Chapter 6

Authorization

Authorization is an important security aspect of a DPW. All subjects of an office are not authorized to access all objects. In a DPW the subjects are the users of the office system and the objects are mainly the pages belonging to a page cube. Authorization in a DPW environment is dynamic in nature. Along with the document flow, authorization also flows synchronously. The existing authorization models for general workflow are briefly reviewed in this context and found that they are not sufficient to address the issues of authorization in case of a DPW. In an office environment, who accessed which object, at what time, is also an important aspect for office security. Therefore, an authorization audit trail is also equally important. In this chapter, an authorization policy and an authorization model suitable for a DPW are discussed.

6.1 The Authorization Policy

Before discussing the authorization model, we discuss here an authorization policy which comprises of the following elements:
1. Authorization Administration: A centralized policy for authorization administration is adopted. Since an in-line TTP called an arbiter is mandatory for secure production of MPMSDs, therefore the arbiter is also made the central authority for authorization administration. Only the arbiter can grant or revoke authorization to or from a subject for an object.

2. Access Protocol: A subject other than the arbiter cannot access the objects directly. The subject, submits an access request for an object to the arbiter. The arbiter finds out whether an authorization can be granted by consulting the set of authorization rules. If the request is as per rules then the arbiter grants an authorization, otherwise it denies the access. If an authorization is granted then the arbiter performs the access requested on the object on behalf of the subject and the result is communicated to the subject in response to the request.

3. Authorization Flow: The authorization in document production workflow is dynamic in nature. When a case flows from reviewer to reviewer, authorization on the case also flows from reviewer to reviewer synchronized with the document flow. Authorization on an object is granted and revoked based on the occurrence of a pair of conjugate events. For example, authorization of a reviewer on a MPMSD for reading previous parts and adding comments to the case is granted as soon as the document is received by the reviewer and the authorization is revoked as soon as the the case is sent to the next reviewer.

4. Object Privacy Policy: Information stored in the objects are always encrypted with an internal encryption key and the encryption key is known only to the arbiter. This is to enhance privacy to mitigate the
threat even from the system administrator of the host system.

6.2 General Authorization Models

The most general authorization model is probably the Discretionary Access Control (DAC) using access control matrix model, introduced by Lampson, and afterward developed by Graham and Denning and Harrison, Ruzzo and Ulman[32]. The model is used as a security model in operating systems and in database environments. The model represents the authorization as a matrix. Let $A$ be the authorization matrix. The matrix rows correspond to the subjects and the columns to the objects. Entry $A[s, o]$ contains the access modes for which subject $s$ is authorized on object $o$. Since the matrix is usually quite sparse, typical implementations adopt one of the three representations: Access Control List(ACL), Capability List(CL) and Authorization Table(AT).

The ACL approach represents the corresponding access matrix by column. To each object $o$, a list of pairs $(s, A[s, o])$ is associated indicating the subjects and their access modes on the object $o$. Therefore, only non-null matrix entries are considered. In this approach all subjects granted access on an object can be easily found; however, it is inefficient to look for all the objects a subject can access. The CL approach represents the corresponding access matrix $A$ by row. To each subject $s$, a list of pairs $(o, A[s, o])$ is associated for each object $o$ such that $A[s, o]$ is not null. Therefore, if a subject holds no rights on an object, this object does not appear in the list. In this approach, given a subject, all the objects, the subject is authorized to access can be easily found; however, it is inefficient to find out the set of subjects granted access on a given object. The paradigm of authorization in ACL as well as in
CL approaches is distributing the authorizations either object wise or subject wise. In ACL-based operating systems and databases, in general today, the owner of an object can grant or revoke or modify access rights. Similar is the case with CL-based systems. Moreover, the presence of a superuser or system administrator, whose power is unlimited, can be a major threat of security breach in office document systems.

The AT approach represents the access matrix by a central table of tuples $(s, o, A[s, o])$. This approach has also advantages and disadvantages. The AT is suitable for centralized authorization administration, where the authorization granting and revoking is done by a central authority. Finding a set of authorized objects for a given subject and a set of subjects authorized to access a given object is easier by simply querying the table. The document production in a DPW is done under a central authority, called the arbiter. In the document production workflow environment, where the authorization is dynamic and proprietary in nature, the authorizations for a reviewer are granted for a fixed time period only and the authorization propagates to the next reviewer, the authorization table representation is easier to manipulate. The main disadvantage of the AT approach is related to the size of the table. The table may contain many tuples of inactive subjects/objects making the table size large.

Role-Based Access Control (RBAC) appears to be unavoidable in all groupware solutions in office environments. A user is allowed to access an object on the basis of the role(s) that the user plays at the time of access. If the responsibility of a user $s$ changes due to transfer to another department of the office or due to promotion etc., current role of $s$ can be reassigned to his/her replacement, and new roles can be assigned to $s$ as required by the new responsibilities. It is not necessary to go through the ACLs of all the objects.
one by one to remove entries pertaining to $s$ in his/her previous capacity and add entries pertaining to the new capacity. Roles greatly simplifies the task of security management. Moreover, this enables a user to play multiple roles with different responsibilities, which is common in an office. RBAC also has a disadvantage in DPW. For example, If a document is sent to a reviewer and the authorization is granted to the role of the reviewer, then any other employee belonging to the same role can at least read the document, which may be a serious breach of security.

### 6.3 The Workflow Authorization Models

Synchronization of authorization flow with the workflow is a fundamental security requirement in workflow environments [4]. Other essential requirements include role based security policy and separation of duties [9]. Separation of duties are imposed to reduce the risk of frauds by not allowing any individual to have sufficient authority within the system to perpetrate a fraud on his own. WFMSs like Lotus Notes provide role-based access control but do not have a formal model to synchronize authorization flow with the workflow. Recently Atluri and Huang [4, 5] proposed a Workflow Authorization Model (WAM) that provides synchronization of authorization flow with workflow, role-based authorization and separation of duty. The WAM model properly addresses the issue of authorization flow when there are temporal constraints in the definition of tasks of the workflow. This means that there is a fixed time of starting and completing a particular task and these are defined a priori in the workflow template itself. The genesis of the model is in automatic time-bound execution of tasks and the corresponding authorization flow. Our problem is slightly different. We cannot set a priori the
time period for a review process. We need event-based authorization flow. Apart from temporal constraints, a DPW has other types of constraints as discussed in section 6.1.

6.4 The Authorization Model for DPWs

In this section we propose a DPW Authorization Model (DPWAM). Authorizations in DPW are event-based and dynamic in nature. By dynamic we mean that authorizations are granted to a subject, acting as a reviewer, on some objects based on occurrence of an event and the authorizations are to be revoked automatically on occurrence of the other conjugate event. That is, the subject will have authorizations for certain privileges only during the time period between the occurrences of the conjugate events. The conjugate events may vary from object type to object type. For example, a reviewer of an active MPMSD under review, will have the authorization for the privilege to read the previous parts and to comment on the document as soon as she receives the document. The authorizations will be revoked as soon she sends the document to the next reviewer. That means the reviewer will not be allowed to read or comment on the document after forwarding it to the next reviewer. Here receive and send are the conjugate events. In case of paper document, since the document itself is moved physically from the current reviewer to the next reviewer, therefore the revocation of authorizations is automatically accomplished. In a more complex situation, the revocation may be partial. Even after forwarding the document, the $i^{th}$ reviewer may be allowed to read the document, but only up to the $i^{th}$ part of the document. In a DPW, request and response for a page may also be conjugate events.

The page cube model can be extended in the following way to incorporate
event-based dynamic authorization model for a DPW. An office worker may play one or more roles in an office. Some roles may be grouped together to form a larger role. This forms a role hierarchy. Therefore, role may be a new dimension of the page cube. To take care of the disadvantage of general RBAC model for a DPW, discussed in section 6.2 we may assume that the leaf nodes of the role hierarchy are atomic and the internal nodes are non-atomic and consists of either atomic or non-atomic children roles. By atomic, we mean that only one user can be assigned to an atomic role. Normally in a DPW, atomic roles are assigned as reviewers. In an office, atomic roles can be specified from a non-atomic role. For example in a University there are three Assistant Registrar. AssistantRegistrar is a non-atomic role, which consists of three atomic roles: Assistant Registrar(Finance), Assistant Registrar(Academic), Assistant Registrar(Administration).

So far, we discussed only intra-dimensional graphs in our Page Cube model. Apart from the intra-dimensional graphs, limited within a dimension, like the category graph, a Page Cube may have Inter-Dimensional Graphs (IDG).

The DPWAM comprises of five components: User-Role Graph(URG), Role-DPW Graph(RDG), Rule-Base(RB), Authorization-Base(AB) and Authorization Audit Trail(AAT).

6.4.1 User-Role Graph

A user may be assigned to many roles and a role may be played by many users, if the role is non-atomic. Therefore, it is a many to many relation. An user may be assigned to a particular role at a particular time and the assignment may be revoked at some other time. The history of user role assignment may be captured as an IDG.
Definition 6.1 A User-Role Graph is a bipartite graph where user and role are the two types of nodes. An assignment is represented by an edge which connects a user node with a role node and the edge is labelled as \((t_+, t_-)\), where \(t_+\) is the time of plugging (assigning) and \(t_-\) is the time of unplugging (revocation of assignment). A user remains active in a role till it is unplugged.

In an office not only the active assignment of users to roles, but also the record of history of assignments is equally important for security auditing and dispute resolution. Using a page cube, this is easily accomplished.

6.4.2 Role-DPW Graph

A role may have access privileges to many DPWs and a DPW may have many roles as reviewers. Due to obvious reasons, definition of authorization rules based on role rather than on users is easier. A role may be assigned to a particular DPW at a particular time and the assignment may be revoked at some other time. The history of role-DPW assignment may also be captured as an IDG, called Role-DPW Graph.

Definition 6.2 A Role-DPW Graph is a bipartite graph where role and DPW are the two types of nodes. An assignment is represented by an edge which connects a role node with a DPW node and the edge is labelled as \((p, t_+, t_-)\), where \(p\) is the position of the role as a reviewer in the DPW, \(t_+\) is the time of plugging and \(t_-\) is the time of unplugging. A role remains active as a reviewer of a DPW till it is unplugged.

Within a DPW, the same role may appear as reviewer more than once but at different positions. A connected pair of nodes in both URG and RDG may have parallel edges with different labels.
6.4.3 Rule-Base

Authorization constraints for a DPW can be represented as a rule. A rule can be defined in Event-Condition-Action (ECA) paradigm. The structure of a Rule-Base is

\[ \text{Rule-Base}(\text{ruleId}, \text{event}, \text{conditions}, \text{actions}, \text{privileges}, t_+, t_-) \]

ruleId is the unique identifier of a rule. Event may be one of the pair of conjugate events: request, response. The conditions may be of two types: conditions on the subjects and the conditions on the objects. The conditions on the subject may be defined by the predicates on user, user-role graph, role-DPW graph etc. Whereas, conditions on objects may be defined by page profiles, dimension hierarchies, dimensional graphs etc. For example, path existential conditions used in the protocol for production of cases discussed in Chapter 4 is an object condition. The conditions may have a partial order. Subject conditions followed by object conditions. Within a subject condition, the partial order may be user > user-role > role-dpw. Similar is the case in object conditions. The actions are basically two: grant and revoke authorizations. The privileges may be read, write, plug, unplug, close, reopen, archive, dearchive and burn. \( t_+ \) signifies the time of inclusion of a rule in the Rule-Base and \( t_- \) signifies the time of exclusion of a rule from the Rule-Base.

On occurrence of an event, a rule is triggered. If the conditions are true, actions are taken: either to grant an authorization for certain privileges or revoke an authorization. An active rule-based authorization constraint modelling for general workflow is available in [12, 13].
6.4.4 Authorization-Base

Authorization-Base is a collection of authorizations granted and revoked during production of pages in different DPWs in an office. An authorization is a tuple as defined in Chapter 4, (4.5). Here subject attribute is replaced by three attributes: \((user, role, dpw)\). The modified authorization tuple is

\((user, role, dpw, object, privileges, t_+, t_-)\)

6.4.5 Authorization Audit Trails

In the DPWAM model, the size of AB grows fast. The growth of AB will have an effect on performance. Moreover, to keep AB small, we cannot simply delete the authorization tuples from AB as soon as it is revoked. Because, in an office audit trail of access of objects may be an important function. For example, who accessed a particular page, during a particular time can be found out from AAT. The access records are also persistent in nature. Therefore, the records are to be stored securely. One simple way is to move the revoked authorization tuples from AB to a Revoked Authorization-Base (RAB) of similar structure. The RAB will be maintained in reference storage and this will be accessed very rarely, only during audit. Hence its growth will have negligible effect on overall performance. Moreover, it can be archived. RAB will serve for audit trails for document access history in an office.

6.5 Discussion

In this Chapter authorization for DPW problem is discussed. The authorizations are dynamic in nature and moreover history of authorization is also
a security requirement of an e-office. Therefore, issues of authorization for DPW are addressed with the help of an authorization model proposed here. DPWAM is an extension of Page Cube model. The authorization management is a central one controlled by the arbiter. Since the arbiter is necessary for other aspects of security, as discussed in Chapter 4, therefore, authorization management is a natural extension of the responsibility of the arbiter.