*MTA SZTAKI Közlemények 36/1987 pp.* 123-144

# MATHEMATICAL MODELS OF DATA SECURITY PROCESSES IN CENTRALIZED AND DISTRIBUTED DATA BASE SYSTEMS

*B. SZAFRANSKI*

*Zawadzkiego str. 18/38*
*Warsaw, Poland*

Abstract

Program data security in currently exploited data base systems in general is based upon access control mechanism. The detailed analysis of data processing from data base leads to conclusion, that mechanism of this type do not guarantee secure data processing because of keeping secrecy. Therefore in the systems of especially rigorous requirements on efficiency data security it is nenecessary to inculcate besides access control mechanism also data flow control mechanism. The paper includes main elements of formal models of such mechanism. Subsequently basing upon properties of the models and conclusions from analysis of semantics of data processing from data base it has been demonstrated, that some models elements generate universally bounded lattices, that have been defined as data flow lattice and operation scope lattice. Continuing considerations we have defined the algebraic structure as a composition of above mentioned lattices. This structure fulfils all properties of the lattice and it is called data security lattice. Next we assumed that in each node of Distibuted Data Base System may exist local data base system in which data security policy is based on the local data security lattice. The formal model of determining the data security for the whole system is presented and a method of deriving a common lattice, which is called superlattice from local lattices is demonstrated. The consideration are illustrated with a number of examples.

## 1. Introduction

My paper deals with data security in database. It is sure, that user who gives away his data under control of data base system requires the guarantee from the system, that other users can use his data according to his demands. Program data security mechanism in currently exploited data base system are based upon either data access control conception or data flow control conception.

## The data access control

The mechanism of data access control /Fig.1.1/ base on conception of access privileges matrix. This matrix determines the access privileges of active objects (users, programs) to passive objects (data units). On the Fig.1.1 there is showed an example of access privileges matrix. You can see that in such systems we differ a number of active objects (in this case JONES,...,SMITH) and a number of passive objects (data units A, B, C). Moreover, there exist a number of operations / in our example GET,PUT/. These operations can be used by active object to process passive objects. The policy of data access control rely on checking if the operations issued in this process are admissible for this active object /in contex of his access privilege/. Looking on the picture you can notice, for example, that GET operation issued by JONES to process A will be legal, but PUT will not.

1. The base elements:
   - active objects /like users, programs/,
   - passive objects /like data units/,
   - data processing operations /like GET, PUT/,
   - access privileges matrix, which determines access privileges of active objects to passive objects
2. Interpretation of access privileges matrix

Example:

| | data unit A | data unit B | data unit C | ... |
|---|---|---|---|---|
| JONES | GET | GET, PUT | X | |
| .<br>.<br>. | | | | |
| SMITH | GET | GET | GET | |

JONES can read data from data units A and B, write to data unit B, but he can't process data from data unit C. SMITH can read data from data units A, B, C.

3. The rule of acting:

Data access control mechanism compares the operation specified in the program with the privilege in the matrix ,to reject unlegal operations.

Fig.1.1. The idea of data access control.

## The data flow control

The mechanism of data flow control /Fig.1.2/ checks the correctness of data flows between objects. Data flow between objects A and B appeares then, when data in any way are transfered from object A to B. Copying data from file A to B is a simple example of data flow. Data flow control conception requires creating of set of secrecy classes. For example such set can contain public, confidential, secret classes. The objects are given the secrecy class from this set. The policy of flow control rely upon flow relation, which determines ordering of secrecy classes. For example, according to flow relation data can flow from confidential file to secret file, but can't flow in opposite direction.

1. The base elements:
   - objects /like users, data units/,
   - secrecy classes /like secret, public, top secret/,
   - flow relation,
   - flow operations /operations, which transfer data between objects; operations like PUT, GET, COPY, UPDATE/,
   - secrecy attributes matrix.
2. Interpretation of flow relation and secrecy attr.matrix:

Example:

| Objects | Secrecy class |
|---|---|
| JONES | Secret |
| SMITH | Top secret |
| Data unit A | Public |
| Data unit B | Secret |
| Data unit C | Top secret |

Flow relation determines ordering of secrecy classes set,
Example:

(public, confidential, secret, top secret)

If ordering of above set is from left to right we can say:

"Data can flow from confidential data unit to secret, but those can not flow in opposite direction"

3. The rule of acting:

The data flow control mechanism examines each data processing operation, to reject these operations, which cause unlegal flow.

Fig.1.2. The idea of data flow control

## 2. The need of data security processes integration.

To clarify this problem we will consider very simple example, showed on Fig.2.1. You can see, that there exist two users U1 and U2 and two data units (files A and B). The user U1 can read the data from the file A and write to file B, but the user U2 can read from the file B only. The data access control mechanism can examine legality of operations using by the users. But it doesn't prevent the unlegal cooperating of users. For example, the user U1 according to his privileges reads up data from file A and he writes up these data to the file B. And now, user U2 can read these data from the file B because he has privilege of reading this file. It is easy to notice, the unlegal flow was realized, despite existing of data access control mechanism. Next we assume, the file A is secret and the file B is public. Then applying of data flow control mechanism could eliminate this unlegal flow, because it wouldn't permit data flow from the secret to public file.
From the other side , the data are transfered between objects as an result performing of different operations. Some of them are presented on Fig.2.2. It is obvious that they cause different effects in data base. For example, effects of doing UPDATE and ERASE operations can be potential uncomparable. The UPDATE operation is usually used to update one or more fields in record occurences, while the single performing of ERASE operation could cancel a big part of network data base, but unfortunatly they are treated by data flow control mechanism in the same way, as the flow operations. Therefore the users can require to differ these operations. From this point of view the data flow control mechanism don't satisfy such demand of users.

At the end we came to conclusion:

It is necessary to build integrated data security mechanism, which includes possibilities of data access control and flow control.

## 3. The simplified models of data security processes

The whole models of data access control and data flow control are in [ 1 ]. Below I citized only such elements of them, which are important from the aim of this paper.

### 3.1. Data access control

Def.1
Data access control model is e three:

$$AM=\langle Z, T, \Psi \rangle$$

where:
Z – objects names set,
T – operation names set,
$\Psi$ – operation scope relation, $\Psi \subset T \times T$.

To clarify this relation, let us introduce formally the definition of operation scope:

Def.2

An operation scope $t_i \in T$ is an operation set $\{ t_j \} \in 2^T$ such, that the ability of doing the operation $t_i$ implicates the ability of doing each operation $t_i \in \{ t_j \}$.

Def.3

The operation scope $\{ t_j \} \in 2^T$ of operation $t_1$ is smaller/equal/ than the operation scope $\{ t_k \}$ if only if $\{ t_j \} \subset \{ t_k \}$.

The operation scope relation is defined on the pairs of operation names.

Def.4

For $t_1, t_2 \in T$ we can say that $(t_1, t_2) \in \Psi$ if and only if the operation scope $t_1$ is smaller/equal/ than operation scope $t_2$.

Def.5

Data security mechanism basing upon data access control works correctly if it ensures that no data access is realized, which is not in accordance with determined operation scope relation.

3.2. Data flow control

Def.6

A data flow control model is a three:

$$FM = < Z, K, \varphi >$$

where:

Z – objects names set,
K – secrecy classes set,
$\varphi$ – flow relation,
$\varphi \subset K \times K$.

Def.7

For $k_1, k_2 \in K$ we say that $(k_1, k_2) \in \varphi$ if and only if data from secrecy class object $k_1$ can flow to object of secrecy class $k_2$.

Def.8

Data security mechanism basing upon data flow works correctly if it ensures, that no data flow is realized, which is not in accordance with defined flow relation.

## 3.3. Data security control

Def.9

A data security model is a three:

$SM = < Z, A, \lambda >$

where:

Z – object names set,
A – $A = K \times T$
$\lambda$ – security relation,
$A \times A$.

$((a_i, a_j), (a_l, a_k)) \in \lambda$

when $(a_i, a_l) \in \varphi \wedge (a_j, a_k) \in \Psi$

Def.10

Data security mechanism based on model SM works correctly if it ensures after performing any operation set, that no access and flow is realized, which is not in accordance with defined security relation.

According to the above definition data security model have the possibility of data access control anf flow control. Therefore it can be a base of integrated data security mechanism.

## 4. The lattice models of data security processes

### 4.1. The lattice structure

Def.11

The algebraic structure $< X, \quad , +, \quad >$ is a lattice if:
1/ X is a finite set,
2/ $<X, \leq>$ is a partially ordered set,
3/ $\oplus$ is a binary operator on X such that $x \oplus y$ is the least upper bound for any $x, y \in X$,
4/ $\odot$ is a binary operator on X such that $x \odot y$ is the greatest lower bound for any $x, y \in X$.

Operators $\oplus$ and $\odot$ we called the least upper and greatest lower bounds operators, respectively.

Def.12

A lattice is an universally bounded lattice if there exist elements 1 and 0 such that $x \leqslant 1$ and $0 \leqslant x$ for all $x \in X$.

We call the elements 1 and 0 the universal upper and universal lower bounds, respectively.

## 4.2. Data flow lattice

D. Denning in her work entittled A Lattice Model of Secure Information Flow proved on the base of data processing semantics analysis, that secrecy classes set K and flow relation create universally bounded lattice. The lattice will be called a flow lattice and noted as:

$< K, \varphi, \square, \Delta >$ with universal bounds $k_{max}$ and $k_{min}$ in which $\square$ and $\Delta$ are the least upper and greatest lower bounds operators, respectively.

## 4.3. Operation scope lattice

After analysis of relations between operations from the set T you can prove, that model elements AM, that is: operation names set T and scope relation create a universally bounded lattice $<T, \Psi, +, \bullet>$. For this reason it must be proved, that the following properties of the lattice are satisfied:

1/ T is a finite set,
2/ $< T, \quad >$ is a partially ordered set,
3/ + and $\bullet$ are the least upper and greatest lower bounds operators, respectively,
4/ there exist universal lower and upper bounds $t_{max}$, $t_{min}$, repectively.

Now we will point out the satisfying some of these conditions. The satisfying of condition (1) issues from practical features of real systems. Subsequently, the relation must be reflexive, transitive and antisymmetric; it results from condition (2). It is obvious, that is reflexive, because operation scopes of the same opeartions are equal. The fact, that is antisymmetric results from the real assumption, that operation scopes are not redundant. You can prove for most of data processing operations from data base, that $\Psi$ is transitive. Let us consider as an instance the set T={FIND, READ, UPDATE}. It is obvious, that before data reading you should first find them and before updating you should first read them. Therefore practical sense of the data processing requires besides the permission of executing the operation READ also the permission of executing the operation FIND. The permission of executing the operation UPDATE should contain the permission of executing the operation READ and through transitiveness the permission of executing the operation FIND.
The above considerations let us come to conclusion, that $<T, \Psi>$ is a partially ordered set.

Now we will consider a fragment of typical program demonstrated by means of data flow diagram of Fig.4.1. To check the correctness of such a program it would be required to examine the legality of each issued operation. It is worth to emphasize, that such and similar program sequences may occure many times in the program. Therefore it is important to search the ways of examining legality of the access which would decrease the number of necessary checks. For this reason it would be enough to check only the legality of operation having the greatest scope of activity /for example operation UPDATE/ to decide, whether the fragment of the prugram is correct. Moreover the way of checking correctness requires existence of an operator defined on T, which guarantees, that the way of defining the "resultant" operation does not reject the realizing of the legal access. It is the least upper bound operator, which satisfies the over mentioned conditions.
Continuing considerations you can prove the practical sense of remaining elements of the earlier defined lattice.
For instance, the sence of existence the lower bound operator results from the necessity of defining the data processing competence on the grounds of user's competence and the competence of program and terminal.
Summarising you can state, that if you introduce the lattice of operation scope into basic model AM, it is not only formal manipulation, but it results from the semantics of data base processing.

## 4.4. Data security lattice

Up to now we have discussed two independent lattices for data access control model and data flow control model. Basing upon above considerations we will define a composition of these lattices in this way:

$$< A, \lambda, \boxplus, \boxtimes > = < K, \wp, \square, \Delta > \blacksquare < T, \Psi, +, \cdot >$$

where:

$A =$ KxT - is a set of all pairs, where the first element is from secrecy class set and the second from the operation names set.

$\lambda \subset$ AxA - is a relation defined as follows:
$((a_i, a_j), (a_l, a_k)) \in \lambda$ when $(a_i, a_l) \in \wp \wedge (a_j, a_k) \in \Psi$

$\boxplus$ - is a binary operator defined as follows:
for any $(a_i, a_j), (a_l, a_k) \in A$
$(a_i, a_j) \boxplus (a_l, a_k) = (a_i \square a_l, a_j + a_k)$

$\boxtimes$ - is a binary operator defined as follows:
for any $(a_i, a_j), (a_l, a_k) \in A$
$(a_i, a_j) \boxtimes (a_l, a_k) = (a_i \Delta a_l, a_j \cdot a_k)$

$a_{max}, a_{min}$ - are the elements of such, that:

$$a_{max} = (k_{max}, t_{max})$$
$$a_{min} = (k_{min}, t_{min})$$

On the way of formal provement you can show, that such defined composition satisfies all properties of a lattice. Tetrad $\langle A, \lambda, ⊞, ⊠ \rangle$ is said to be a data security lattices.

## 4.5. Examples

Figure 4.3 demonstrates a simple linear ordering of secrecy classes set, which satisfies the properties of data flow lattice. Graphic demonstration of the ordering is the preceding graph. Fig.4.4 shows linear ordering of operation set, which satisfies the properties of operation scope lattice. On the other hand, the Fig.4.5 shows graphic interpretation of data security lattice derived from lattices of Fig.4.3 and Fig.4.4. It is required from data security model not to allow data of higher secrecy class flow to lower secrecy class with simulataneous legality examining of performed operations. From the security class demonstrated as an instance on Fig.4.5 results, that object permitted to update public file can read and write to the public file, but it can not update the private file, becouse in that case flow relation is not satisfied.

## 5. Data security in Distributed Data Base System

I think that you can agree with opinion that modern data base systems are frequently distributed. Usually it means that the data are kept at widely dispersed locations. In other words, data base may be distributed in phisicaly separated nodes. To manage such systems we create the distributed data base systems, shortly called DDB Systems. Each node of DDB system may contains of local data base management system DBMS. Each DBMS has some capabilities of data security and therefore it is reasonable idea to design common data security policy for any DDB system. Quality of target policy strictly depends on correctness and completness of the designing process. Therefore this process should be supported by mathematical models and methods. DDB system with data security possibilities may be represented by generic architecture of Fig.5.1. From this figure you see that local possibilities of data security are reflected by local data security lattices and the global policy of data security in whole DDB system by global data security lattice so called superlattice. It is very important for such a system that all local DBMS implement the same kind of data security, for example only data access control or only data flow control conception. Formally it means for example, that sets X can differ, but X must have the same nature of elements (for example security levels or categories or data processing operations). The draft of methodology, which supports the process of determing the common data security policy in DDB system shown on Fig.5.2. This methodology consists of some theoretical foundations and auxilliary algorithms. Theoretical foundations include the formal conditions for consistency diagnostics of local lattices set, the way of superlattice construction and the set of operations to transform the superlattice according to changes in configuration of DDB system (including or excluding the nodes). The algorithms are necessary to make the process of superlattice construction easier and faster. In this paper I'm not going to present the whole methodology, but I'll explain the practical interpratation two following topics:

- consistency conditions of local lattices set,
- construction of superlattice.

The formal specifications of those problems are described in details in [ 2 ].

## 5.1. Consistency and strict consistency of local lattices set

When we want to integrate local DBMS in one DDB system we can expect two situations:

- the data security policies in nodes are consistent
- and the other, when are not consistent.

For example, in two nodes the data processing operations could be ordered, from data security point of view, in different way and therefore we must check this situation, to answer the question:

- can we or not integrate such local DBMSs in DDB system?

For this aim, I have introduced formal conditions of consistency or strict consistency of local lattices set.

Satisfying of consistency condition means that we can not create two (or more) such chains of elements from union of local sets, which order any two elements, in opposite direction (in sense of lattice relations). Please now look at the Fig.5.3. There is a set of lattices in graph form, which is inconsistent, because there exist two chains (C,F,B) and (B,E,C), which order elements C and B in opposite directions. Therefore we can state that integrating the nodes which refere to local lattices set showed on this figure is not possible from data security point of view.

In the case of strict consistency condition we have other situation. This condition prevents introducing additional ordering between elements of local sets, which can arise as a result of combination of local relations in global lattice. An example of such situation is shown on Fig.5.4.

You can see that this set of local lattices set is not strictly consistent because lattices L2 and L3 additionally ordered elements B and C in lattice L1.

However, such set of lattice can be base of superlattice constructing, but the designers of data security mechanism must remember that global data security policy can change the local policies in some nodes.

## 5.3. Construction of superlattice

When checking of consistency conditions is finished we can begin to construct the superlattice, obviously only for consistent or strictly consistent local lattices set.

This process consists of two stages.

The first stage includes the construction of global partially ordered set

antisymetric. Suitable proof id presented in the paper [ 2 ].
In the second stage we must check whether received partially ordered set $\langle X, \delta \rangle$ generates the lattice. It results from this, that in general even strict consistent local lattices set must not generate lattice. Let we consider example of Fig.5.6.
You can see, that strict cinsistency local lattices set of figures a and b generates the partially ordered set, but not the lattice, because elements E,M and A,G have not the least upper and greatest lower bounds. For example you see that elements E,M have three upper bounds (F,K,1), but none of them have properties of the least upper bound.

## 6. Conclusions

Properties of data security lattice model can be utilized to construction of more effective access control mechanisms, data flow mechanisms or data security mechanisms. The above effectiveness results from the following conditions:

- the storage size necessary for writing privileges decreases evident through defining the ordering relations of secrecy classes and operation sets;
- introducing of lower and upper operators decreases the number of accesses to operation scope, data flow and data security relation, respectively. It activates considerably the process of legality checking;
- the proof of the property should be one of the elements of any mechanism design. Such requirement is especially referred to data security mechanisms in data base systems. In case, where a lattice for a real data base system can be derived, compactness and completness are defined univocally. In this case the lattice is supposed to be a base of data security mechanism;
- the lattice has the form of dynamic structure, which can be easily copied in computer algorithms.

$\langle X, \delta \rangle$ from local partially sets $\langle X_i, \delta_i \rangle$ and the second includes checking if received pair $\langle X, \delta \rangle$ generates the lattice.
In the first stage we create the global set X as an union of local sets and additional sets D and G(see Fig. 5.5).
The sets D and G will be empty, when in union of local sets exist the elements, which will be the greatest lower and least upper bounds in future, constructing superlattice. If such elements do not exist, then we must add elements named here $x_d$ and $x_g$.
Next, we create the global relation $\delta$ . To do it, we include to $\delta$ all pairs, which belong to local relations and its combinations. To prove, that relation received in this way partially ordered set X we should prove that it is reflexive, transitive and

Bibliography

| | |
|---|---|
| 1. SZAFRAŇSKI B. | A Data Security Model in Data Base, ICS PAS Reports,Warsaw,1979, No 352. |
| 2. SZAFRAŇSKI B. | The Model of Extended Data Macroflows Control in Distributed Data Base Activity, Proc. of the Conf. on Systems Sciences,Wrocław,1986. |

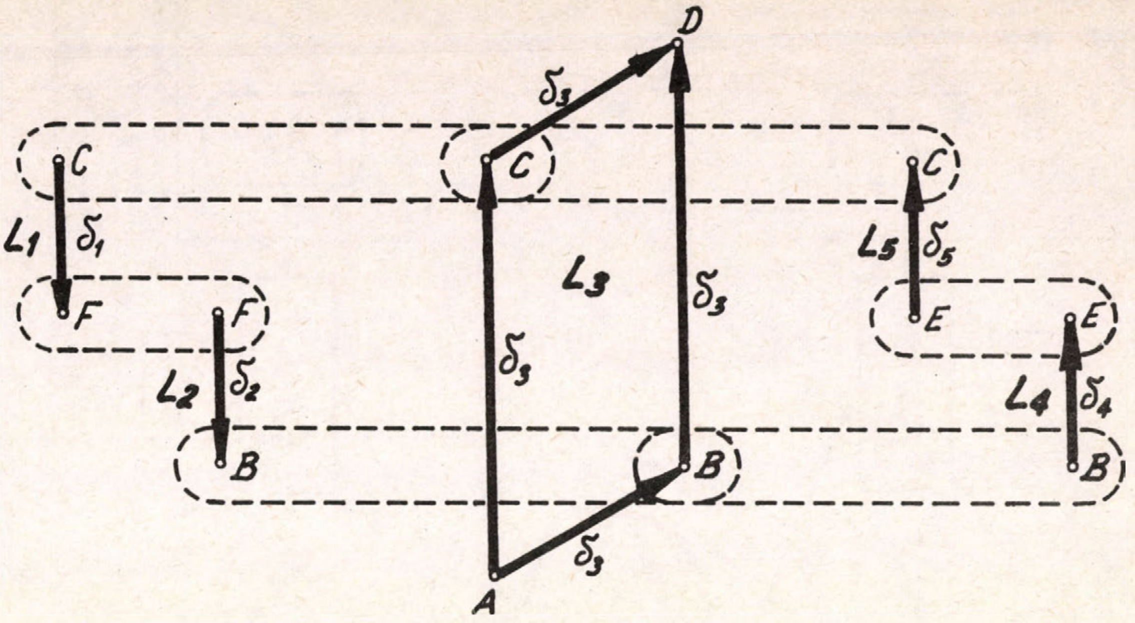

D
C
C
C
L1
δ1
L3
L5
δ5
δ3
δ3
F
F
E
E
L2
δ2
L4
δ4
B
B
B
δ3
A

Fig. 2.1

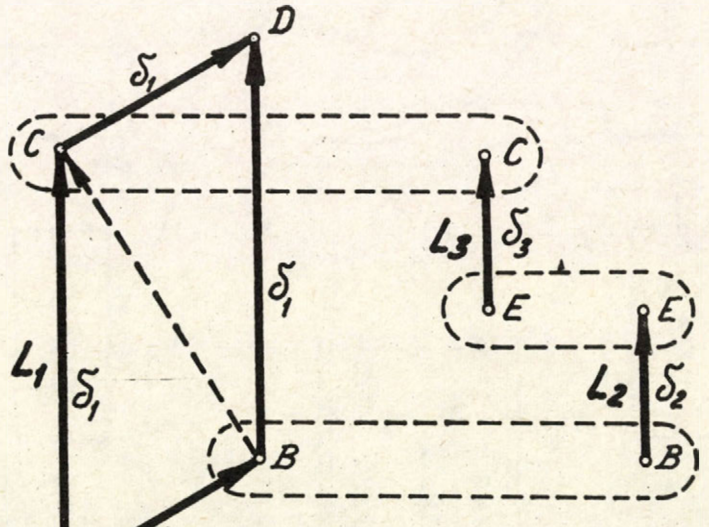

D
δ1
C
C
L3
δ3
δ1
E
E
L1
δ1
L2
δ2
B
B
δ1
A

Fig. 2.2

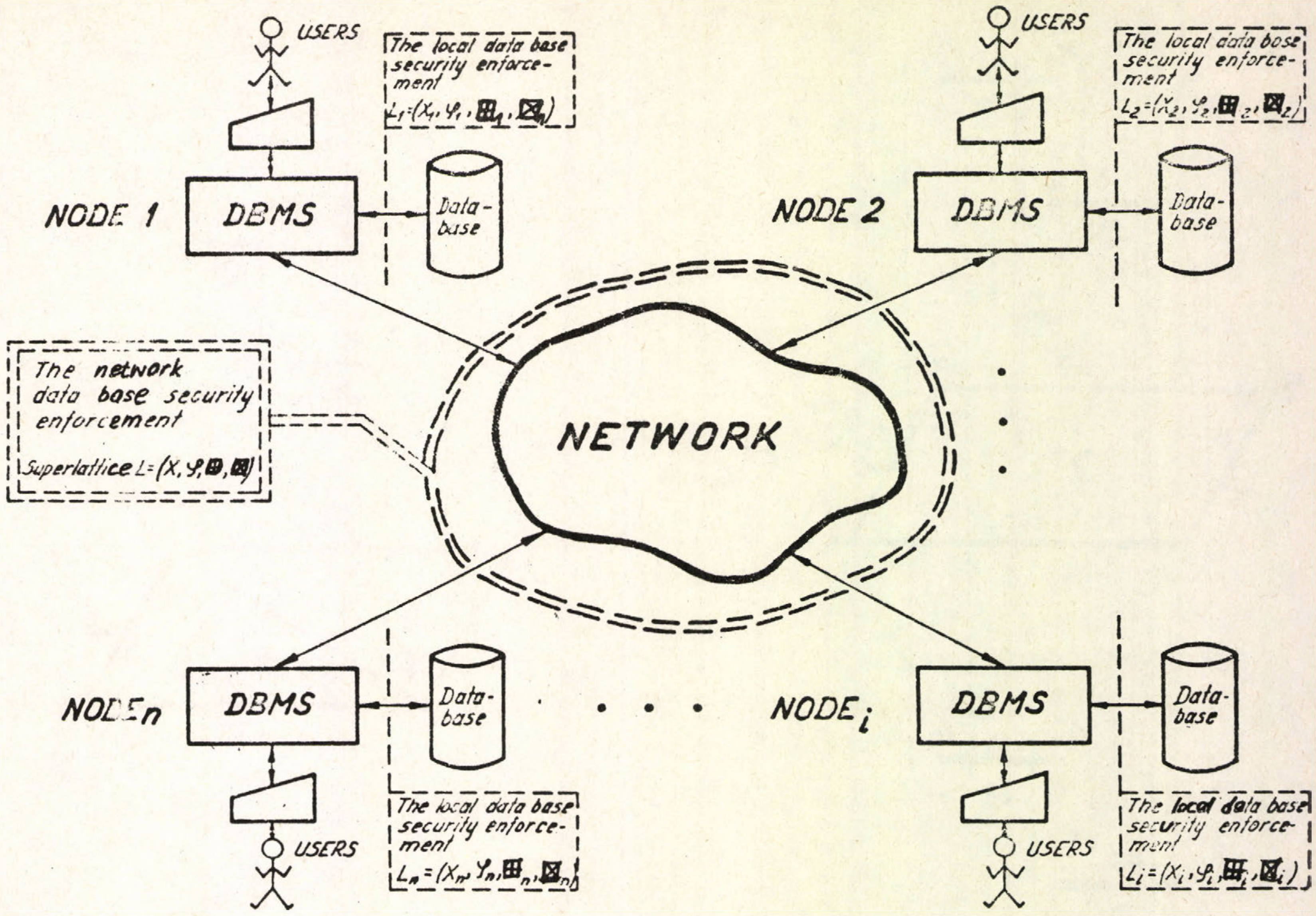

USERS
The local data base security enforcement
L₁=(X₁, φ₁, ⊞₁, ⊠₁)
NODE 1
DBMS
Data-base
USERS
The local data base security enforcement
L₂=(X₂, φ₂, ⊞₂, ⊠₂)
NODE 2
DBMS
Data-base
The network data base security enforcement
Superlattice L=(X, φ, ⊞, ⊠)
NETWORK
NODEn
DBMS
Data-base
The local data base security enforcement
USERS
Lₙ=(Xₙ, φₙ, ⊞ₙ, ⊠ₙ)
NODEᵢ
DBMS
Data-base
The local data base security enforcement
USERS
Lᵢ=(Xᵢ, φᵢ, ⊞ᵢ, ⊠ᵢ)

Fig. 4.1

$K = \{1, 2, 3\}$
$1 \equiv$ IN PUBLIC
$2 \equiv$ IN PRIVATE
$3 \equiv$ IN SECRET
$(k_i, k_j) \in \varphi$ IF $(k_i \leq k_j)$.
$k_i \square k_j = max(k_i, k_j)$.
$k_i \Delta k_j = min(k_i, k_j)$.
$k_{min} = 1, k_{max} = 3$

FIG. 4.2. DATA FLOW LATTICE.

$$\begin{tikzcd}
3 \\
2 \arrow[u] \\
1 \arrow[u]
\end{tikzcd}$$

$T = \{1, 2, 3\}$
$1 \equiv$ FIND
$2 \equiv$ READ
$3 \equiv$ UPDATE
$(t_i, t_j) \in \Psi$ IF $(t_i \leq t_j)$.
$t_i + t_j = max(t_i, t_j)$.
$t_i \cdot t_j = min(t_i, t_j)$.
$t_{min} = 1, t_{max} = 3$

FIG. 4.3. OPERATION SCOPE LATTICE

$A = \{(k_i, t_j) : k_i \in K, t_j \in T\}$.
$((k_i, t_j), (k_l, t_k)) \in \lambda$ IF
$(k_i, k_l) \in \varphi \wedge (t_j, t_k) \in \Psi$
$(k_i, t_j) \boxplus (k_l, t_k) =$
$= (max(k_i, k_l), max(t_j, t_k))$
$(k_i, t_j) \boxtimes (k_l, t_k) =$
$= (min(k_i, k_l), min(t_j, t_k))$.
$a_{min} = (1, 1), a_{max} = (3, 3)$.

$$\begin{tikzcd}
 & & (3,3) & & \\
 & (2,3) \arrow[ur] & & (3,2) \arrow[ul] & \\
(1,3) \arrow[ur] & & (2,2) \arrow[ul] \arrow[ur] & & (3,1) \arrow[ul] \\
 & (1,2) \arrow[ul] \arrow[ur] & & (2,1) \arrow[ul] \arrow[ur] & \\
 & & (1,1) \arrow[ul] \arrow[ur] & &
\end{tikzcd}$$

FIG. 4.4. DATA SECURITY LATTICE

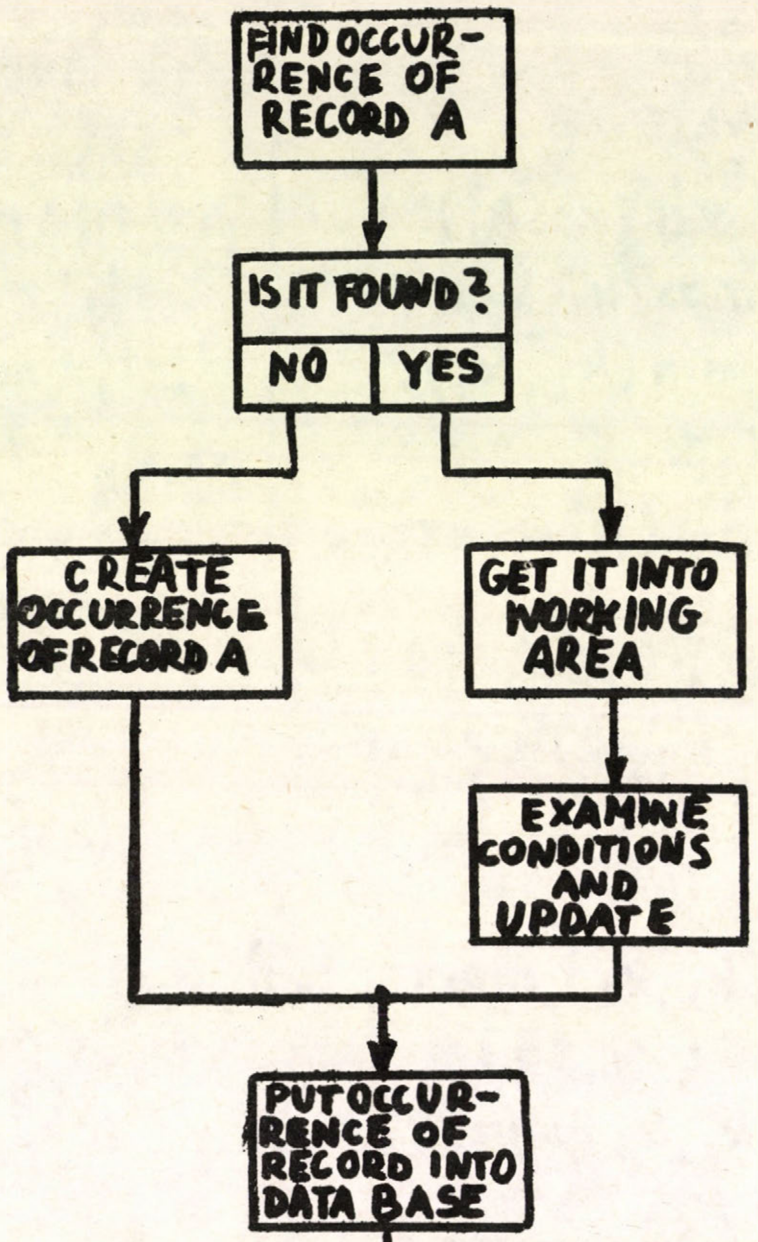

FIND OCCURRENCE OF RECORD A
IS IT FOUND?
NO
YES
CREATE OCCURRENCE OF RECORD A
GET IT INTO WORKING AREA
EXAMINE CONDITIONS AND UPDATE
PUT OCCURRENCE OF RECORD INTO DATA BASE

Fig. 5.1

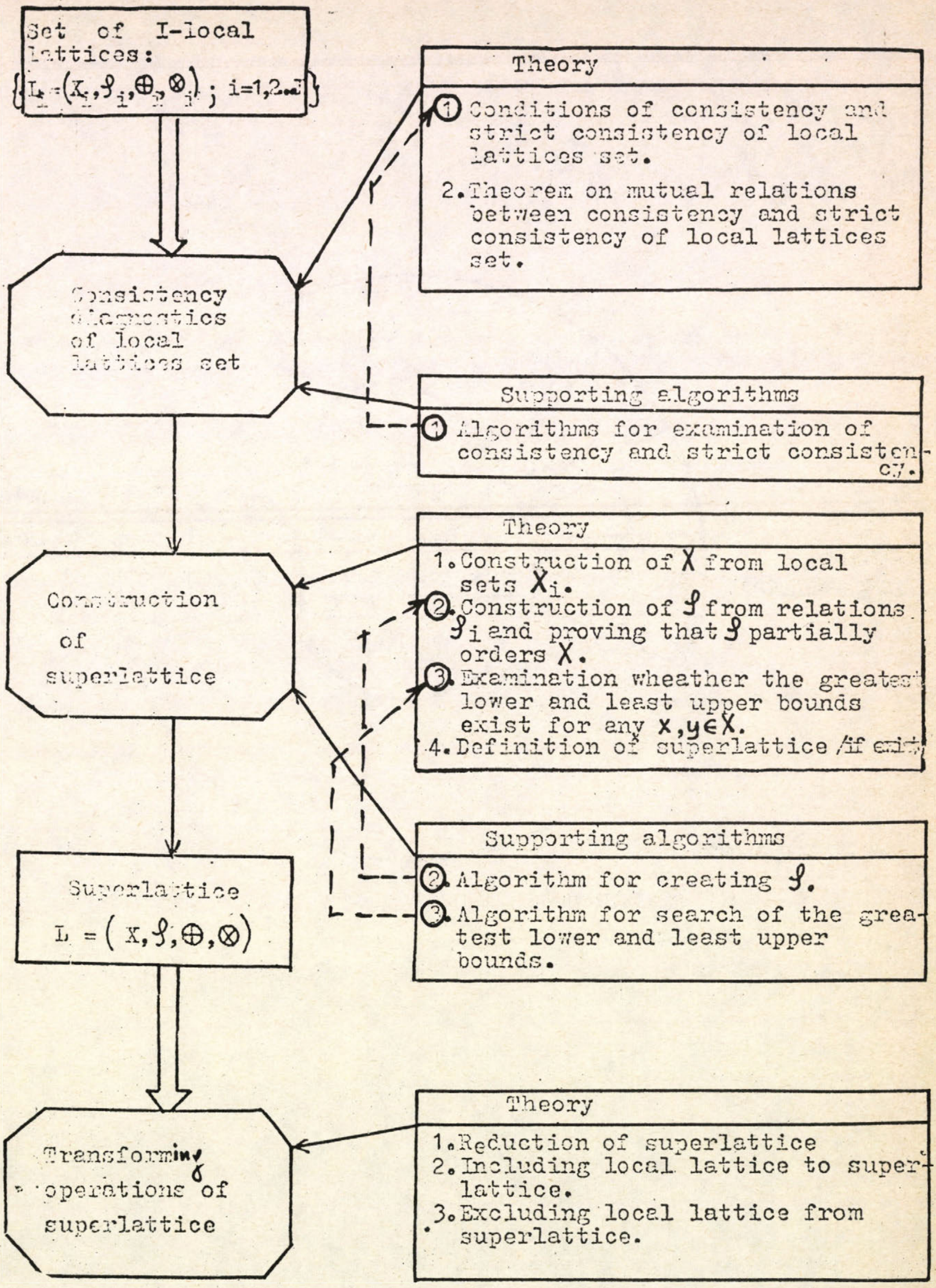

Set of I-local lattices: $\{L_i = (X_i, \mathcal{S}_i, \oplus_i, \otimes_i)\ ;\ i=1,2..I\}$
Consistency diagnostics of local lattices set
Construction of superlattice
Superlattice $L = (X, \mathcal{S}, \oplus, \otimes)$
Transforming operations of superlattice
Theory
1 Conditions of consistency and strict consistency of local lattices set.
2. Theorem on mutual relations between consistency and strict consistency of local lattices set.
Supporting algorithms
1 Algorithms for examination of consistency and strict consistency.
Theory
1. Construction of $X$ from local sets $X_i$.
2. Construction of $\mathcal{S}$ from relations $\mathcal{S}_i$ and proving that $\mathcal{S}$ partially orders $X$.
3. Examination wheather the greatest lower and least upper bounds exist for any $x, y \in X$.
4. Definition of superlattice /if exist
Supporting algorithms
2. Algorithm for creating $\mathcal{S}$.
3. Algorithm for search of the greatest lower and least upper bounds.
Theory
1. Reduction of superlattice
2. Including local lattice to superlattice.
3. Excluding local lattice from superlattice.

Fig. 5.2

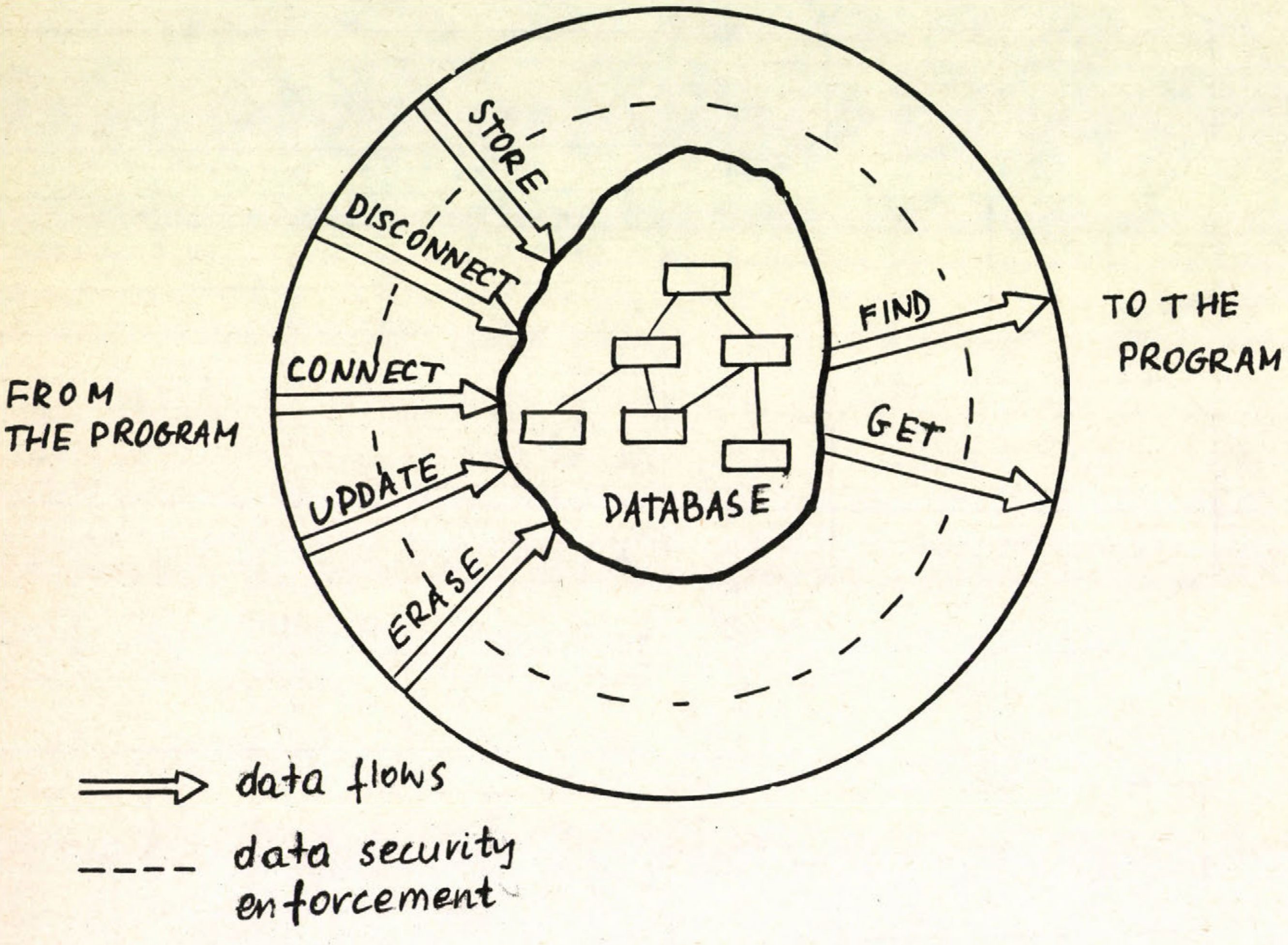
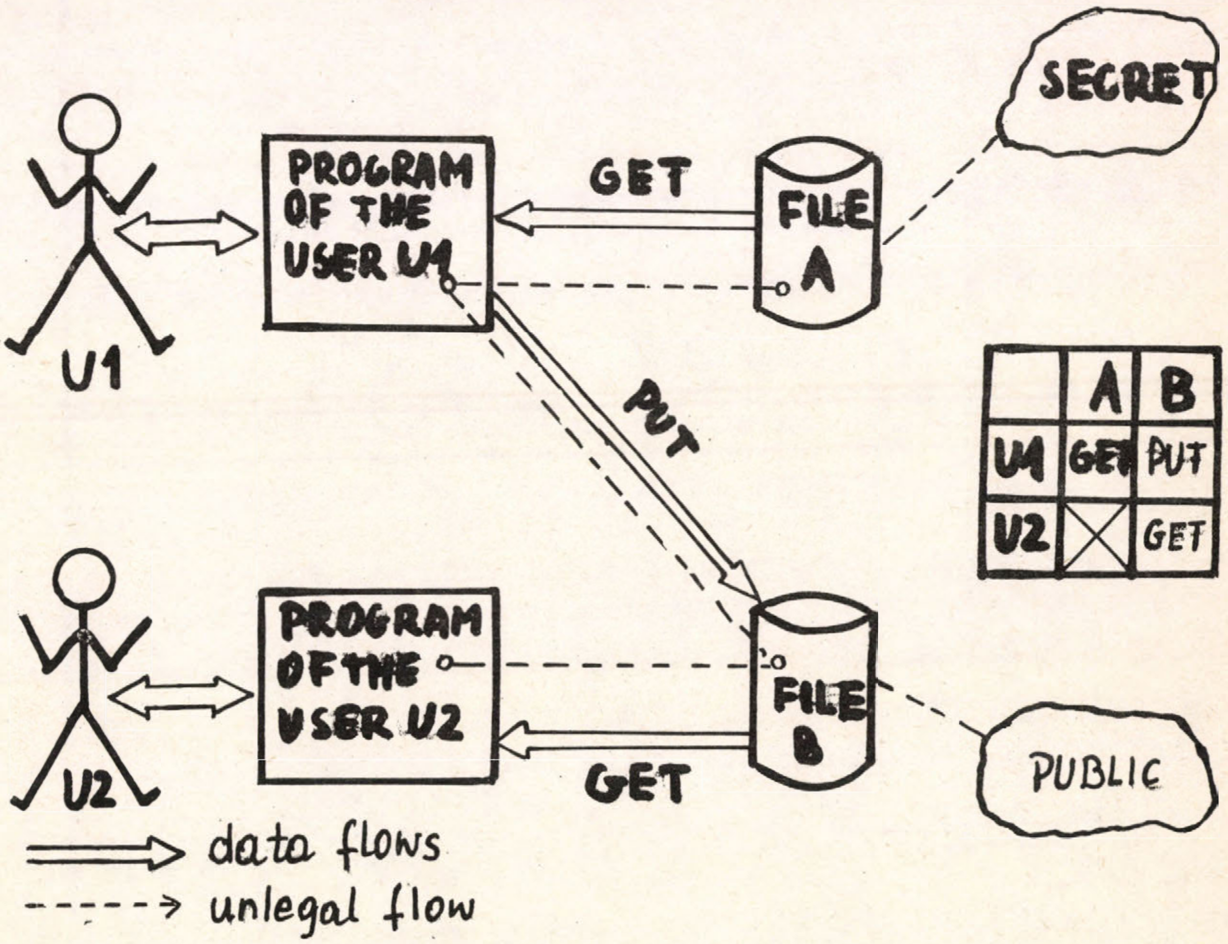

STORE
DISCONNECT
CONNECT
UPDATE
ERASE
FROM THE PROGRAM
FIND
GET
TO THE PROGRAM
DATABASE
data flows
data security enforcement

Fig. 5.3



| | A | B |
|---|---|---|
| U1 | GET | PUT |
| U2 | ☒ | GET |

Fig. 5.4

I. Construction of global set

$$X = \left(\bigcup_{i=1}^{I} X_i\right) \cup D \cup G$$

where:

$$G = \begin{cases} \emptyset - \text{if in } \bigcup_{i=1}^{I} X_i \text{ exists the element which satisfy condition of universal upper bound in creating relation.} \\ x_g - \text{in other case, } x_g \notin \bigcup_{i=1}^{I} X_i. \end{cases}$$

$$D = \begin{cases} \emptyset & \\ & \text{as above} \\ x_d & \end{cases}$$

II. Construction of global relation

$$\bigwedge_{x,y \in X} (x \delta y) \Longleftrightarrow \Big( \bigvee_{\substack{\{z_n\} \subset X \\ n=1,2,\dots,N}} \bigwedge_{n=2,3,\dots,N} \bigvee_{i \in \{1,2,\dots,I\}} (z_{n-1} \delta_i z_n) \wedge$$

$$\wedge (z_1 = x) \wedge (z_N = y) \vee (x = x_d) \vee (y = x_g) \Big).$$

Fig. 5.5

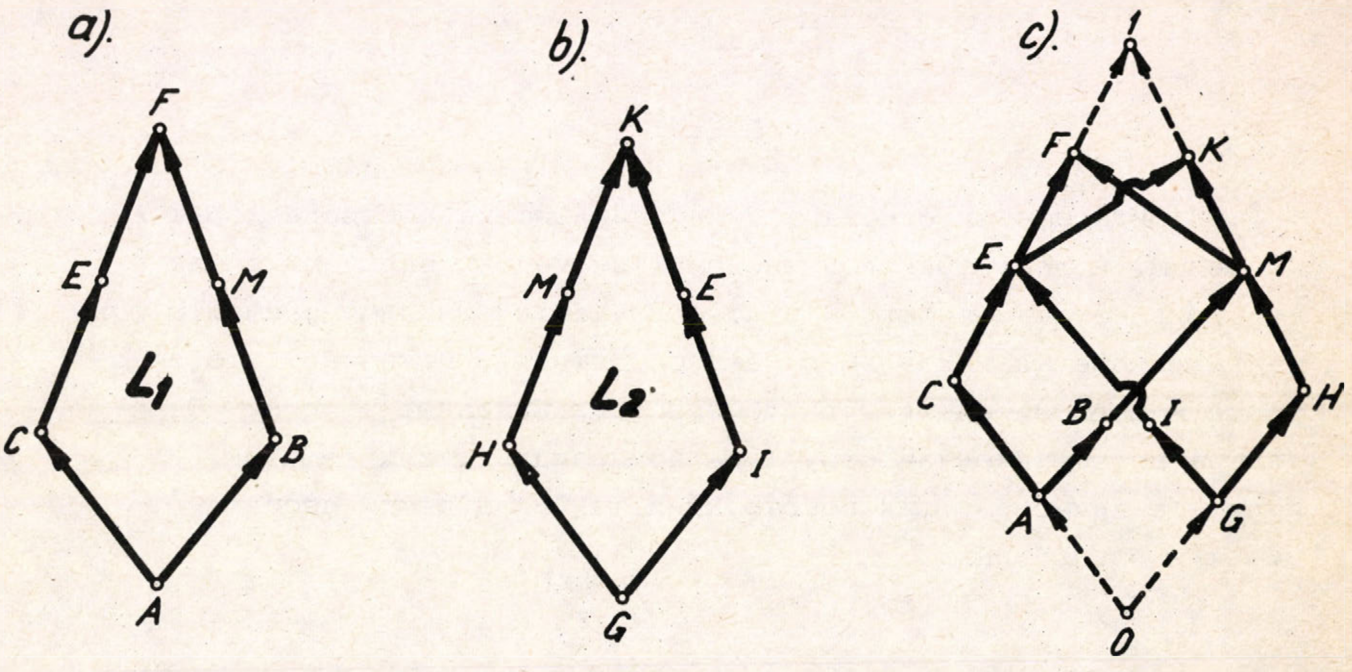


Fig. 5.6

## Математические модели процессов защиты данных в централизованных и разделенных системах баз данных

Б. Шафрански

### Резюме

Защита данных основана принципиально на контроле добытия к данным. Более эффективными являются методы включающие также контроль потока данных. В статье изучаются главные элементы формальных моделей механизмов такого контроля. Показано, что некоторые модели определяют равномерно ограниченные решетки. Определена алгебраическая структура композиций таких решеток. Результаты применены для конструкции так называемой "security" решетки баз данных.

## KÖZPONTOSITOTT ÉS ELOSZTOTT ADATBÁZIS RENDSZEREKRE VONATKOZÓ ADAT-BIZTONSÁGI ELJÁRÁSOK MATEMATIKAI MODELLJEI

B. Szafranski

### Összefoglaló

Az adat-védelem általában az adat-elérés ellenőrzésén alapszik. Biztonságosabbak azok a módszerek, amelyek az adat-mozgás folyamatait is ellenőrzik. A cikkben az utóbbi ellenőrzéseket is magába foglaló mechanizmusok formális modelljeinek elemei találhatók. A szerző megmutatja, hogy bizonyos esetekben a modellek egy egyenletesen korlátos hálót alkotnak, és megvizsgálja az igy kapott háló algebrai strukturáját. Az eredményeket az adat-bázisok u.n. biztonsági hálójának konstruálásához használja.