

At the heart of Access Control in FLEXCUBE[®] RETAIL is the Security Management System (SMS). One of the purposes of this application is the maintenance and control of user access to the system. The Security Management System (SMS) acts as the interface between users and the FLEXCUBE[®] RETAIL system. It ensures that only authorized users have access to the FLEXCUBE[®] RETAIL system. Access control is centralized at the Head Office, all users are created at the Head Office and user attributes are also modified at the head office then downloaded to the Branches. Each user has a user profile associated with them and this profile details the user's identity, user number, passwords, access rights, language, user branch, categories and level. A user is attached to a template, which defines the main access rights of the user. Each user also belongs to at least one class, which is used to group users for the purposes of authorization. An authorizer can only authorize the work of a user belonging to his/her group. Access control in this system is Role-Based with roles being defined by the user's template and class. The system also provides access control to non-permanent staff e.g. consultants and temporary staff. These users have an expiry date on their user profile that is defined during the process of creating the user profile. The user profile is deactivated automatically when the expiry date is reached [25].

5.4.1. Roles

Within the bank, staffs are grouped into departments and within each department staff have different responsibilities they perform. The role of a staff is usually determined by their department and responsibilities within the department. In the system this same approach has been adopted where roles in the system are defined by the combination of templates and classes. With the templates being collection of user attributes such as access rights, login times etc and the classes the organizational units (Branches/Departments) to which the user belongs.

5.4.2. Templates

The system groups common access right requirements for users with templates. This makes management of access rights easier since any changes would have to be done to the template and. It also provides a uniform way of defining access rights for users and guarantees that all users belonging to a template have the same attributes. The bank has defined twenty

(templates) for its retail banking operations. The user attributes defined by templates are shown below:

- Templates are attached to branches. Some of these templates are used in only the head office, others are for only branches and some are used by both. Examples of some templates are Teller, Branch Manager etc.
- The applications are listed on the menu and for each item on the menu a template can be assigned the right to add, amend, authorize, cancel, delete, inquire and modify transactions.
- Some additional restrictions can be placed on transactions that can be performed by a template. The transactions can be limited to same branch, across branches, normal accounts, restricted, staff accounts and all accounts.
- The login times are also defined in the template. User logins can be restricted based on to specific hours, days, weekends and holidays.
- Currency limits for users are also set at the template level. An upper limit is set for each of the currencies that the user will be working with.
- Password lifetimes for all users belonging to the template are also set. The number of days it takes for the user passwords to expire are set at this option.
- The level of the template. In the system every template is associated with a level. The basic function of this is to assist in defining which applications a user can authorise. Every application has a level associated with it. Users can only authorize transaction from an application that has a lower level than that of the user's template.
- Each template is associated with a category of users that can be assigned to it. Users in the bank are categorised based on their responsibilities. There three(3) categories of users;

- Security Manager (SM)

The Security Manager has the highest privilege in the system. They have access rights that enable them perform management level operations. Their profile does not allow them perform application related activities such as cash transactions etc. They perform the following operations:

- User Profile Maintenance
- Template Profile Maintenance
- Class Profile Maintenance
- User Class Linkage
- Task Access Control
- System Security [57]

- System Operator (OP)

The activities of the system Operator are limited to only;

- End of Day processing
- File Transfer
- Archival/Retrieval

They are not allowed to perform any application or Security Manager related activities [57].

- Others (OT)

These are the normal users of the system; they have access to the system applications. They cannot perform any Security Manager or System Operator activities. Their role within the system includes:

- Table Maintenance
- Performing Host Transactions [57]

5.2.4. Classes

Each valid user in the system belongs to at least one class or more classes. The main purpose of classes is to group users for the purpose of authorization. At the Bank classes are created based on Departments or units. Any action performed by a user (maker) belonging to a particular class will have to be authorized by an authorizer belonging to the same class. This prevents authorizers from authorizing transactions of users from different departments. For example all users in the Treasury Department belong to a class TR and users of the Loans Department belong to the class LN. An LN user cannot authorize actions of a TR class user unless that LN class user also belongs to the TR class (since a user can belong to more than one class).

5.3.5. User Id

Each user in the system has a unique user id. These user ids are created at the head office by the security managers. Each user ID has a password associated with it. Some user ids have a secondary password. The bank has 821 active user ids in the system and all these user ids are attached to the various branches/departments. The user id has the following attributes:

- User name: The user names is unique for each user across the bank.
- Branch template: this si the template to which the user is attached. A user can be attached to only one template and this template defines the access rights of the user.
- Profile start and end dates: The profile start and end dates are used to for temporary staff whose user profile has an expiry date marked by the profile end date.
- Vacation start and end dates: This attribute is used to disable a user whilst they are away on vacation.

The process of creating a user id takes care of the first linkage of assigning users to templates. The second process of linking users to classes completes the process of assigning a user to a role. This complex relationship is illustrated below.

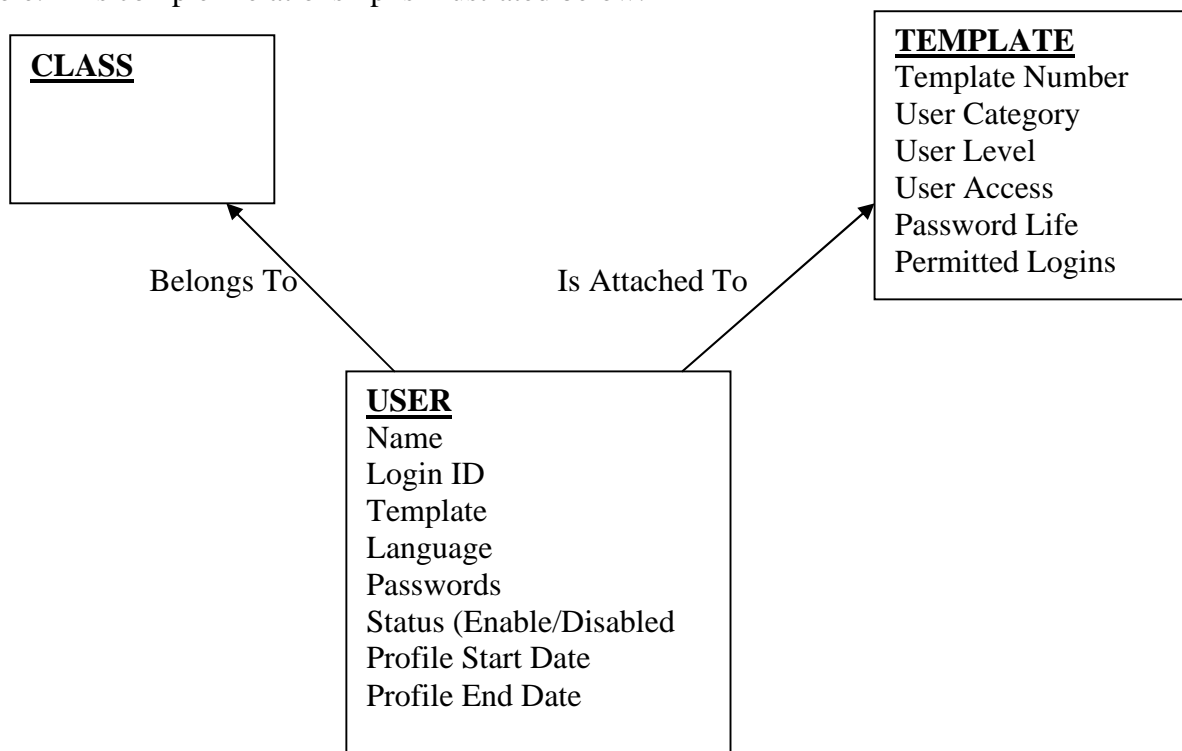


Figure 3: Relationship between Class, user and templates [57]

5.2.6. Separation of Duties:

This system implements a rich set of conflict of interest policies that ensure a secure system.

- All administrative activities are subject to separation of duties to prevent any security lapses. A maker/creator who is usually clerk or an officer initiates a transaction in the system. A supervisor who after reviewing the transaction authorizes the transaction to complete the transaction. The supervisors in the system do not have access rights to create or initiate any transactions.
- To prevent corporation between users belonging to different departments, authorizers can only authorize transactions originated from a branch department they belong to.
- The system screens all transactions and ensures that the maker and the authorizer are not the same user, the maker and authorizer belong to the same class, and the level of the authorizer is higher than that of the maker.
- Users have currency limits beyond which they cannot initiate or authorize transactions.

5.2.7. Assessment

We use the account opening procedure at one of the branches to further illustrate how the system implements separation of duties. The main roles at a branch are that of the Tellers, the chief cashier, customer service officer, and operations manager. When a prospective customer visits the branch, they first meet with the customer service officers who discuss the various products available to the prospective customer. The customer is then given the account opening forms for the product of their choice and fills it out. The customer service officer then logs into the system and creates the customer account using the account maintenance menu. The system generates an account number for the customer. The customer account created is not complete and cannot be used for any transactions until it has been authorized. The customer service officer sends the account opening documents to the operations manager who checks that all the requirements have been met before authorizing the customer account. When the account is created the customer is able to do transactions on the account. When the customer wants to make a cash deposit, they fill a deposit form and present it together with the cash to the teller. The teller then updates the account details with the deposit made. In the case of a cash withdrawal the customer fills a withdrawal form and presents this to the Teller. The teller then debits the customer's account and issues the customer with the requested amount provided the customer has sufficient balance. In the case where the amount exceeds the debit limit of the Teller, they request authorization from the Manager before they can complete the transaction. The Chief cashier manages the vault and receives or issues cash from the vault to the tellers on request. The manager authorises transactions initiated by the chief cashier. In this example the task of authorization is done solely by the manager, opening a customer account, cash transactions and vault transfers are done by the customer service officer, tellers and the Chief cashier none of the individuals can on their own complete the task.

We make an assessment of the separation of duties in the above business process using the approach suggested by [56]. This approach makes use of a matrix in which the above business process functions are listed. This matrix is used to show the functions that are compatible and do not create a conflict of interest within the organisation. In the matrix an **X** indicates a conflict of interest that is not compatible with Separation of Duties.

Functions	Create/Modify customer Accounts	Debit/Credit customer accounts	Transfer To and from Vault	Authorize Account Creation/Modification	Authorize customer Debit/Credit	Authorize vault transfer
Create/Modify customer Accounts		X	X	X	X	X
Debit/Credit customer accounts	X		X	X	X	X
Transfer To and from Vault	X	X		X	X	X
Authorize Account Creation/Modification	X	X	X			
Authorize customer Debit/Credit	X	X	X			
Authorize vault transfer	X	X	X			

Table 1: Branch Process Separation of Duty Matrix [56]

From the table above clearly Tellers initiating and completing transactions presents a conflict of interests within the organisation. This was clearly at variance with the separation of duties within the system.

6. Discussion

As we know that the Access control mechanism in enterprise level is very difficult because of number of users, information object, and various kinds of Security policy like subject to object security and organization structure and interrelation of business process etc. so access control mechanism DAC, MAC and RBAC have little limitation applies in the Enterprise level. RBAC policies can be modeled such that can be easily be integrated with the application that easy to understand analysis and use.

- From our case study the bank used Role based access control to manage the administration of user profiles. Users were assigned to roles based on their work function and the departments or branches to which they belong. This enabled the bank ensure that there is uniformity in access rights assignments and mitigated against the risk of a user acquiring access rights that are beyond their responsibilities or departments.
- The bank made use of separation of duties to prevent conflict of interests, fraud and also for error correction. The bank implemented a strict mutual exclusion of roles. Users in the supervisor role could not initiate or create any transactions. Even though within the hierarchy of the bank a supervisor should be able to assume the access rights of a teller, the mutual exclusion of roles ensured that their access rights only limited them to the authorization of transactions initiated by their subordinates.
- The bank implemented static separation of duties. For any given type of transaction a user could only perform one operation. Customer service officers had access rights that limited their activities in the system to only the creation and maintenance of customer accounts. They cannot perform any teller activities neither can tellers perform any customer service activities. The advantage of using static separation of duties lies in its simplicity in reducing the risk of conflict of interest, fraud and errors. However, the disadvantage of applying such a simple policy is that in the event that there is an absence of the users responsible for the performance of some important task the operations of the bank would be adversely affected.
- The system also met the separation of duty requirements of the bank by ensuring that users belonging to the same department or branch can only complete transactions. This way there can be no collusion between users belonging to different departments to complete a task. Another separation of duty requirement was also that users should only be able to log in from their branches. The system implemented this by attaching users to a host and branch template. This way the user can only belong to one branch.
- We made an assessment of the Separation of Duties in one of the business processed in the bank using an approach from [56] and this answered our research question. Our assessment indicated that tellers were able to initiate and complete transactions, provided it is within the currency limit of the teller's role. This exception is as a result of the large number of transactions processed daily by tellers without it there would be long waiting times at the branches which would in turn adversely affect the Bank. The bank however put in other controls to reduce the risk and impact of fraud. One of such is to ensure that the transactions are within a limit acceptable by the bank and also all transaction are recorded in a transaction journal which is reviewed by the Internal Control unit.

7. Conclusion

In this thesis we have described Separation of Duty and Role Based Access Control System with its kind and their property which shows that how these are implemented in the organization and Enterprise level. Also we have shown a Case Study where an organization implemented of Role based access Control and Separation of Duty and make an assessment of it. We also discovered that an exception was made to allow the bank meet its service delivery needs. An interesting area of future work will be an assessment of the effect of Separation of Duties in the bank and a measure of its impact.

References

- [1] [Su et al, 2006] X. Su, D. Bolzoni, P. van Eck, R. Wieringa “*A Business Goal Driven Approach for Understanding and Specifying Information Security Requirements*”, 2006
- [2] [SIS, 2001] SIS, “*SIS-ISO/IEC 17799:2000 Information Technology –Code of Practice for Information Security Management*”, Swedish Standards Institute, 2001
- [3] [Schweitzer, 1996] J.A Schweitzer, “*Protecting Business Information: a manager’s guide*”, Butterworth-Heinemann, Boston, 1996
- [4] [Denning, 2000] D. E. Denning, “*Information Warfare and Security*”, Addison-Wesley, Reading, MA, 2000
- [5] [Ferraiolo et al, 2003] D.F Ferraiolo, D.R. Kuhn, R. Chandramouli, “*Role-Based Access Control*”, Artech House, MA, 2003
- [6] [Anderson, 2001] R. Anderson, “*Security Engineering – A guide to building dependable distributed systems*”, Wiley, NY, 2001
- [7] “*Role-Based Databases Security, Object Oriented & Separation of Duty*” by J Matunda Nyanchama & Sylvia Osborn * email: {matunda, sylvia}@csd.uwo.ca October 11, 1993
- [8] “*Constraint Generation for Separation of Duty*” by Hong Chen Dept. of Computer Science Purdue University chen131@cs.purdue.edu Ninghui Li Dept. of Computer Science Purdue University ninghui@cs.purdue.edu
- [9] “*Separation of Duty*” last modified Mon Jan 9 13:56:57 EST 1995 <http://hissa.nist.gov/rbac/paper/node6.html> by John Barkley
- [10] A case study of separation of duty properties in the context of the Austrian “*eLaw process*” by Andreas Schaad & Pascal Spadone SAP Research 805 Av. Maurice Donat 06250 Mougins, France {andreas.schaad,pascal.spadone}@sap.com Helmut Weichsel Federal Chancellery Vienna Ballhausplatz 2 1014 Vienna, Austria helmut.weichsel@bka.gv.at
- [11] “*Separation of Duty in Role-Based Environments*” by Richard T. Simon The Open Group Research Institute Eleven Cambridge Center Cambridge, MA 02142 by r.simon@opengroup.org Mary Ellen Zurko The Open Group Research Institute Eleven Cambridge Center Cambridge, MA 02142 m.zurko@opengroup.org
- [12] Seltzer, J.H., and Schroeder, M.D. “*The Protection of Information in Computer Systems,*” Proceedings of IEEE, 63(9), 1278- 1308, 1975.
- [13] Clark, D.D., Wilson, D.R. “*A Comparison of Commercial and Military Computer Security Policies*” Proc. 1987 IEEE Symposium on Security and Privacy, 184-194, April 1987.
- [14] Nash, M. J., Poland, K. R. “*Some Conundrums Concerning Separation of Duty*” Proc. 1990 IEEE Symposium on Security and Privacy, 201-207, May 1990.

- [15] J. Tidswell and T. Jaeger. “An access control model for simplifying constraint expression”. In Proc. ACM Conference on Computer and Communications Security (CCS), pages 154–163, 2000.
- [16] “Approvability” by Jon A. Solworth University of Illinois at Chicago 851 S. Morgan Street, M/C 152 Room 1120 SEO Chicago IL 60607-7053 solworth@cs.uic.edu
- [17] Security Engineering: A Guide to Building Dependable Distributed Systems Chapter 9 Banking and book keeping by Casey Schaufler
- [18] Kuhn, D. R., “Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems,” Proceedings 2nd ACM Workshop on Role-Based Access Control, Fairfax, VA, 1997. pp. 23– 30.
- [19] Ferraiolo, David. “Role-Based Access Control”. Norwood, MA, USA: Artech House, Incorporated, 2003. <http://site.ebrary.com/lib/bthbib/Doc?id=10081904&ppg=135>
- [20] Ferraiolo, D., Cugini, J., Kuhn, D. R. “Role-Based Access Control (RBAC): Features and Motivations” Proc. 1995 Computer Security Applications Conference, 241-248, December 1995. Wesley Longman, 1994.
- [21] “ Access control from From Wikipedia, the free encyclopedia” last modified 23:59, 23 October 2006 http://en.wikipedia.org/wiki/Access_control
- [22] “Role-Based Access Control From Wikipedia, the free encyclopedia” last modified 20:19, 20 November 2006 http://en.wikipedia.org/wiki/Role-Based_Access_Control
- [23] “Role Based Access Control, Chapter 8 Access Control and Authorization” <http://www.cgisecurity.com/owasp/html/ch08s03.html>
- [24] “Constraints for Role-Based Access Control” by Fang Chen and Ravi S. Sandhu George Mason University, Fairfax, VA fchen@issc.gmu.edu
- [25](I-flex, 2006) i-flex solutions” FLEXCUBE”; web page: http://www.iflexsolutions.com/iflex/pdf/website/flexcube_corebanking.pdf, Last accessed: 18-12-2006
- [26][Anderson, 2001] R. Anderson, “Security Engineering – A guide to building Dependable distributed systems”, Wiley, NY, 2001
- [27]“An Introduction to Role-Based Access Control” NIST/ITL Bulletin, December 1995 <http://csrc.nist.gov/rbac/NIST-ITL-RBAC-bulletin.html>
- [28] “Role-Based Access Control”: The NIST Solution by Name: Hazen A. Weber Date: October 8, 2003 Certification: GSEC Version: 1.4b Option: 1

- [29]. “*Task-Role Based Access Control (T-RBAC)*”: by An Improved Access Control Model For Enterprise Environment Sejong Oh 1, Seog Park 1 1 Sogang University, Dept. of Computer Science, 121-742, Seoul, Korea {sejong, spark}@dmlab.sogang.ac.kr
- [30] “*Implementing Web Access Control System for the Multiple Web Servers in the Same Domain Using RBAC*” by Concept Won Bo Shim', Seog Park2 Dept. of Computer Science, Sogang University cool96@chcli.ac.kr1, spark@dmlab.sogang.ac.kr2
- [31] Ferraiolo, D., Cugini, J., Kuhn, D. R. “*Role-Based Access Control (RBAC): Features and Motivations*” Proc. 1995 Computer Security Applications Conference, 241-248, December 1995. Wesley Longman, 1994
- [32]. “*A Policy Based Role Framework for Access Control*” Email: C. Lupu, Damian A. Marriott, Morris S. Sloman, and Nicholas Yialelis Department of Computing, Imperial College, 180 Queen’s Gate London SW7 2BZ. UK E-mail: {e.c.lupu, d.marrioa, m.slomsn, n.yialelis}@doc.ic.ac.uk
- [33] Drafts. “*Role Based Access Control Implementation Standard*” Version 0.1 January 2006
- [34] “*Future Directions in Role-Based Access Control*” by David F. Ferraiolo and D. Richard Kuhn National Institute of Standards and Technology ferraiolo@csmes.ncsl.nist.gov kuhn@nist.gov
- [35] “*The Role based Access control model*” By Ravi S. Sandhu George Mason University, and SETA Corporation, Edward J. Coyne Ha L. Feinstein Charles E Youman Seta corporation
- [36] Ferraiolo, David. “*Role-Based Access Control*”. Norwood, MA, USA: Artech House, Incorporated, 2003. p 17. <http://site.ebrary.com/lib/bthbib/Doc?id=10081904&ppg=35>
- [37] DoD, Trusted Computer System Evaluation Criteria (TCSEC), DoD 5200.28STD.
- [38] Natural Language Processing and Information System: 5th International Conference on Application of Natural Language to Information system, NLDB 2000, Versailles, France, June 2000, Revised Papers: IT- Security and Privacy: Design Use of Privacy- Enhancing Security Mechanisms Chapter 3. IT- Security.
- [39] IEEE Computer, Volume 29, Number 2, February 1996, pages 38-47. “*Role-Based Access Control Models*” by Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein and Charles E. Youman Revised October 26, 1995
- [40] (Sandhu, 1990) R. Sandhu; “*Separation of Duties in Computerised Information Systems*”, Proc. of the IFIP WG11.3 Workshop on Database Security, Halifax, U.K, 1990
- [41] (Simon & Zurko, 1997) R. Simon, M.E. Zurko; “*Separation of Duty in Role-Based Environments*”; Proceedings of the 10th Computer Security Foundations Workshop (CSFW’97), page 183, 1997
- [42] (Schaad et al, 2001) A. Schaad, J. Moffet, J. Jacob; “*The Role Based Access Control System of a European Bank: A Case Study and Discussion*”; Proceedings of the sixth ACM symposium on access control models and technologies, 2001

- [43] [Denning, 2000] D. E. Denning, "Information Warfare and Security", Addison-Wesley, Reading, MA, 2000
- [44] Clark, D.D., Wilson, D.R. "A Comparison of Commercial and Military Computer Security Policies" Proc. 1987 IEEE Symposium on Security and Privacy, 184-194, April 1987.
- [45] J. H. Seltzer and M. D. Schroeder. "The protection of information in computer systems." Proceedings of the IEEE, 63(9):1278-1308, September 1975.
- [46] Sandhu, R. "Transaction Control Expressions for Separation of Duties" Proc. 4th Aerospace Computer Security Conference, 282-286, Dec. 1988.
- [47] M. J. Nash and K. R. Poland. "Some conundrums concerning separation of duty". In Proceedings of IEEE Symposium on Research in Security and Privacy, pages 201-209, May 1990.
- [48] V. D. Gligor, S. I. Gavrila, and D. F. Ferraiolo. On the formal definition of separation-of-duty policies and their composition. In Proceedings of IEEE Symposium on Research in Security and Privacy, pages 172-183, May 1998.
- [49] J. Crampton. Specifying and enforcing constraints in role-based access control. In Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies (SACMAT 2003), pages 43-50, Como, Italy, June 2003.
- [50] T. T. Simon and M. E. Zurko. "Separation of duty in role-based environments". In Proceedings of The 10th Computer Security Foundations Workshop, pages 183-194. IEEE Computer Society Press, June 1999
- [51] J. Crampton. "Authorizations and Antichain". PhD thesis, Birbeck College, University of London, UK, 2002.
- [52] S. B. Lipner. "Non-discretionary controls for commercial applications". In Proc. IEEE Symp. Security and Privacy, pages 2-10, 1982.
- [53] T. Jaeger and J. E. Tidswell. "Practical safety inexible access control models". ACM Transactions on Information and System Security (TISSEC), 4(2):158-190, 2001.
- [54] J. Tidswell and T. Jaeger. "An access control model for simplifying constraint expression". In Proc. ACM Conference on Computer and Communications Security (CCS), pages 154-163, 2000.
- [55] Thomas, R. K., Sandhu, R. S. "Conceptual Foundations for a Model of Task-based Authorizations" Proc. of The Computer Security Foundations Workshop VU, June 1994
- [56] [Fox & Zonneveld, 2006] C. Fox, P. Zonneveld; "IT control Objectives dor Sarbanes-Oxley: The Role of IT in the Design and Implementation of Internal Control Over Financial Reporting, 2nd Edition", IT Governance Institute, 2006
- [57] [Iflex, 2004] iflex; "Flexcube implementation manual", 2004