Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Granular Computing and Flow Analysis on Discretionary Access Control:
Solving the Propagation Problem

Tsau Young (`T. Y.') Lin and Jene Pan
Department of Computer Science
San Jose State University
San Jose, CA 95192-0249
tylin@cs.sjsu.edu

Abstract. Based on granular computing, information flows in Discretionary Access Control (DAC) are examined. DAC are classified in the following nested order: From general to specific, binary neighborhood systems(binary relations),topological spaces (reflexive and transitive relations) and clopen spaces (equivalence relations) in geometric (algebraic) terms. In security terms, the two smaller classes meet information flow security and Chinese wall security policy in respective order. Roughly, information flow security policy (IFSP) means any data can never flow or propagate into the enemy hands of the initial owner. Chinese wall security policy is IFSP, in which enemy is a symmetric relation.

Keywords*—component, formatting, style, styling, insert *(key words)

I. INTRODUCTION

It has been known for a long time that information flow on Discretionary Access Control (DAC) is a difficult problem; e.g. [41]. For example, in UNIX system, the owner of data is the one who may authorize the access of the data. So if Peter's data D flows through Larry to John, then we have a dilemma: D has three owners. The most undesirable situation for Peter is that the new owner may authorize Peter's enemy to read the data D. In this paper, we will solve this type of the problem. However, our solution is not direct in the sense that we are not providing a procedure to convert an insecure system to secure one. What we have done is to determine the mathematical structures of DAC. From the understanding of DAC, we can choose appropriate DAC so that the dilemma will not occur.

Our approach is based on an emerging field, called Granular Computing (GrC). From GrC's view, we have found that DAC have the following structures: From general to specific in a nested order, (1) geometrically, binary neighborhood systems, topological space, and clopen spaces. (2) Algebraically, binary relations, reflexive and transitive relations, and equivalence relations. Based on such understanding, we can show that a DAC is secure in (1) information flow security policy (IFSP), if DAC defines a topological space (or equivalently a reflexive and transitive relation), or (2) Chinese wall security policy (CWSP), if DAC defines a clopen space (or algebraically an equivalence relation). Roughly, IFSP means any data can never flow or propagate into enemy hands of the initial owner. CWSP means any data can never flow or propagate between those who are mutually in conflict. Mathematically, if the enemy relation in IFSP is symmetric, then IFSP = CWSP. Chinese wall security policy have been reported in 2003 [22]. So this paper is focused more on the information flow security.

II. PROPAGATION PROBLEMS

To understand the information flows, especially, in commercial security has been a difficult problem. Here is a summary of the essence of the **problem** from a file in the internet [42]. We reformulate it as follows (only the quoted statements are the original)

1. "Most people familiar with DAC"

1. "Example: Unix user-group-other permission bits.
2. "Might set a file private so only group friends can read it"

We have revised the next bullet to:

2. "Discretionary means anyone with access can propagate information" based on DAC. Note that the data sent

by e-mails: Mail sigint@enemy.gov < private

is assumed to have been regulated by DAC (as well as MAC). To see the detail problem, we need the notion of security.

A. Information Scurity

A common definition of a secure system is: The following quote is from [9]: "A system is secure if it is adequately protect information that it processes against unauthorized disclosure, modification and deny of service." We will focus on the first two items. It was paraphrased in [43] as follows:

3. A system is secure if any operation, such as disclosing or sending information, on the system is authorized.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

This rather trivial assertion was used to show that MAC with trusted subjects is secure. In DAC, the situation is quite different. For example. in UNIX system, the owner of data is the one who may authorize the access of the data. To discuss the details, we need to set up some basic notions and notations.

B. DAC and UNIX

UNIX's user-group-other permission bits allow group friend to access user's data. For simplicity, we will assume all data of a user has the same permission bits. So we will use, by abuse of language, user's name to refer to him or his data. We denote this group friend of the user • by F-list (•); this list also has been called access list. In UNIX, by default, the compliment is the enemy list, which means no one in the list can access owner's data. This list is denoted by E-list (•).

However, in UNIX, the E-list(•) has been treated quite casually. But in MAC, the E-list is much more seriously treated; it is called explicitly denied list that take precedence over any permissions. In Chinese wall security policy, the enemy list (conflict of interest class) is also the primary concern. In this paper, we will adopt the later attitude.

1) E-list(•) is a seriously considered list
2) E'-list(•) is the compliment of E-list(•)
3) F-list(•) is the usual friend list of F-list

Obviously, F-list is a subset of E'-list and hence is disjoint from E-list.

C. Information Propagation on DAC

Now, let us back to the discussion of security. Let Peter be the owner of a piece of data D. Let Larry be in F-list(Peter) and John in F-list(Larry). Suppose Peter's data D has been flowed into Larry, then to John. Note that once Larry and John have accessed the data D, each becomes a new owner. In other words, D has three owners. Then

4. what will be the initial owner Peter's concern ?

His concern is new owners may authorize someone in
1. F-list(New owner)∩E'-list(Peter), or
2. F-list(New owner)∩E-list(Peter)
to access data D.

Case 2 is not acceptable, from Peter's view. This is the propagation problem, more generally

Propagation problem may occur**,** if F-list(•) ∩ E-list(**Peter**) is not empty for some user •, such as Larry or John, who is a direct or indirect friend of Peter. We will use T(Peter) to denote this group of direct or indirect friends or their data; it will be defined formally in later section and called the trajectory of Peter. So previous conditions "F-list(•) ∩ E-list(**Peter**)" is reduced to "T(**Peter**) ∩ E-list(**Peter**)".

5. Propagation problem exits, if T(•) ∩ E-list(•) is not empty

To simplify further discussions, following UNIX's practices, we assume

6. Convention: F-list(•) = E'-list(•)

III. GRC AND REFROMULATION OF DAC

In 1989 Brewer and Nash (abb, BN) proposed a very interesting security policy, called Chinese wall security policy [2]. The paper essentially addresses some type of information flow problem on DAC. To maintain the continuity, we will adopt its notations and language here.

A user in UNIX is the owner of an account. Without the lost of generality, we have assumed that the files in an account have the same permission bits and refer these files as the dataset of the account, owned by a company. BN referred the dataset of a company an object. So the UNIX-permission-bits essentially specify a group of users who can access the information of an object, say X. Here the access means to read and may be to save the information of X to the datasets of this group. We will abstract such an access by the concept of information flow. Namely, when such an access occurs, we say information in object X has flowed into an object Y ; here Y is a generic object in this group.

We have taken pains to abstract the UNIX files into objects. However, the notion of objects shall **not** be restricted to UNIX. Abstractly, an object consists of information (dataset) and its container (the account). So information in a DAC can flow from an object and be received by objects. Following BN, the collection of all such objects will be denoted by O. Therefore, the permission bits are abstracted to the following:

5. To each object, say $X \in O$, we associate a group *E-list*(X) = $\{Y_j. j = 1, 2,\ldots\}$ of objects who may not receive information from X. In other words, information in X may not flow into any object Y_j, $j = 1, 2,\ldots$. The set *E-list*(X) is, in MAC, referred to as explicitly denied list. We simply call it the enemy-list of X. In Chinese wall security model; this is called conflict class.

6. By default, the friend list of X, denoted by, *F-list*(X), consists of a group of objects, to which information of X can be flowed. This list has been referred to as access list; we will call it friend list [23].

Definition 1 *E-DAC model is a map*

$$E : O \to 2^O : \ X \to E(X).$$

that associates each object an enemy-list, where 2^O denotes the collection of all subsets of O, namely, the power set of O. Observe that in this paper, we denote DAC by E-DAC. Here E is used to emphasize that the information is NOT flowed into enemy's datasets. Such a list has been called explicitly denied list [23].

Definition 2 *F-DAC is a map*

$$F : O \to 2^O : \ X \to F(X).$$

that associates each object a friend-list. The F in F-DAC is used to emphasize that the information is flowed into friends' datasets

Abstractly, E(X) and F(X) concern the "same" concept.

A. Granular Computing on DAC

The concept of *F-DAC* and *E*-DAC can be viewed geometrically: Let *U* be a set and p be a point in U. Here, U is a geometric view of *O*, and we shall consider the geometric abstraction of F-DAC or E-DAC. Namely,

Definition 3 Geometric abstraction of E-DAC (or F-DAC) is a map

$$B : U \to 2^U : \ p \to B(p)$$

that associates to each point p a subset B(p), which may be empty. This map is called a binary neighborhood system (BNS) or a binary granulation (BG); by abuse of language, we will also call the collection $\{B(p) \mid p \in U\}$ a BNS. B(p), called a neighborhood, is geometric abstraction of E(X) or F(X).

The notion of BNS is equivalent to that of binary relation(BR).

Proposition 1. BNS induces and is induced by a BR.

Let B be a BNS, then we define the binary relation (recall that a binary relation is a subset of the Cartesian Product $U \times U$)

$$B_R = \{(p, u) \mid u \in B(p) \text{ for all } p \in U\}$$

Conversely, we define (right) binary neighborhood by

$$B(p) = \{u \mid (p. u) \in B_R\}.$$

Definition 4 The binary relation induced by F is called Direct Information Flow Model (DIF), and denoted F again or in some occasion by more graphically notation $\Rightarrow$. Similarly, the binary relation induced by E is called No Direct Flow Model (NDIF) and denoted by E again or more graphically $\not\Rightarrow$.

Pawlak introduced (1982) the concepts of rough sets based on the upper and lower approximations. Mathematically, these concepts are actually quite standard in the theory of topological spaces. They have been extended to neighborhood systems and granular computing [11],[12], [13], [17],[18].

Definition 5 For any subset S of U, we define

1. Derived set: A point p is a limiting point of S, if every B(q), such that $p \in B(q)$, contains a point of S other than p. The set of all such limiting points of S is called the derived set D[S]. The limiting point is also called accumulation point.

2. Closure: $C[S] = S \cup D[S]$; note that, in general, such C[S] is may not be closed, unless U is a topological space. Pawkak's upper approximation is a special case of this concept.

3. Closed Closure: We may define the closed closure as follows: S together with its transfinite derived set, which is derived transfinitely many times, that is, $CC[S] = S \cup D[S] \cup D[D[S]] \cup D[D[D[S]]] \ldots$ (transfinite). Such a closure is closed, even in neighborhood system spaces.

4. Interior: $I[S] = \{p \mid B(p) \subseteq S\}$. Pawkak's lower approximation is a special case of this concept.

The Closed Closure of a point has special meaning in DAC. It is related to the trajectories (orbits) of information flows.

Definition 6 Let O be the set of all objects and X an object. Then

$$T(X) \equiv CC[\{X\}] = \{X\} \cup D[\{X\}] \cup D[D[\{X\}]] \cup D[D[D[\{X\}]]]. \ldots \text{ (transfinitely many)}$$

is called the trajectory of information flows from X.

IV. INFORMATION FLOW SECURITY POLICY

The nature of the information flow is "continuous flowing." So we need to trace its trajectories, namely, we have to apply $\Rightarrow$ repeatedly to the objects. So we define

Definition 7 *Information Flow from X to Y is defined to be the compositions of finite sequences of $\Rightarrow$ (Direct information flow):*

$$X = \{X_0 \Rightarrow X_1 \Rightarrow X_2 \ldots \Rightarrow Xn\} = Y$$

Here n varies through some integers. Note that this includes the case, $X \Rightarrow X$. The collection of such (X, Y) is a reflexive binary relation, denoted by C and called Information Flow Model.

Corollary 1 C is the transitive closure of D.

The following corollary is immediate from the definitions.

Corollary 2 $T(X) = \{Y \mid (X, Y) \in C\}$

Intuitively, T(X) consists of all possible points that the information flows are allowed to reach.

MAIN THEME: The central theme of this paper is to discuss: How DAC can be designed properly so that Information flows can never flow into enemies' hands. Formally, we say

Definition 8 *The requirement that information flows will never flow into enemy list is called information flow security policy (IFSP). In other words,*

$$T(X) \cap E(X) = \varnothing, \ \forall \ X \in O,$$

where T is the trajectory and E is the E-DAC.

An object X is said to be secure, if $T(X) \cap E(X) = \varnothing$.

A. Illustrations

Let us illustrate the concepts by examples:

Example 1 Assume we have five objects A, B, C, D, E and their E-DAC is given below:

$A \to_E \{B, D, E\}$

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

$C \to_E \{B, D, E\}$

$D \to_E \{A, C, E\}$

$E \to_E \{A, B, C, D\}$

By default, we assume the complement of each friend-list is an enemy-list. So F-DAC is:

(1) $A \to \{A, C\}$

(2) $B \to \{B, D\}$

(3) $C \to \{A, C\}$

(4) D → {B, D}

(5) E → {E}

In this example, the F-DAC is secure in the sense of no information of an object X may flow into enemies' objects. Let us look at the trajectory of each object

From (1), T(A) = {A, C} , then we examine (1) and (3), we should have added the items in the right-handed side of the arrows into T(A). However, in this case, there is no change in T(A)

Observe that T(A) does not meet E(A) = {B, D, E}, so A is secure object.

Similarly from (2) and (4), we get T(B) = {B, D}. Observe that T(B) = {B, D} does not meet E(B) = {A, C, E}, so B is secure object.

From (3), (1); (4), (2), we can conclude C and D are secure respectively.

From (5), we get T(E) = {E}; it does not meet E(A) = {A, B, C, D}, so E is secure.

So Example 1 meets IFSP.

Next let us look at a negative example

Example 2 The enemy lists are:

$A \rightarrow_E \{B, D, E\}$

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

$C \rightarrow_E \{B, D, E\}$

$D \rightarrow_E \{A, C, E\}$

$E \rightarrow_E \{C, D\}$

By default, the friend-lists are:

(1) A → {A, C}

(2) B → {B, D}

(3) C → {A, C}

(4) D → {B, D}

(5) E → {E, A, B}

From (1), (3), we observe that the trajectories are T(A) = T(C) = {A, C}, which does not meet E(A)= E(C)={B, D, E}. So A and C are secure

From (2) and (4), we have T(B)=T(D)={B, D}, which does not meet E(B)=E(D)={A, C, E}. So B and D are secure.

From (5), we get T(E) = {A, B, C, D, E}. Next, we consider the arrows that start from the objects in T(E), namely, (1), (2), (3), (4), and (5). However T(E) does not change from this new consideration. Now, we observe that T(E) does meet E(E) = {C, D}. So we conclude E is not secure.

So we conclude that Example 2 does not meet IFSP, so are not secure.

B. Main Theorems

Let us recall some definitions

1. A symmetric binary relation B is a binary relation such that for every $(u, v) \in B$ implies $(v, u) \in B$.

2. $B' = V \times V \sim B$, which is the complement set of B, defines the complement binary relation (CBR).

3. A binary relation B is *anti-reflexive* if B is non-empty and no pair (v, v) is in B. Observed that B is anti-reflexive iff B' is reflexive.

4. A binary relation B is *anti-transitive* if B is non-empty and if (u, v) belongs to B implies that for all w either (u, w), (w, v) or both belongs to B. Observed that B is anti-transitive iff B' is transitive.

Let us state the main theorem, which can be viewed as a generalization of Chinese wall security **theorem [22].**

Theorem 1. Information Flow Security Theorem.
E-DAC enforces IFSP if E-DAC is anti-transitive. Equivalently E-DAC enforces IFSP if F-DAC is transitive.

Proof: Observe that since F-DAC is transitive the trajectories stay in F(X), which is the compliment of E(X) by Convention. so T(X) is disjoint from E(X), hence it satisfies IFSP.

Computational Issues

Mathematically, the propagation problem has been solved. However, the complexity issues may arise, when one check the transitivity of F-DAC. Here we will illustrate the problem and solution by simple examples; see Section VI

V. CHINESE WALLOW SECURITY POLICY

Next we re-examine the Chinese Wall Security Theorem in terms of IFSP. We quote 2 important statements from BN [2]:

1. "The Chinese wall security policy combines commercial discretion with legally enforceable mandatory controls …, perhaps, as significant to the financial world as Bell-LaPadula's policies are to the military."

2. "People are only allowed access to information which is not held to conflict with any other information that they already possess."

Proposition 1. Simple Chinese wall security policy implies that F-DAC is an equivalence relation:

1. We observe that if two datasets are accessible by the same agent, we should conclude that the information of two datasets can be flowed into each others

2. From the second assertion of BN, we conclude that an agent can access any information that is not in conflict with the information they already possess. So in $F(X)$, which is outside of $E(X)$, all information can flow into each other. Hence F-DAC is an equivalence relation.

Definition 9

1. Simple Chinese wall security policy (SCWSP) means F-DAC is an equivalence relation.

2. Aggressive Chinese wall security policy (ACSWP) means C is an equivalence.

Theorem 2 Chinese Wall Security Theorem.
SCWSP implies ACSWP.

Proof: Note that C is transitive closure of F-DAC, and observe that C is an equivalence relation iff F-DAC is.

Corollary 3 SCWSP implies ISFP.

This is immediate: since equivalence relation is transitive.

VI. Granular Computing on DAC

In this section, we will fix some errors in [22]. Let $R \subseteq V \times U$ be a binary relation. We will re-express R by BG or BNS [17].

1. A binary relation defines a mapping B, called binary granulation

$B : V \to 2^U : p \to B(p)$, where $B(p) = \{u \mid (p, u) \in R\}$

2. Conversely, given B, we can define the binary relation R:

$R = \{(p, u) \mid u \in B_p \forall p \in V\}$

Next let us recall from [17, 18], the following observation

Definition 10 *A binary granulation, as a mapping, induces naturally a partition by the inverse of a map:*

$\{B^{-1}(w) \mid w \in 2^U\}$*, called inverse image of w under B.*

We call it the induced partition of B, and denoted by E_B. The equivalence class, $[p]E_B = B^{-1}(B_p)$, or simply write as [p], is called the center or core of B_p.

Proposition 2
1. The center [*p*] *consists of all those points that have the same binary neighborhood* B_p *("same" in the sense of set theory).*

2. If B is symmetric, anti-reflexive and anti-transitive, then the complement binary relation B' is an equivalence relation.

3. If $B \subseteq V \times V$ *is a symmetric binary relation, and* E_B *is its induced equivalence relation, then each B-binary neighborhood is a union of* E_B*-equivalence classes.*

4. If B is symmetric, anti-reflexive and anti-transitive, then B' is the induced equivalence relation E_B.

Proof: The first and second assertions are obvious. We consider the third. Let B_p be the binary neighborhood of p. Let $x \in B_p$, and assume x and y are E_B-equivalence. By definition of equivalence in E_B, they have the same neighborhood, $B_x = B_y$. By the symmetry of B, $x \in B_p$ implies $p \in B_x$. As $B_x = B_y$, by symmetry again $y \in B_p$. In other words, both x and y are in B_p. Since y is arbitrary, this proves $[x] \subseteq B_p$, that is B_p contains the equivalence class [x] of its member x.

Now we consider the fourth assertion: Let $v \in V$ and B_v be its binary neighborhood. First, we want to prove $B' \subseteq E_B$: This means we need to show that for any u, which is B'-equivalent to v, is E_B-equivalent to v, that is, $B_v = B_u$.

For this purpose, we will show $B_v \subseteq B_u$: Note that $(u, v) \in B'$ by assumption. Let p be a point in the B_v. We have $(v, p) \in B$. Then, we would like to show that (u, p) belongs to B also. For this purpose, we assume, to the contrary, that $(u, p) \in B'$. Then we have $(v, p) \in B'$ because $(u, p) \in B'$ and $(u, v) \in B'$. The conclusion is absurd, so we conclude that $(u, p) \in B$. That is, $p \in B_u$. So we have $B_v \subseteq B_u$. By similar arguments, we can show that $B_u \subseteq B_v$. So we have proved that $B_u = B_v$. This conclude the proof: $B' \subseteq E_B$. QED

VII. Conclusion

In this paper, we address the most "ancient" problem in computer security. We are able to state the most general theorem for a DAC that can be secure from the point of view of information flow. There are more works that are coming, namely, if a given an F-DAC does not meet information flow security policy (IFSP), could we make a minimal changes so that it will meet the IFSP?

VIII. Programming Illustartions

A. Programmatic illustration

We also build a window application to form the friend lists, denoted by F-list(•), and trajectory lists, denoted by T(•), for each object • from a given E-DAC. The program analyzes the security of information flow for each of object and to determine if F-DAC satisfies ISFP.

The main design of the program is:

1. Structure

The information flow analysis model will contains the list of all objects in the given E-DAC from a file

Each object will have its own E list, F list, T list, and I list such that

F = the complement of E

T = the trajectory

$I = E \cap T$

2. Algorithm

- Read and parse the input file to form the list of all objects in the information flow analysis model.
- Build E list
- Build F list
- Build T list.
- Build I list

Build T list is the most critical part of program to achieve $O(n^3)$ performance.

To avoid infinitely looping and minimal run time, we have to make sure that the F list of each possible object can only be visited one time for traversing process while building the T list.

For each object, a temporary queue is used to hold all other possible objects that the information can be flowed to when traverse its F list. Insert any object in the F list to the built T list and place it in the queue if it has not been found in the built T list and is not the object which its T list is the built T list.

The pseudo code in C# style of Microsoft Visual Studio 2005 for this function as follows:

```
function buildTSet()
{
    place this main user in the temporary queue
    while (there is an item in the queue)
    {
        dequeue the 1st item in the queue known as curUser;
        foreach (user u in curUser's F-list)
        {
            if (u is not in the main user's T-list)
            {
                Add user u to T-list;
                If user u is not curUser, enqueue to place u into queue
            }
        }
    }
}
```

For easier understanding, let's walk through step by step on this function to build T-list for object E of example 2 in section A Illustrations above. The F-DAC is below:

$A \rightarrow \{A, C\}$

$B \rightarrow \{B, D\}$

$C \rightarrow \{A, C\}$

$D \rightarrow \{B, D\}$

$E \rightarrow \{E, A, B\}$

Place E in the queue.

Go to while loop, remove E from the queue for visiting its F-list to traverse. Go to foreach loop for traversing E's F-list. Object E is added to its T-list as the list is initially empty but is not placed in the queue since it is current object whose F-list is currently examined. Object A, the next object is in E's F-list, is also added to E's T-list since A is not found in the T-list; however, A now is placed in the queue as A's F-list has not yet examined. Same as A, the last object B in E's F-list is added to T-list and placed in the queue. After exiting foreach loop, the queue has 2 objects A and B. Go to while loop, A is removed from the queue for visiting its F-list to traverse. Go to foreach loop for traversing A's F-list. Object A now is already in E's T-list, so it is not added to E's T-list nor placed in the queue. C, the next and last object in A's F-list, is not in E's T-list. It is added to the T-list and placed in the queue, as C's F-list has not yet been visited. The 2nd time exiting this foreach loop, the queue also has 2 objects B and C. Same as A, D is added to E's T-list and placed in the queue when traversing B's F-list. The queue now also has 2 objects C and D after the 3rd time exiting foreach loop. Go back to the while loop to remove C from the queue for visiting its F-list to traverse. As A and C are both in E's T-list and their F-list have been visited, no action is performed. Same as C, no object is added nor placed in the queue while traversed D's F-list. After the 5th time exiting the foreach loop, the queue now is empty since C and D have been removed from the queue. The while loop is exited and E's T-list is completely built.

3. Testing result

We have done the test for all possible cases of 4 and 5 objects. The results are following:

a. Four objects

For IFSP:

Total number of objects = 4

Total number of cases examined = 4096

Total number of 0 object secure = 699

Total number of 1 object secure = 1140

Total number of 2 objects secure = 1098

Total number of 3 objects secure = 804

Total number of secure (all 4 objects are secure) = 355

For ACWSP:

Total number of cases examined = 355

Total number of cases met CWSP requirement = 15

b. Five objects

For IFSP:

Total number of objects = 5

Total number of cases examined = 1048576

Total number of 0 object secure = 412004

Total number of 1 object secure = 336210

Total number of 2 objects secure = 176980

Total number of 3 objects secure = 84720

Total number of 4 objects secure = 31720

Total number of secure (all 5 objects are secure) = 6942

For ACWSP:

Total number of cases examined = 6942

Total number of cases met CWSP requirement = 52

REFERENCES

[1] Bell, D. 1987. Secure computer systems: A network interpretation. In Proceedings on 3rd Annual Computer Security Application Conference. 32-39.

[2] David D. C. Brewer and Michael J. Nash: "The Chinese Wall Security Policy" IEEE Symposium on Security and Privacy, Oakland, May, 1988, pp 206-214.

[3] Richard A. Brualdi, Introductory Combinatorics, Prentice Hall, 1992.

[4] W. Chu and Q. Chen Neighborhood and associative query answering, *Journal of Intelligent Information Systems*, 1, 355-382, 1992.

[5] S. A. Demurjian and S. A. Hsiao "The Multimodel and Multilingual Database Systems-A Paradigm for the Studying of Database Systems, " IEEE Transaction on Software Engineering, 14, 8, (August 1988)

[6] Denning, D. E. 1976. A lattice model of secure information flow. Commun. ACM 19,2, 236-243.

[7] Hsiao, D.K., and Harary, F.,"A Formal System for Information Retrieval From Files," Communications of the ACM, 13, 2(February 1970). Corrigenda CACM 13,3 (March, 1970)

[8] Wong, E., and Chiang, T. C., "Canonical Structure in Attribute-Based File Organization," Communications of the ACM, Vol. 14, No. 9, September 1971.

[9] C.E. Landhehr, and C. L. Heitmeyer:Military Message Systems:Requirements and Security Model, NRL Memorandom Report 4925, Computer Science and Systemss Branch, Naval research Laboratory.

[10] T. T. Lee, "Algebraic Theory of Relational Databases," The Bell System Technical Journal Vol 62, No 10, December, 1983, pp.3159-3204

[11] T. Y. Lin, Neighborhood Systems and Relational Database. In: *Proceedings of 1988 ACM Sixteen Annual Computer Science Conference*, February 23-25, 1988, 725

[12] "A Generalized Information Flow Model and Role of System Security Officer", Database Security: Status and Prospects II, IFIP-Transaction, edited by C. E. Landwehr, North Holland, 1989, pp. 85-103.

[13] T. Y. Lin, Neighborhood Systems and Approximation in Database and Knowledge Base Systems, Proceedings of the Fourth International Symposium on Methodologies of Intelligent Systems, Poster Session, October 12-15, pp. 75-86, 1989.

[14] T. Y. Lin, "Chinese Wall Security Policy - An Aggressive Model", Proceedings of the Fifth Aerospace Computer Security Application Conference, December 4-8, 1989, pp. 286-293.

[15] "Attribute Based Data Model and Polyinstantiation," Education and Society, IFIP-Transaction, ed. Aiken, 12th Computer World Congress, September 7-11, 1992, pp.472-478.

[16] T. Y. Lin, "Neighborhood Systems - A Qualitative Theory for Fuzzy and Rough Sets, "Advances in Machine Intelligence and Soft Computing, Volume IV. Ed. Paul Wang, 1997, Duke University, North Carolina, 132-155. ISBN:0-9643454-3-3

[17] T. Y. Lin "Granular Computing on Binary Relations I: Data Mining and Neigh-borhood Systems." In: Rough Sets In Knowledge Discovery, A. Skoworn and L. Polkowski (eds), Physica-Verlag, 1998, 107-121

[18] T. Y. Lin "Granular Computing on Binary Relations II: Rough Set Representations and Belief Functions." In: Rough Sets In Knowledge Discovery, A. Skoworn and L. Polkowski (eds), Physica - Verlag, 1998, 121-140.

[19] T. Y. Lin "Chinese Wall Security Model and Conflict Analysis," the 24th IEEE Computer Society International Computer Software and Applications Conference (Compsac2000) Taipei, Taiwan, Oct 25-27, 2000

[20] T. Y. Lin "Feature Completion," Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol 5, No. 2, May 2002, pp. 57-62. This is the proceeding for the workshop "Toward the Foundation on Data Mining" in PAKDD2002, May 6, 2002.

[21] T. Y. Lin "A Theory of Derived Attributes and Attribute Completion," Proceedings of IEEE International Conference on Data Mining, Maebashi, Japan, Dec 9-12, 2002.

[22] T. Y. Lin: Chinese Wall Security Policy Models: Information and Confining Trojan Horses. DBSec 2003: 275-287

[23] Teresa F. Lunt: Access Control Policies for Database Systems. DBSec 1988: 41-52

[24] A, Motro: "Supporting Gaol Queries", in :Proceeding of the First International Con-ference on Expert Database Systems, L. Kerschber (eds)m April 1-4, 1986, pp. 85-96.

[25] S. Osborn, R. Sanghu and Q. Munawer," Configuring RoleBased Access Control to Enforce Mandatory and Discretionary Access Control Policies," ACM Transaction on Information and Systems Security, Vol 3, No 2, May 2002, Pages 85-106.

[26] Z. Pawlak, Rough sets. International Journal of Information and Computer Science 11, 1982, pp. 341-356.

[27] Z. Pawlak, "On Conflicts," Int J. of Man-Machine Studies, 21 pp. 127-134, 1984

[28] Z. Pawlak, Analysis of Conflicts, Joint Conference of Information Science, Research Triangle Park, North Carolina, March 1-5, 1997, 350-352.

[29] Polkowski, L., Skowron, A., and Zytkow, J., (1995),"Tolerance based rough sets." In: T.Y. Lin and A. Wildberger (eds.), Soft Computing:Rough Sets, Fuzzy Logic Neural Networks, Uncertainty Management, Knowledge Discovery, Simulation Councils, Inc. San Diego CA, 55-58.

[30] Sandhu, R. S. Latticebased enforcement of Chinese Walls. Computer & Security 11, 1992, 753-763.

[31] Sandhu, R. S. 1993. Latticebased access control models. IEEE Computer 26, 11, 9-19.

[32] Sandhu,R.S.,Coyne,E.J.,Feinstein,H.L.,and Youman, C. E. 1996. Rolebased access control models. IEEE Computer 29, 2 (Feb.), 38 - 47.

[33] Sandhu,R.AND Munawer, Q. 1998. How to do discretionary access control using roles. In Proceedings of the Third ACM Workshop on RoleBased Access Control (RBAC '98, Fairfax, VA, Oct. 22{23), C. Youman and T. Jaeger, Chairs. ACM Press, New York, NY, 47-54.

[34] W. Sierpinski and C. C, Kreiger, General Topology, University Toronto press, 1952

[35] T.C. Ting, "A User-Role Based Data Security Approach", in *Database Security: Status and Prospects*, C. Landwehr (ed.), North-Holland, 1988.

[36] Demurjian, S., and Ting, T.C., "Towards a Definitive Paradigm for Security in Object-Oriented Systems and Applications," J. of Computer Security, Vol. 5, No. 4, 1997.

[37] Liebrand, M., Ellis, H., Phillips, C., Demurjian, S., Ting, T.C., and Ellis, J., "Role Delegation for a Resource-Based Security Model," Data and Applications Security: Developments and Directions II, E. Gudes and S. Shenoi (eds.), Kluwer, 2003.

[38] Phillips, C., Demurjian, S., and Ting, T.C., "Towards Information Assurance in Dynamic Coalitions," Proc. of 2002 IEEE Info. Assurance Workshop, West Point, NY, June 2002.

[39] L.A. Zadeh, Fuzzy sets and information granularity, in: M. Gupta, R. Ragade, and R. Yager (Eds.), Advances in Fuzzy Set Theory and Applications, North-Holland, Amsterdam, 3-18, 1979.

[40] L. Zadeh, "Some Reflections on Information Granulation and its Centrality in Granular Computing, Computing with Words, the Computational Theory of Perceptions and Precisiated Natural Language." In: T. Y. Lin, Y.Y. Yao, L. Zadeh (eds), Data Mining, Rough Sets, and Granualr Computing T. Y. Lin, Y.Y. Yao, L. Zadeh (eds)

[41] R. Turn: Advances in computer system security, 1988.

[42] http://www.scs.cs.nyu.edu/aos/notes/l19.pdf