J. KSIAM **Vol.8**, No.1, 11-30, 2004

A MATHEMATICAL MODEL FOR HISTORY-BASED ACCESS CONTROL

HEE-YOUNG, KIM

ABSTRACT. Access Control is one of essential branches to provide system's security. Depending on what standards we apply, in general, there are Role-based access control, History-based access control. The first is based on subject's role, The later is based on subject's history. In fact, RBAC has been implemented, we are using it by purchasing some orders through the internet. But, HBAC is so complex that there will occur some errors on the system. This is more and more when HBAC is used with other access controls. So HBAC's formalization and model which are general enough to encompass a range of policies in using more than one access control model within a given system are important. To simplify these, we design the mathematical model called non-access structure. This Non-access structure contains to historical access list. If it is given subjects and objects, we look into subject grouping and object relation, and then we design Non-access structure. Then we can determine the permission based on history without conflict.

1. INTRODUCTION

Many different access control policies and models have been developed to suit a variety of goals. Universally, these include MAC, DAC, Chinese Wall Policy, Role-based access control, and so on. Since each policy has different strength, there are the notational differences. And then it is difficult to combine them. So there have been many computer science research about integration of these policies. In the early days, access had controlled by the level of user and directory. For example, a colonel can read the document with the top secret, a soldier cannot read that. So this access control is very restrictive. And since administrator can have all rights in the system, his authority is very higher than other users. Then this is also problem. Therefore in 1976,

Bell and LaPudula [2] proposed the combination of Mandatory Access Control and Discretionary Access Control. This uses the notion of security level and compartment. But this problem is that access in the same compartment and same security level is freely controlled. Besides, it is still restrictive. And many people needed to policy which is able to control this situation. In 1989, Brewer and J.Nash [1] presents a basic mathematical theory which implements Chinese Wall Policy and shows that it cannot be correctly represented by a Bell-LaPudula model [2]. These policy has drawn attention to the fact that objects can be leaked the partial information. In order to prevent partial information from leaking, Chinese Wall Policy uses the notion of Wall. Then these policy is concerned not about subject's leaking, but about object's leaking. If subjects contains same grouping, then subjects can share any information. Then the problem is that Chinese Wall is broken because of sharing with subjects. Suppose that these policies be combined. Then administrator must check these policies' consistency. He needs the standards which can check that consistency. And we can check these components efficiently. This is not easy because policies based on history deal with access case by case. Therefore we need to build general model to do that. In previous work, because there are access history list in history database, administrator look into that and then administrator determine whether the subject can access to the object or not. These behavior makes system not to work efficiently. In this paper, we will hold the general access control model. And this model is considered on history. First of all, we will consider 3 representations of access control. That is, we can conclude that many policies are classified to 3 kinds of representation, rank, subject grouping, object relation. MAC, DAC are related to the rank of subject and object. And Role-based access control is related to the grouping, Chinese Wall Policy is related to the relation of object. Based on these 3 representation, we will build the model, and then formalize this model's access control. Then we will check whether the system bring to conflict among many policies without one's knowledge or not. So we will consider 3 components, rank, object's relation, subject's grouping. And we will think the history database and then we will lay out mathematical model based on this. If this designs, then we will prevent from occurring an error in the system which uses different access control simultaneously.

2. The Basic of Access Control

A fundamental tenet of information security is controlling access to the critical resources that require protection from unauthorized modification or disclosure. The essence of access control is that permissions are assigned to individuals or system objects, which are authorized to access specific resources. In some instances, access controls are not invoked by hardware or software, but are administrative in nature the segregation of responsibilities. Then What is Security Policy? Security Policy specifies who is allowed access to which data items. It's sorts depends on the object. To implement each policy, we need each model proper to each policy.

Access control is the prevention of unauthorized access to the resources of an IT product, programs, processes, systems, or other IT products. Some suppliers consider preventing unauthorized users from logged on users accessing objects for which they have no authorization. The 'strength' of access control is often described in terms of 'factors'. The greater the number of factors, the stronger the control.

one-factor: password

two-factor: password + token

three-factor: password + token + biometric

four-factor: password + token + biometric + geography

five-factor: password + token + biometric + geography + user profiling Only the first three are in common usage, but we can expect user profiling to play a greater part in the future.

Password : something you know

Token : something you own (for example, smart cards)

Biometric : something you are (A unique and measurable

characteristic of a human being used to identity an individual)

Geography : the system or location you are using

User profiling involves developing a profile of the user concerned. The profile can be built by a history of events and actions. It therefore involves monitoring the user's behavior. Any behavior 'out-of-character' would trigger a silent alarm to an administrator who could investigate further. Thus, in access control, out-of-character behavior might require further authentication, or the intervention of a supervisor. User profiling also has a place in fraud prevention. If a credit card owner only ever uses it in one

particular store and neither buys a cigarettes nor alcohol, but then suddenly uses it in a different store ro purchase several hundreds of cigarettes and a dozen bottles of whisky, then the till operator would receive a silent warning to pay extra attention to this transaction. The problem is not in the use of such technology, but in the abuse of that technology. First of all, the assembly of a user profile is by definition an invasion of privacy. Secondly, we can guarantee that the owners of the profiles will treat them as a commercial asset that can be sold to other large marketing organizations.

- Access permissions

  It is a set of permissions associated with every file and directory that determines who can read it, write to it, or execute it. Only the owner of the file can change these permissions.
- Access Control List (ACL)

  An access control list (ACL) is a common mechanism used to specify Access Permissions. In its simplest form, it is a list for each managed object in a system. The list specifies which subjects can perform which types of access to which object. If the subject's ID is not included in the ACL, then that subject is not allowed to access the object.

3. Two kinds of Access Control

Access Control is two kinds of things. One is History- based access control, which approval is determined by the fact based on histories. The other is Role-based access control [3], which approval is determined by that a user is assign to any roles. In History-Based Access Control, there are Out-of-the-k policy, Keeping out Rogues policy, Frustrating Peepers Policy (Chinese Wall Policy) and Slowing down Hogs [6]. These approval is determined by that Subject has been access to object. The previous access has an effect on the next access. In case of Chinese Wall Policy, if there is a relation between two objects and a user have been access to one of two objects, then a user can not access to the other of two objects. In this way, since it has an effect on the the next access, it call the History-based access control.

3.1. **Military Access Control.** It is a mechanism that enforces the corporate policy or security rules that deal with the sharing of data. This is done by comparing the sensitivity of the resource (for example, file or storage device) with the clearance of the

entity (for example, user or application). That is, it is only controlled by level as order occurs in Military. Such rules could include " only members of the accounts department may read or change payroll data", and "classified data may only be accessed by staff with a 'classified' clearance level". For example, there are a colonel, a major, a soldier in Military. Suppose a colonel can read the document A which has the information of top secret and a major do B which has the information of secret, and a soldier do C which has the information of unclassified. Then a colonel can read A,B,C, and a major can read B,C, and a soldier can read C. As we see, fix a level of subject and object each. So since the level of a colonel is the same as the level of A, then a colonel can read access to A. Since the level of a soldier is different to the level of A, then a colonel cannot read access to A.

3.2. **Mandatory Access Control (MAC).** It is single level access control. Subject is linked to information ranked as unclassified, restricted, confidential, secret, top secret. Subjects has the principle of least privilege, that is, Need-to-Know principle. And that information is associated with compartments. For example, nuclear, cryptography, intel and so on. Here clearance is the maximum classification to which subject is permitted access. And Need-to-Know is the subject's need-to-know. That is, the total sum of all categories to which he is permitted access. So subject is denoted by (clearance, NTK). Therefore MAC is the thing that add category to Military AC.

3.3. **Bell-LaPadula model.** One of formal model of security policy that describes a set of access control rules is Bell-LaPadula model. By conforming to a set of rules, the model inductively proves that the system is secure. A subject's access to an object is allowed or disallowed by comparing the object's security classification with the subject's security clearance. The three basic rules are the *-property, the simple property, and the tranquility property.

- Object

  Objects are passive entities between which information flows under the direction of active entities called subjects. These could thus include directories/folders, files, fields, screens, keyboards, memory, magnetic storage, printers and so on. An object is effectively data or a data container. Access to an object implies access to the data contained by the object.

— Security Classification —

A security classification is an hierarchical sensitivity label applied to an object. It is used to determine which users may access what data, which generally based upon their own hierarchical security clearance level. In order for the owner of 'information' to implement proper security controls, administrator must first classify that information with one of a number of classifications. In the sector, Information is often classified into following. Although in reality each administrator can set its own classifications with its own levels of security for each classification :

$$Public < Sensitive < Private < Confidential$$

For example, sensitive data is information that requires a higher level of security that public data. The owner should protect it from loss of confidentiality as well as integrity from unauthorized individuals.

Within Government, the classification are normally:

$$Unclassified < restricted < confidential < Secret < Topsecret$$

- Subject

  In an automated information system, a subject is any active entity that causes information to flow among passive entities called objects. The subject could thus be a user or a process.

  — Security Clearance —

  Assuming that a system's objects are all given an hierarchical label defining their sensitivity (classification), a subject's security clearance is the corresponding label that defines the degree of sensitivity that can be accessed. Clearance level labels could, and for administrative ease possibly should, be given the same names as classification level labels. Under such circumstances, a subject with a clearance level up to 'secret' would be able to access objects with an classification level up to, but not higher than, 'secret'.

- Simple Property

  It is one of the three main properties of the Bell LaPadula security model. It states that a subject may only have read access to an object if the security level

of the subject dominates that of the object. We will denote it by S $\geq$ O. When we think about it, it's quite simple. A user may only read a file if he or she has a security level equal to or greater than that of the file. It means that someone with a 'secret' security level cannot read a file with a 'top secret' security level, but can read a file with a 'secret' or 'confidentiality' or 'restricted' security level.

- *-Property (star-property)
  It is called 'confinement property'. A subject is only allowed write access to an object if the security level of the object is greater than or equal to the clearance level of the subject. This makes it impossible for data from a highly cleared subject to become available to users with a lower security clearance in an object with a low security level. The purpose is to confine sensitive data at its correct level. Without this rule, Alice with a high security clearance could copy sensitive data into the document of Bob with low security clearance, thus allowing 'confidential' data to move from a 'top secret' to an 'unclassified' level.

3.4. **Role based Access Control.** This is a scenario. There is the subset of subjects A,B,C. Suppose that A can buy a vehicle. Then A must order the vehicle which A would like to have, and then pay the money of the vehicle, maybe A will sign the order for paying the money. Then, A must be certificate. However, this work isn't done by A, one of B,C must verify a sign. This work's purpose is that protects a transaction's from being worked by one user for personal profit. This simplified form is the rank based access control. In 1998, D.Gligor and Ferraiolo [3]pull Separation of Duty together. In this paper define formally a wide variety of separation-of-duty(SoD) properties and establish their relationships within a formal model of role-based access control(RBAC). The formalism helps remove all ambiguities of informal definition, and offers a wide choice of implementation strategies. Besides, we explore the composability of SoD properties and policies under a simple criterion. We conclude that practical implementation for SoD policies requires new methods and tools for security administration even within applications that already support RBAC, such as most database management systems. Separation of duty has had wide application in business, industry, and government. Its purpose is to ensure that failures of omission or commission within an organization are caused only by collusion among individuals, and that chances of collusion are minimized by assigning individuals of different skills or divergent interests to separate

tasks. For example, SoD is enacted whenever conflict of interest may otherwise arise in assignment of tasks within an organization. Let us consider Role based access control.

Then any subject will related to $(object, operation)$. That is, at least two user in the same group operates the object. If Jane would like to buy cosmetic, then there are 3 operation $(order, sign, verifysign)$. Then 3 user can execute each operation. Or Jane execute $(order, sign)$, another user can execute $(verifysign)$. That is, a transaction is consisted of 3 operations.

$$\sum_{i=1}^{3} f(Alice, object, op_i) = 0$$

$$if \sum_{i=1}^{3} H(Alice, object, op_i) = 2$$

3.5. **History based Access Control - Chinese Wall Policy.** The Chinese Wall Policy combines commercial discretion with legally enforceable mandatory controls. It is required in the operation of many financial services organizations and is perhaps as significant to the financial world as Bell-LaPadula's policies are to the military policy. It can be most easily visualized as the code of practice that must be followed by a market analyst working for a financial institution providing corporate business services. Such an analyst must uphold the confidentiality of information provided to him by his firm's clients. This means he cannot advise corporations where he has insider knowledge of the plans, status or standing of a competitor. But the analyst is free to advise corporations which are not in competition with each other, and also to draw on general market information. Unlike Bell-LaPudula policy, access to data is not constrained by attributes of the data but by what data the subject already holds access rights to. Essentially, datasets are grouped into " conflict of interest classes " and by mandatory ruling all subjects are allowed access to at most one dataset belonging to each such conflict of interest class. The actual choice of dataset is totally unrestrained provided that this mandatory rule is satisfied. As a result, this policy is not modelled by Bell-LaPudula policy. It is important to implement the Chinese Wall Policy. Especially, correct implementation is valuable to English Financial Institutions since it provides a legitimate defense against certain penal classes of offense under their law.

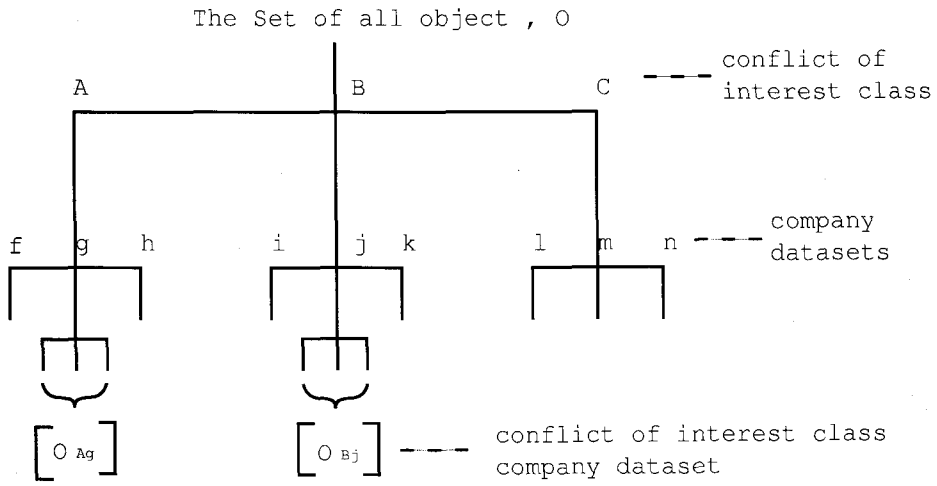

FIGURE 1. Database Organization: All corporate information is stored in a hierarchically arranged filing system such as that shown in figure.

The basis of the Chinese Wall policy is that people are only allowed access to information which is not held to conflict with any other information that they already possess; That a user already possessed the only information must be information that user has previously accessed. Thus a new user may freely choose to access whatever datasets he likes; we will concern a new user doesn't possess any information and therefore no conflict can exist. But sometime later, such a conflict may exist.
Suppose our user accesses the Oil company-A dataset LG first. we say that our user now possesses information concerning the Oil Company-A LG. Sometimes later he requests access to the Bank-A Wooribank. This is quite permissible since LG and Wooribank datasets belong to different conflict of interest classes and therefore no conflict exists. But if he requests access to the Oil Company-B SK the request must be denied since a conflict does exist between the requested dataset and one already possessed.
What we have just described is a Chinese Wall. We note that our user has complete freedom to access anything he cares to choose. Once that initial choice has been made, a Chinese Wall is created for that user around that dataset and we can think of "the wrong side of this Wall" as being any dataset within the same conflict of interest class as that dataset within the Wall. Nevertheless the user still has freedom to access any other dataset which is in a different conflict of interest class. However as soon as that

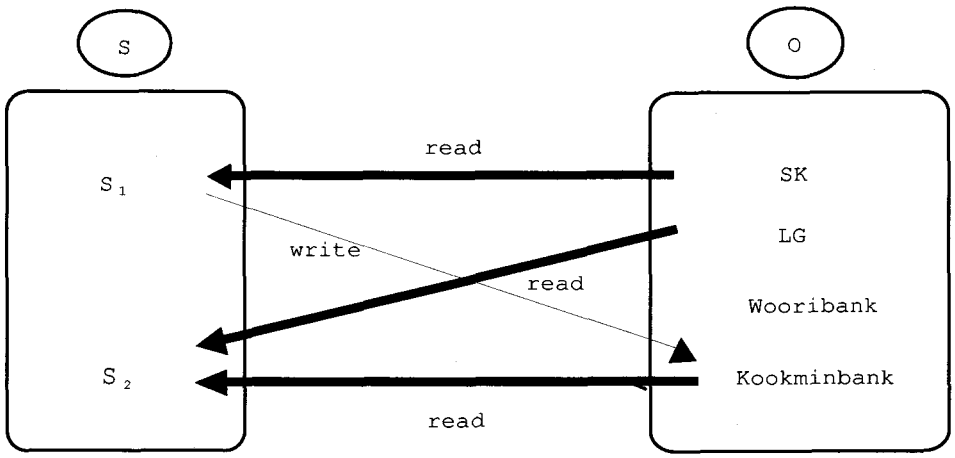

FIGURE 2. Example

choice is made, the Wall changes shape to include the new dataset. Thus we see that the Chinese Wall policy is a subtle combination of free choice and mandatory control.

Notation— Let S be a set of subjects, O be a set of objects and L be a set of security labels $(x_i y_j)$, where x_i represents the element of conflict of interest classes and y_j represents the element of company datasets at same conflict of interest class. Here x_i means the i-th conflict of interest classes and y_j means j-th company dataset. One such label is associated with each object. i.e. If O has the security label $(x_i y_j) \in L$, then O is contained to the conflict of interest class x_i the company dataset y_j. Hence $O_{x_i y_j}$ means that objects which are contained to conflict of interest class x_i company dataset y_j. Then $O_{x_i y_j}$ contain at least one objects.

- **Simple Security Rule**
  Read access is only granted if the object requested:
  a) is in the same company dataset as an object already accessed by that subject,
  or
  b) belongs to an entirely different conflict of interest class

- ***-Property Rule**

  Write access is only permitted if

  a) read access is permitted by the simple security rule, and

  b) no object can be read which is in a different company dataset to the one for which write access is requested.

- **Theorem based on above Rules**

  (1) Once a subject has accessed an object, the only other objects accessible by that subject lie within the same company dataset or within a different conflict of interest class.

  (2) A subject can at most have access to one company dataset in each conflict of interest class.

  (3) Assume that there exist some conflict of interest class and the number is ν, and there exist some company datasets in each conflict of interest class and the number of company datasets is ν_μ. Then the minimum number of subjects which will allow every company datasets to be accessed by at least one subject is ν_μ.

4. Formalization of Access Control

To formalize basic functions, we need to object, subject, and action. Here we have to consider object's relation, subject's grouping, and history list. History list is modelled by Non-access structure. If any subject are trying to any object, there will be given to 2 factors containing to basic factor, that is, subject's grouping, object's relation. Based on this 5 factor, we will design Non-access structure. Then history is laid on edges of Non-access structure. Therefore we will determine access requested as we see the history in Non-access structure. First of all, we will define the function associated with history access list which presents whether or not Subject had accessed to Object before. We will call it history function.

$$H : Subject \times Object \times act \rightarrow \{0, 1\}$$

- If S_i had accessed to o_α for executing, then $H(S_i, o_\alpha, act) = 1$.
- If S_i had not accessed to O_j for executing, then $H(S_i, o_\alpha, act) = 0$.

At the next time, we will define the function which presents whether or not Subject can have access to Object when Subject requests access to Object. We will call it enforcement function.

$$f : \mathit{Subject} \times \mathit{Object} \times \mathit{act} \rightarrow \{0, 1\}$$

- $f(S_i, o_\alpha, act) = 0$ means S_i cannot access to o_α for executing.
- $f(S_i, o_\alpha, act) = 1$ means S_i can access to o_α for executing.

What 0 means is the thing not to be able, and what 1 means is the thing to be able.

4.1. **Object-based Formalization.** Let us consider the object's relation. Let $o_\alpha \in [O_{x_i y_j}]$. Define $o_\beta \sim o_{j'}$ if $o_\beta \in [O_{x_i y_{j'}}]$, and tell that there are the same conflict of interested class. And so define $o_\alpha \nsim o_\beta$ if $o_\beta \in [O_{x_{i'} y_{j'}}]$, and tell that there are the different conflict of interested class. Note that $y_j \neq y_{j'}$ if $x_i \neq x_{i'}$. Here "$\tilde{}$" is only reflexive, but not symmetric and transitive. Suppose that two objects o_α, o_β are given. If $o_\alpha \in [O_{x_i y_j}]$, $o_\beta \in [O_{x_i y_{j'}}]$, then these objects have the relation. Else $o_\alpha \in [O_{x_i y_j}]$, $o_\beta \in [O_{x_{i'} y_{j'}}]$, then these objects don't have the relation. Assume $o_\alpha \in [O_{x_i y_j}]$.

(1) $f(S_i, o_\alpha, read) = 0$ if $H(S_i, o_\beta, read) = 1$ for some $o_\beta \in [O_{x_i y_{j'}}]$
(2) $f(S_i, o_\alpha, read) = 1$ even though $H(S_i, o_\beta, read) = 1$ for $o_\beta \in [O_{x_i y_j}]$, or even though $H(S_i, o_\beta, read) = 1$ for $o_\beta \in [O_{x_{i'} y_{j'}}]$
(3) $f(S_i, o_\alpha, write) = 0$ if $H(S_i, o_\beta, read) = 1$ for some $o_\beta \in [O_{x_i y_{j'}}]$.
(4) $f(S_i, o_\alpha, write) = 1$ if $H(S_{i'}, o_\beta, read) = 0$ for $\forall o_\beta \in [O_{x_i y_{j'}}]$ or $\in [O_{x_i y_j}]$.

These indicates access rule by simple property and *-property.

Now let us try to explain the Chinese Wall Policy applying to the scenario about the hospital. Consider the situation where administrator want to allow people to access only one of two relations in a database but not both. People might wish to know that a patient is attacked with which disease if accessing both the relations.i.e. Administrator might wish to allow people to access either information contains the date and the name of medical procedures performed in a hospital or information that contains the names of patients and the date they last came in. Individually, these relations don't allow people to deduce information about illness of individual patients. Let 3 subjects be denoted by $S_{peo}, S_{doc}, S_{hst}$ and 3 objects be denoted by $O_{pro}, O_{name}, O_{inout}$ are given. And there is the relation between O_{pro} and O_{name} (written $O_{pres} \sim O_{name}$) and there is

no relation between O_{pro}, O_{inout} and O_{inout} (written $O_{pro} \nsim O_{inout}$ and $O_{pro} \nsim O_{inout}$). Note that doctor knows the illness of patient. In this scenario, the doctor can read the prescription and name about his patient. Here the thing that S_i executes O_j ,where i $\in \{doc, peo, hst\}$ and j $\in \{pro, name, inout\}$, will consider {read, write}. As we saw, history function and enforcement function are returned to $0, 1$. Here what 0 means is the thing not to be able, and what 1 means is the thing to be able.

Assume that $rank(O_{pro}) = rank(O_{name}) = rank(O_{inout})$ and $rank(S_{hst}) = rank(S_{peo}) = rank(S_{doc})$, where rank is defined to security level of the Subject and Object. There was, in practice, a need for "trusted subjects". Let S_i,where i $\in \{hst, peo, doc\}$ and O_j,where j $\in \{pro, name, inout\}$.

- $f(S_i, O_j, read) = 0$ if $H(S_i, O_{j'}, read) = 1$ for some $O_{j'} \sim O_j$. But in this scenario, $f(S_{doc}, O_{pro}, read) = 1$ in spite of $H(S_{doc}, O_{name}, read) = 1$.
- $f(S_i, O_j, write) = 0$ if $H(S_{i'}, O_j, read) = 1$ for some $S_{i'} \neq S_i$ and $H(S_{i'}, O_{j'}, read) = 1$ for some $S_{i'} \neq S_i$ and for $\forall\, O_{j'} \sim O_j$. But in this scenario, $f(S_{doc}, O_{pro}, write) = 1$ in spite of $H(S_{doc}, O_{name}, read) = 1$.
- $f(S_i, O_j, write) = 1$ if $H(S_i, O_j, read) = 1$ and $H(S_{i'}, O_{j'}, read) = 0$ and $\forall S_{i'} \neq S_i$.

If any subject try to access to object for reading, then this subject had accessed to any objects of same conflict of interested class, depending on this subject's accessability of any objects of different interested class. And if any subject try to access to object for writing, then that object had not required for reading and writing by any other subjects. Instead, that subject need not to have accessed to that object for reading. Finally,

$$In\ order\ to\ f(S_i, o, read) = 1\ for\ o \in [O_{x_i y_j}]$$

$$H(S_i, o', read) = 0\ for\ o' \in [O_{x_i y_{j'}}]$$

$$In\ order\ to\ f(S_i, o, write) = 1\ for\ o \in [O_{x_i y_j}]$$

$$H(S_{i'}, o', read) = 0\ for\ o \in [O_{x_i y_j}]$$

$$for\ o' \in [O_{x_i y_{j'}}]\ and\ S_i \neq S_{i'}$$

4.2. **Rank-based Formalization.** Consider that $rank(S_i) \neq rank(O_j)$. We will regard actions as read, write access. Of course, there are many actions. e.g. append, modify and so on. We require that if subject has write access to some objects and read access to some objects, then the classifications objects to which subject has write access must exceed or equal the classifications of the objects to which subject has read access. Put more simply, a subject can have "read" access to one object. He would like to have "write" access to another only if the classification level of the second object was greater than or equal to that of the first. If the simple security property's purpose is no-read-up, then *-property was no-write-down. If systems were designed in above way as to ensure subjects could not write down, then Trojan horses could be defeated. Assume that $H(S_i, O_j, read) = 1$. If $rank(O_j) < rank(O_{j'})$, then $f(S_i, O_{j'}, write) = 1$ but $f(S_i, O_{j'}, read) = 0$. i.e. this policy is not allowed to write-down. Else if $rank(O_j) > rank(O_{j'})$, then $f(S_i, O_j, read) = 1$ but $f(S_i, O_{j'}, write) = 0$. Here "write" means 'addition' of the information. But From now on, we will regards "read" as itself and regards "write" as itself. An example of no-write-down is the following statement. A subject with read access to confidential objects has write access to confidential, secret, and top-secret objects, but does not have write access to unclassified objects. Finally,

$$Assume\ that\ rank(S) \neq rank(O).$$

$$If\ rank(S) \leq rank(O), f(S, O, write) = 1$$

$$If\ rank(S) \geq rank(O), f(S, O, read) = 1$$

$$Assume\ that\ rank(S) = rank(O).$$

$$Then f(S, O, read\ and\ write) = 1$$

4.3. **Subject-based Formalization.** Subject may be user, or program, or process. So, we will be try to consider subject's grouping. As we discussed, we had considered the relation between objects. Since subject is the same as object, we will consider grouping between subjects. Then this group is the set of subjects, which is related to each other. If that happens, we will denoted by $[S_i]$, this means the gathering to which S_i are related. That is, subjects in $[S_i]$ have the relation mutually. Therefore it is not important that $[S_i]$ is marked by any subject in that representative. Then we

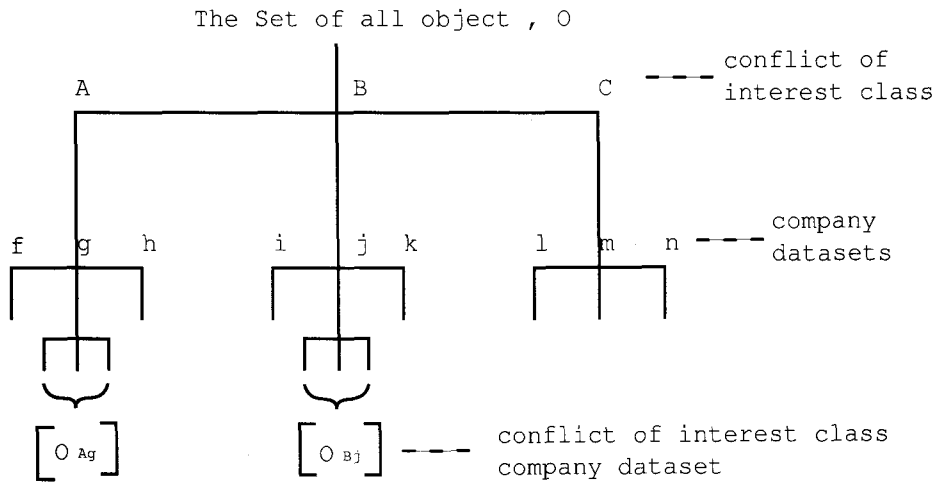

FIGURE 3. Subject Grouping

will call S_i the representative of subjects. And then we will denote subject's grouping by $[S_i]$, so this grouping $[S_i]$ is considered as one subject. This is similar to the notion of coset of abelian group. Here "$\tilde{}$" is reflexive, symmetric, and transitive. That is the equivalent relation. For example, assume that there exist S_1, S_2, S_3. We think that S_1, S_3 are being group and S_3 is only one grouped. Then $S_1 \nsim S_3$ and $S_1 \sim S_2$ and $S_3 \nsim S_2$, so $S_1, S_3 \in [S_1]$, $S_3 \in [S_3]$. Assume that any subject S_4 is given, this is related to S_1, S_3. Then $S_4 \in [S_1]$ and $S_4 \in [S_3]$. Since this relation is transitive, S_1, S_2, S_3, S_4 have the same representative. Namely, $S_1, S_2, S_3, S_4 \in [S_1]$. Well then, we can ask how subjects are grouped. Imagine that there is a mathematical department in the university. Then there will exist many people. These belong to any group, for instance, the first master's course, the second master's course, any professor' laboratory, or the staff of a mathematical department etc. Then these are already grouped. Or, subjects are friends each other. Then we will state some properties as regarding the grouping of subjects. Finally,

$$f(S_i, o_j, read) = 1 \ for \ \in [O_{x_i y_j}]$$

$$if \ H(S_{i'}, o_{j'}, read) = 0$$

$$for \ S_i \sim S_{i'} \ and \ o_j \in [O_{x_{i'} y_{j'}}]$$

$$f(S_i, o_j, write) = 1 \ \in [O_{x_i y_j}]$$

$$if\ H(S_{i'}, o_{j'}, read) = 0$$

$$for\ S_{i'} \nsim S_i\ and\ o_{j'} \in [O_{x_i y_{j'}}]$$

5. Generalization

Suppose that a subject are trying to act to a object. Then we will have to consider 3 things, subject and object' rank, subject's grouping, object's relation. At the first time, we will the rank of Subject and object. Easily, if the rank of subject is higher than that of object, then subject can read the object. In the opposite direction, subject can write the object. Therefore if the rank of subject is the same as that of object, then subject can read and write the object.

$$If\ rank(S) \geq rank(O),\ then\ f(S, O, read) = 1$$

$$If\ rank(S) \leq rank(O),\ then\ f(S, O, write) = 1$$

$$If\ rank(S) = rank(O),\ then\ f(S, O, read\ and\ write) = 1$$

As we see, we will regard action as 2 kinds of form, read and write. Practically, action is considered by many kinds of form. A subject can modify the object, and can append to the object and so on. But we will use the simple form. And we will think action as read or write. At the second time, we will the subject grouping. Subject's grouping is 2 kinds of form, friend or enemy. And if a subject would like to read the object, we will examine another subject's access history list such that another subject is friend of a subject. So, if another subject had ever read the object, then a subject can read the object. If S_1, S_2 are friend each other, they belong to same grouping $[S]$. Well then, the enemy is the member which belongs to different grouping $[S']$, that is, these subjects doesn't belong to grouping having to a subject. Therefore once a friend had ever read the object, certainly we regards a subject as reading the object, then enemy doesn't matter with a subject's history. Finally,

$$If\ S\ group\ S',\ H(S, O, read) = 1,\ then\ f(S, O, read) = 1$$

$$If\ S\ group\ S',\ H(S, O, write) = 1,\ then\ f(S, O, write) = 1$$

At the final time, we will think the object case. Object's relation is that we give the connection between objects having the probability leaking the partial information. If a subject had read object A, then a subject cannot read object B since information

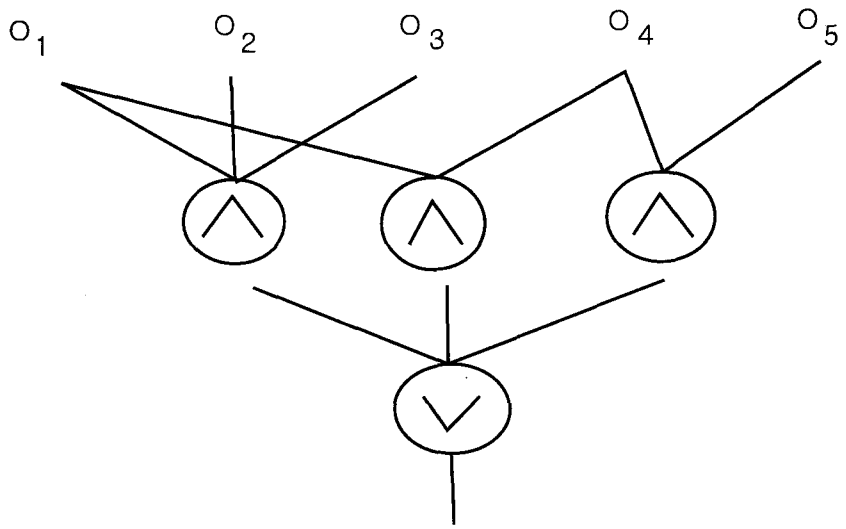

FIGURE 4. Non-access structure considering object's relation

is leaked unconsciously. Therefore we will denote by $O_1 \sim O_2$. For example, Suppose Alice can read {A,B,C} when we compare only the rank of Alice with the rank of document {A,B,C}. Here "There are objects in the same set" means that they may not read at the same time. That is, Alice can read Documents A,B,C simultaneously. But we must consider power set of objects.

$$\{A\}, \{B\}, \{C\}, \{A, B\}, \{A, C\}, \{B, C\}, \{A, B, C\}$$

In fact, since there is the notion of time, it is not easy to express access history. Then we will fix the history function. Therefore as we see figure 5.1, we make Non-access control. Next to make such model, we write the history list to edges connected to objects. That is, history function returns subject's representative. And according to operation of structure, set which is returned is representative.

If subject S_i had access to object O_j, then $H(O_j, read) = [S_i]$, where $[S_i]$ is representative.

This flexible history function will be applied to the model. Now while we consider object's relation, we can consider subject's relation. Then we can conclude the following.

Assume that S_i group S_i', S_i''.

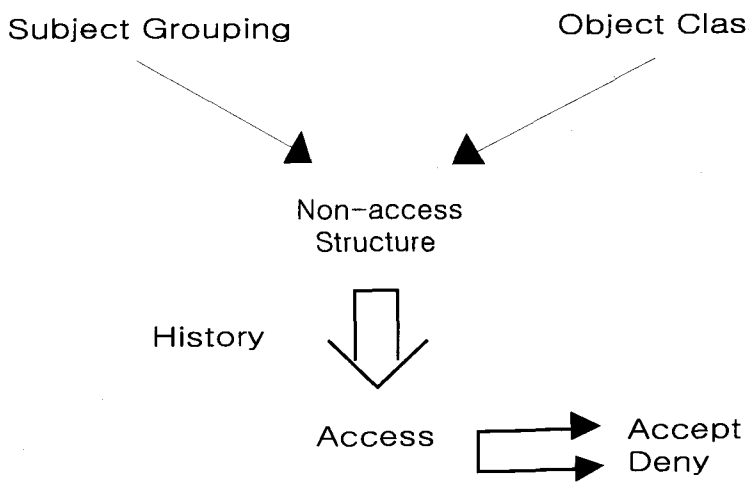

FIGURE 5. Blueprint

$$And\ assume\ that\ O_j \sim O'_j, O''_j.$$

$$f(S_i, O_j, read) = 1$$

$$if\ \cup(H(O'_j, read) \cap H(O_j, read)) = \emptyset\ for\ all\ O'_j\ O_j$$

$$f(S_i, O_j, write) = 1$$

$$if\ \cup(H(O_j, read)) = [S_i]\ and\ \cup(H(O'_j, read)) = \emptyset\ for\ all\ O'_j\ O_j$$

6. CONCLUSION

To block the possibility of a hostile site being able to deduce the same information from data provided by two different programs it provides, programs that have opened a socket are not allowed to access sensitive relations and programs that have accessed one of the sensitive relations are not allowed to open sockets. Another example, any user can read the date and the names of medical procedures performed in a hospital, the same user can read the names of patients and the date they last came in. But if this occurs, then user knows treatment histories of individual patients. If any patient would not like to be known to another people except the attending doctor, then subject and object's rank, subject's grouping, and object's relation are meaningful. Therefore if the subject had access to one of related objects, then the subject cannot access to anything which is related to that object. As we saw figure 5.1, we will design Non-access structure. There

edges between objects and "and" mean the history list containing which subjects had read that object. If $\{O_1, O_2, O_3\}$, $\{O_1, O_4$, and $\{O_4, O_5$ are each read simultaneously, O_1 edges have the history list S_1, S_2. Assume that $S_1, S_3 \in [S]$, $S_2 \in [S']$. If S_3 are trying to O_4, then $f(S_3, O_4, read) = 0$ because of $H(S_1, O_1, read) = 1$. Then we can express this thing formally.

$$Assume\ that\ rank(S_i) = rankO_j$$

$$Assume\ that\ S_i, S_i' \in [S_i], S_i'' \in [S_i''].$$

$$And\ assume\ that\ O_j \sim O_j',\ O_j''.$$

$$f(S_i, O_j, read) = 1$$

$$if\ \cup (H(O_j, read) \cap H(O_j', read)) = \emptyset\ for\ all\ O_j'$$

$$f(S_i, O_j, write) = 1$$

$$if\ \cup (H(O_j, read)) = [S_i]\ and\ \cup (H(O_j', read)) = \emptyset\ for\ all\ O_j'$$

Therefore system's administrator can verify that system is secure when several policies are operated by system.

REFERENCES

[1] F.C.Brewer, J.Nash "The Chinese Wall Security Policy" (1989)

[2] D.E.Bell and L.J.LaPadula "Secure Computer Systems: Unified Exposition and Multics Interpretation" (1976)

[3] Virgil D.Gligor, Serban I.Gavrila and David Ferraiolo "On the Formal Definition of Separation-of-Duty Policies and their Composition IEEE Symposium on Security and Privacy (1998)

[4] Vincent C.Hu, Devorah A.Frincke, and David F.Ferraiolo "The Policy Machine For Security Polcy Management" NIST (1999)

[5] Ravi S.Sandhu "Lattice-Based Access Control Models" IEEE Computer (1993)

[6] Guy Edjlali, Anurag Acharya, Vipin Chaudhary "History-based Access Control for Mobile Code" (1998)

[7] Douglas R.Stinson "Cryptography"

[8] Harold F.Tipton and Micki Krause "Information Security Management", 4th Edition (2000)

[9] David Ferraiolo and John Barkley "Specifying and Managing RBAC within a Corporate Intranet" (1997)

[10] Ravi S.Sandhu "On Five Definition of Data Integrity" (1993)

[11] Vincent C.Hu, Deborah A.Frincke, and David F.Ferraiolo "The Policy Machine For Security Policy Management" (2001)

[12] Bell D.E. and Lapadula L.J. "Secure Computer systems: Mathematical Foundations and Model" (1973)

[13] Sandhu R.S. "Role-Based Access Control Models" (1996)

[14] Sandhu R.S. "Lattice-Based Access Control Models" (1993)

[15] Simon R.T. and Zurko M.E. "Separation of Duty in Role-Based Environments" (1997)

[16] Carlos Ribeiro, Andre Zuquete, Paulo Ferreira, and Paulo Guedes "Security Policy Consistency" (2000)

Mathematics, Korea Advanced Institution of Science and Technology
DaeJon, Korea
e-mail: wild1018@knot.kaist.ac.kr