

the following drawings that accompany the detailed description. The drawings can be briefly described as follows.

[0012] FIG. 1 illustrates the architecture of a general attribute-based access control system.

[0013] FIG. 2 illustrates the basic data sets and relations and the mapping of objects to their physical locations.

[0014] FIG. 3 illustrates an example of assignment relations in support of combined RBAC and MLS policies.

[0015] FIG. 4 illustrates a derivation of a capability of a user in a single policy class from the assignment relations.

[0016] FIG. 5 illustrates a derivation of a permission from the assignment relations.

[0017] FIG. 6 illustrates an object attribute (or container) that is accessible to a user.

[0018] FIG. 7 illustrates an example presentation of a personal object system (POS) with respect to FIG. 3.

[0019] FIG. 8 illustrates a portion of assignment relations that are configured to support a Discretionary Access Control (DAC) policy objective.

[0020] FIG. 9 illustrates a portion of assignment relations that are configured to support a Discretionary Access Control (DAC) policy objective, where user BOB has been granted read and write access to object PROPOSAL1.

[0021] FIGS. 10a-e illustrate portions of an assignment relation configured for the support of a workflow application.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0022] FIG. 1 illustrates the architecture of a general attribute-based access control system 20 (the “system 20”) for the specification and enforcement of arbitrary attribute-based access control policies. The system 20 may also be referred to as a “policy machine,” or “PM,” and includes one or more resource repositories 22, one or more client modules 24, an access control database 26, one or more server modules 28, and one or more resource servers 30.

[0023] The resource server 30 stores and retrieves computer-accessible resources referenced by objects to and from the resource repositories 22. The client module 24 authenticates users through an authentication scheme that maps human users to identifiers, executes programs within processes that run on behalf of authenticated users and are identified through unique process identifiers, issues access requests to perform operations on objects, and enforce access control policies with respect to the access requests. The access control database 26 includes basic data sets and basic relations between the basic data sets for specifying policy.

[0024] The basic data sets include data corresponding to the users, user attributes, objects, object attributes, operations, policy classes, and processes. The basic relations include assignments, prohibitions, and obligations. The assignments are used to derive permissions consisting of a user, an operation, and an object, where the pair operation, object is said to be a capability of the user. The prohibitions are representations of the capabilities of the users that are denied to users and processes. The obligations include conditions under which basic data sets and relations are obligated to change in a manner also prescribed in the obligations. The server module 28 computes decisions whether to grant or deny access to a resource referenced by an object to a process, processes events raised by the successful execution of an operation on an object, and assists in the administration of the access control database.

[0025] The client module 24 provides the context in which the user’s PM processes run. A user may log on to the PM by using an authentication sub-module of the client module 24. A successful login may open a user session for creating and

running various PM processes that request access to objects. A policy enforcement sub-module of the client module 24 traps each access request, and then asks the server module 28 to decide whether to grant or deny the access request. The server module 28 computes and returns the decision to grant or deny the access request to the process that issued it. In the case of a decision to grant, if the resource referenced by the object of the access request is stored in a repository, the server module 28 also returns the resource’s physical location. The policy enforcement sub-module of the client module 24 enforces the decision received from the server module 28, granting or rejecting access to the object. For a granted request the client module 24 may require the cooperation of the resource server 30 in performing the granted operation on the resource. The resource server 30 may reside on the same computer as the client module, or on a server dedicated to the storage of PM resources.

[0026] The server module 28 is a software module that receives an access request from the client module, computes a decision to grant or reject the access request, and returns the result. The decision is based on the identity of the user, the identity of the process that issued the request, the requested operation, the requested object, and the assignment and prohibition relations as stored in the access control database 26, as will be described below. The server module 28 also executes the responses to events specified in the obligation relations stored in the access control database 26 and raised by a client’s successful execution of an operation on an object. Finally, the server module 28 can be used to administer the access control database. The server module 28 exposes a standard set of commands that can be used by clients to solicit its services.

[0027] The system 20 provides a method that may be used to specify and enforce an arbitrary attribute-based access control policy. The method may include configuring the access control database to specify a desired policy (i.e., establishing the basic data sets and relations between the basic data sets), authenticating users and establishing and running processes within a session, trapping process access requests of the form (operation, object), computing decisions to grant or reject such requests by applying a reference mediation function, which determines the existence of a permission (user, operation, object) derived from the assignment relations such that the user is the process user, and the capability (operation, object) of the user is not denied for either the user or the process through prohibitions, and dynamically altering the current configuration of the access control database as prescribed by obligation relations that specify the conditions and the manner in which such alteration must take place.

[0028] FIG. 2 illustrates an example structure 40 of the basic relationships of assignments, prohibitions, and obligations between the basic data sets for defining a policy. The configuration of assignments and prohibitions defines the access state, and the access state and the obligation relations define the overall policy. The current access state defines the set of permissible user and subject (processes) accesses. Obligations may dynamically alter relations to include current access relations and obligations as a response to subjects accessing resources.

[0029] In the disclosed examples, some of the basic data sets are denoted as users (U), system operations (OP), objects (O) and processes (P). Users are unique identifiers associated to human users through an authentication scheme.

[0030] Objects are names (with global meaning) for computer accessible resources that must be protected, and perhaps shared, under one or more access control policies. For example, FIG. 3 presents an example of assignment relations