

# Normative Conflicts in a Confidentiality Policy

Frédéric Cuppens

Robert Demolombe

ONERA-CERT

2 Av. E. Belin

31055, Toulouse Cedex

France

email: {cuppens,demolomb}@tls-cs.cert.fr

## Abstract

In this paper, we are interested in the inference problem which arises in databases enforcing a multilevel security policy. This problem occurs when a user can derive unauthorized information with respect to the multilevel security policy from lower sensitive information stored in the database to which the user legally has an access. This problem is especially difficult to control because an unclassified user may try to infer secret information in using very different types of reasoning. In this paper, we focus on two different types of reasoning. The first one is only based on classical derivations in propositional logic and the second one takes into account situations where sensitive information may be illegally disclosed in using abductive reasoning. We propose a formalism based on a combination of doxastic and deontic logics to deal with these inference problems.

## 1 Introduction

In the real world, organizations define security policies that control how members of the organization may access data belonging to this organization. In this paper, we are interested in a special type of security policy namely the so-called multi-level security that is commonly used by military organizations in order to protect the confidentiality of their information. In this context, persons are associated with clearances according to their role within the organization, and every container of information as files, reports, messages is given a classification. Clearances and classifications are usually taken into a lattice of levels (for instance  $\{Unclassified, Confidential, Secret, Top\_Secret\}$ ). With respect to confidentiality, multilevel security could be summarized by one rule:

*A person may learn some data only if his clearance dominates the classification of this data.*

An application that has been of particular interest since the beginning of work on secure computer systems is the design of a database management system (DBMS) which enforces the requirements of the multilevel security policy. Denning in [Den82] showed that in order to have a correct model of the security requirements in a DBMS, it is more convenient to split up the problem of confidentiality into two sub-problems:

1. Internal information flow controls. A solution to this first problem guarantees that the SGBD would neither provide a user with any unauthorized piece of information with respect to the

multilevel security policy.

2. Inference control. A solution to this second problem guarantees that a user cannot derive unauthorized information with respect to the multilevel security policy from lower sensitive information stored in the database to which he legally has an access.

In [Cup93], we showed that to properly analyze the confidentiality of a system in a logical context, we need a formal definition of three concepts:

- The belief of each subject, we denote it  $B_A$ . The formula  $B_AP$  could be read “Agent  $A$  believes that  $p$ ”. In the database context,  $B_AP$  is actually a short cut for “**Agent  $A$  knows that the database believes that  $p$** ”.
- The permission to believe of each subject, we denote it  $PB_A$ . The formula  $PB_AP$  could be read “Agent  $A$  is permitted to believe that  $p$ ”. In the database context,  $PB_AP$  is actually a short cut for “Agent  $A$  is permitted to know that the database believes that  $p$ ”.
- The prohibition to believe of each subject, we denote it  $FB_A$ . The formula  $FB_AP$  could be read “Agent  $A$  is forbidden to believe that  $p$ ”. In the database context,  $FB_AP$  is actually a short cut for “Agent  $A$  is forbidden to know that the database believes that  $p$ ”.

Within this logical formalism, we have proposed to define the control of internal information flows by a logical formula  $B_A\varphi \rightarrow PB_A\varphi$  that could be read:

*If  $A$  believes  $\varphi$  then  $A$  should be permitted to believe that  $\varphi$*

In [BC92], we provide a semantics for this logical formula which leads to a new security condition called causality. Unfortunately, we also showed in [Cup93] that the security enforced via causality is not sufficient to deal with the inferential security problem.

Hence, our objective in this paper is to show how a logical formalism based on the concepts of belief, permission to believe and prohibition to believe may be used to formalize the inference problem in multi-level databases. Our formalism includes the possibility to insert cover stories in the database. A cover story is a lie which may be provided to users which are not enough cleared to believe the actual information. A common use for cover stories is to hide the existence of this otherwise sensitive information.

The remainder of this paper is organized as follows. Section 2 suggests through examples several types of inference in multilevel databases. Section 3 proposes a formalism which combines the doxastic and deontic logics to deal with these inference problems. Section 4 shows how to use this formalism to detect illegal inferences of sensitive information based on a classical derivation (i.e. in using a valid deductive proof in propositional logic). Section 5 shows that our formalism can also take into account situations where sensitive information are illegally disclosed in using abductive reasoning (i.e. a deductive proof may not be possible but a deductive proof could be completed by assuming some data). Finally, section 6 concludes the paper on further work that remains to be done.

## 2 Various types of reasoning

For the sake of simplicity, we will assume in this paper that there are only two security levels, the unclassified level denoted  $U$  and the secret level denoted  $S$ . Under this assumption, the inference

problem arises when a user with an unclassified clearance is able to draw conclusions about secret information in accessing unclassified information. This problem is especially difficult to control because an unclassified user may draw conclusions about secret information in performing several types of reasoning. Garvey and Lunt in [GL92] actually shows that most inferential security problems fall into one of three distinct classes depending on the type of reasoning which is used by the unclassified user.

## 2.1 Deductive channel

A deductive channel is the most restrictive type of inference; it occurs when an unclassified user is able to construct a formal deductive proof in propositional logic of a secret piece of information. Most of current research works (for instance [TFCO93, GLS91, GLQS92, Hin88]) only consider this type of inference channel. We show in section 4 how to detect a deductive channel.

**Example 1** Here is an example of deductive channel. Let us consider the four relations below:

<i>Salary</i> ( <i>Name</i> , <i>Sal</i> )	<i>Function</i> ( <i>Name</i> , <i>Func</i> )	<i>Experience</i> ( <i>Name</i> , <i>Year</i> )																		
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;"><i>Bill</i></td><td style="padding: 2px;">9000</td></tr> <tr><td style="padding: 2px;"><i>Bob</i></td><td style="padding: 2px;">14000</td></tr> <tr><td style="padding: 2px;"><i>Joe</i></td><td style="padding: 2px;">30000</td></tr> </table>	<i>Bill</i>	9000	<i>Bob</i>	14000	<i>Joe</i>	30000	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;"><i>Bill</i></td><td style="padding: 2px;"><i>Technician</i></td></tr> <tr><td style="padding: 2px;"><i>Bob</i></td><td style="padding: 2px;"><i>Engineer</i></td></tr> <tr><td style="padding: 2px;"><i>Joe</i></td><td style="padding: 2px;"><i>Manager</i></td></tr> </table>	<i>Bill</i>	<i>Technician</i>	<i>Bob</i>	<i>Engineer</i>	<i>Joe</i>	<i>Manager</i>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td style="padding: 2px;"><i>Bill</i></td><td style="padding: 2px;">1</td></tr> <tr><td style="padding: 2px;"><i>Bob</i></td><td style="padding: 2px;">5</td></tr> <tr><td style="padding: 2px;"><i>Joe</i></td><td style="padding: 2px;">10</td></tr> </table>	<i>Bill</i>	1	<i>Bob</i>	5	<i>Joe</i>	10
<i>Bill</i>	9000																			
<i>Bob</i>	14000																			
<i>Joe</i>	30000																			
<i>Bill</i>	<i>Technician</i>																			
<i>Bob</i>	<i>Engineer</i>																			
<i>Joe</i>	<i>Manager</i>																			
<i>Bill</i>	1																			
<i>Bob</i>	5																			
<i>Joe</i>	10																			

*Salary\_Scale*(*Func*, *Year*, *Sal*)

<i>Technician</i>	1	9000
<i>Technician</i>	2	9500
⋮		
<i>Technician</i>	40	15000
<i>Engineer</i>	1	12000
<i>Engineer</i>	2	12500
⋮		
<i>Engineer</i>	5	14000
⋮		
<i>Engineer</i>	40	20000
⋮		
<i>Manager</i>	10	30000
⋮		

Let us assume that the data stored in relation *Salary*(*Name*, *Sal*) are always secret data. This means that an unclassified user is forbidden to believe these data. On the other hand, let us assume that the data stored in the three other relations *Function*(*Name*, *Func*), *Experience*(*Name*, *Year*) and *Salary\_Scale*(*Func*, *Year*, *Sal*) are always unclassified data. Hence, an unclassified user is permitted to believe these data. Finally, let us assume that the following logical rule is sound in every model of the database we consider:

$$\forall name, func, sal, year, \\ Function(name, func) \wedge Experience(name, year) \wedge Salary\_Scale(func, year, sal) \\ \rightarrow Salary(name, sal)$$

This rule says that, from the knowledge of the function and years of experience of an employee and from the scale of salaries of the company, we may derive the salary of an employee. If unclassified user knows this general rule, then there is a deductive channel which exactly disclose the secret data stored in relation *Salary*.  $\square$

## 2.2 Abductive channel

An abductive channel exists when it is not possible to construct a deductive proof in using unclassified data stored in the database, but this proof could be completed by making assumptions about certain non protected data. [GLS91] suggests an informal method to control abductive channels. We now propose two different examples of abductive channels and show in section 5 how to formalize this type of inference channel.

**Example 2** This first example of abductive channel is the continuation of example 1. We always assume that the data stored in relation *Salary(Name, Sal)* are secret data and the data in relations *Function(Name, Func)* and *Salary\_Scale(Func, Year, Sal)* are unclassified data. However, let us now assume that the data stored in relation *Experience(Name, Year)* are secret data.

We also assume that the general rule which enables the salary to be derived from the function, years of experience and scale of salaries, is always sound. However, as the data stored in relation *Experience* are now protected, an unclassified user can no longer use this rule to directly infer the data stored in relation *Salary*. Unfortunately, this user can construct an abductive channel by trying to estimate the years of experience of a given employee. If this estimation is sufficiently precise, then an unclassified user will be able to exactly infer the employee's salary.  $\square$

However, one may argue that use of abduction is not necessary in this example. Indeed, if an unclassified user succeeds in evaluating the years of experience of a given employee, then it seems correct to consider that this information is perhaps a common knowledge of unclassified users. If this assumption actually holds, then this just means that data stored in relation *Experience* are wrongly classified. They must be unclassified data and use of abduction is no longer necessary because it is again possible to derive the employee's salary by constructing a deductive channel similar to the one discribed in example 1. We nevertheless consider that use of abduction may be necessary for at least two reasons. First, it is not always easy to give a complete description of every common knowledge of unclassified users. The second reason is explained in the following example.

**Example 3** Let us consider the two relations below:

<i>Stoppage(Name, Start, End)</i>			<i>Reason(Name, Disease)</i>	
<i>Bill</i>	01/01/94	15/01/94	<i>Bill</i>	<i>Influenza</i>
<i>Joe</i>	10/12/93	20/12/93	<i>Joe</i>	<i>Mumps</i>

These two relations are stored in a first database denoted *DB*. Data stored in relation *Stoppage* are unclassified and data stored in *Reason* are secret. Let us also assume that there also exists another relation *Prescription(Name, Medicine)* which says that a certain medicine was prescribed

to a given person. Let us also assume that the relation *Prescription* is stored in a second database *DB'* and that these data are not common knowledge of unclassified users. Finally, let us assume that the following general rule is sound:

$$\forall name, Prescription(name, Aspirin) \wedge Prescription(name, Antibiotic) \\ \rightarrow Reason(name, Influenza)$$

and that the unclassified users know this rule.

As the data stored in relation *Prescription* are not available for the database *DB*, an unclassified user cannot apply this rule to infer the reason of stoppage of a given employee. However, if relation *Prescription* is not protected in the second database *DB'*, an unclassified user can obtain these data by querying *DB'* and in this way construct a deductive channel. It is possible to detect this problem by abducting information related to the relation *Prescription* in database *DB*. This abductive reasoning may be used to infer that information in relation *Prescription* must be classified at secret.  $\square$

These two examples show that abduction may be useful in two different situations:

1. When the model of all common knowledge of unclassified users is not complete. In this case, use of abduction would provide the database security administrator with information whose knowledge enables secret information to be inferred by unclassified users; the database security administrator may analyze this information and decides whether or not they are likely to be common knowledge of unclassified users.
2. When the regulation is not complete. In this case, use of abduction provides the security administrator with a means to complete the regulation and to protect the information an unclassified user might try to gain to infer secret information.

### 2.3 Probabilistic channel

Another type of channel proposed in [GL92] is the probabilistic channel. It exists when an unclassified user is able to estimate the likelihood of a secret data in using unclassified data. In this context, Morgenstern in [Mor88] proposed a formal framework based on Shannon's information theory [SW63]. This approach provides theoretical means to evaluate the quantity of secret information that an unclassified user may gain in using unclassified information. Morgenstern defined the concept of a sphere of influence relative to some secret information: it is the information which provides a positive quantity of information (or more generally a quantity of information which exceeds a given threshold) on the secret information. It is an interesting approach but which seems difficult to put into practice.

There are obviously other types of inference channels based on other types of reasoning (for instance possibilistic reasoning). However, in the remainder of this paper, we only focus on the deductive and abductive channels.

## 3 Formalization of the Inference Problem

### 3.1 Basic Assumptions

We assume that the database content is represented by a set of sentences of a language  $L$  of Propositional Calculus

As a matter of simplification we only consider two classification levels: the “unclassified” level, and the “secret” level. The set of sentences  $U$  of  $L$  represents the set of unclassified sentences, and the set of sentences  $S$  of  $L$  represents the set of secret sentences.

It is assumed that  $U$  and  $S$  are disjoint sets, that is:  $U \cap S = \emptyset$ , and it is also assumed that some sentences may not be classified, that is, we do not necessarily have:  $U \cup S = L$ .

For convenience we will represent the set of consistent subsets of  $U$  instead of  $U$ , and the set of consistent subsets of  $S$  instead of  $S$ . These two sets of sets will be respectively represented by:  $[U](db_1, \dots, db_n)$  and by  $[S](db_1, \dots, db_n)$ , where  $db_1, \dots, db_n$  denotes the consistent subsets of  $U$  or of  $S$ .

We shall call “unclassified database” a consistent set of sentences  $udb$  which is not necessarily included in  $U$ , and we shall call “secret database” another consistent set of sentences  $sdb$  which is not necessarily included in  $S$ .

It is assumed that an unclassified user knows the  $udb$  content. It is not assumed that sentences stored in  $udb$  are true of the world, that means that  $udb$  is considered as a set of beliefs. The fact that an unclassified user  $u$  believes some sentence is represented by the doxastic modality  $B_u$ , and the fact  $u$  believes the sentence  $p$  is represented by  $B_u p$ .

It is assumed that a user believes all the logical consequences of his set of explicit beliefs (omniscience assumption), and that his set of beliefs is consistent. Then the modality  $B_u$  is formalized by the KD45 logic [Che80].

In the same way it is assumed that a secret user  $s$  knows the  $sdb$  content, and that the  $sdb$  content is considered as a set of beliefs. The fact that  $s$  believes  $p$  is represented by  $B_s p$ , where  $B_s$  obeys the axioms of the KD45 logic.

Finally it is assumed that concepts of permission and prohibition are independent in the sense that what is permitted is not necessarily the complement of what is not forbidden.

These two concepts are formalized by two modalities  $P$  and  $F$ . The fact  $u$  is permitted to believe  $p$  is represented by  $P(B_u p)$ , or  $PB_u p$  for short, and the fact that  $u$  is forbidden to believe  $p$  is represented by  $F(B_u p)$ , or  $FB_u p$  for short.

#### 3.1.1 Language Definition

Let  $L$  be a Propositional Calculus language defined as usual.

Let  $a, b, c, d, \dots$  be a recursively countable set of propositional variables denoted by  $VAR$ .

If  $p \in VAR$  then  $p \in L$ .

If  $p \in L$  then  $\neg p \in L$ .

If  $p \in L$  and  $q \in L$  then  $p \vee q \in L$ .

There is no other sentence in  $L$  than the sentences defined by the above rules.

Other logical connectives are defined by:  $p \wedge q \stackrel{\text{def}}{=} \neg(\neg p \vee \neg q)$  and  $p \rightarrow q \stackrel{\text{def}}{=} \neg p \vee q$ .

Let  $L'$  be a language defined as follows.

If  $p \in L$  then  $p \in L'$ .

If  $db_1 \in L, db_2 \in L, \dots, db_n \in L$  then  $[U](db_1, \dots, db_n) \in L'$  and  $[S](db_1, \dots, db_n) \in L'$ .

If  $p \in L'$  then  $B_u p \in L', B_s p \in L', PB_u p \in L'$  and  $FB_u p \in L'$ .

There is no other sentence in  $L'$  than the sentences defined by the above rules.

### 3.1.2 Semantics

Let  $\Sigma$  be the logic we are defining. A model  $M$  of  $\Sigma$  is defined as follows:

$$M = \langle W, R_u, R_s, D_u, D_s, R_{xu}, R_{xs}, U, S, P \rangle$$

where:

$W$  is a set of worlds.

$P$  is a function from  $\text{VAR}$  to  $2^W$ , and  $P$  is inductively extended to sentences in  $L$  by:  
 $P(\neg p) = W - P(p)$  and  $P(p \vee q) = P(p) \cup P(q)$ .

$R_u, R_s, D_u, D_s, R_{xu}$  and  $R_{xs}$  are relations on  $W \times W$ .

$U$  and  $S$  are two functions from  $W$  to  $2^L$ .

$M$  is a model of  $\Sigma$  iff it satisfies the following constraints.

C1: if  $p$  is a satisfiable<sup>1</sup> sentence of  $L$  then  $P(p) \neq \emptyset$ .

C2: relations  $R_u, R_s, R_{xu}$  and  $R_{xs}$  are serial, transitive and euclidean.

C3: let  $DB(U(w))$  be a set of subsets of  $L$  defined by:

$$DB(U(w)) = \{db : db \subseteq U(w) \text{ and } db \text{ is consistent} \}$$

we have:

$$\forall w \forall w' (w D_u w' \text{ iff } \exists db (db \in DB(U(w)) \text{ and } \forall w'' (w' R_{xu} w'' \text{ iff } M, w'' \models db)))$$

where  $db$  is interpreted as the conjunction of the sentences in  $db$ .

$$C4: \forall w \forall db (db \in DB(U(w)) \Rightarrow \exists w' \forall w'' (w' R_{xu} w'' \text{ iff } M, w'' \models db)).$$

C5: let  $DB(S(w))$  be a set of subsets of  $L$  defined by:

$$DB(S(w)) = \{db : db \subseteq L \text{ and } db \text{ is consistent and } \forall p (p \in S(w) \Rightarrow db \not\models p) \}$$

we have:

$$\forall w \forall w' (w D_s w' \text{ iff } \exists db (db \in DB(S(w)) \text{ and } \forall w'' (w' R_{xs} w'' \text{ iff } M, w'' \models db)))$$

$$C6: \forall w \forall db (db \in DB(S(w)) \Rightarrow \exists w' \forall w'' (w' R_{xs} w'' \text{ iff } M, w'' \models db)).$$

---

<sup>1</sup>Here "satisfiable" means satisfiable in Classical Propositional Calculus (CPC).

## Comments

Function  $P$  defines all the possible states of the world. Relations  $R_u$  and  $R_s$  respectively represent unclassified user's beliefs and secret user's beliefs for each state of the world.

$DB(U(w))$  can be intuitively interpreted as the set of allowed unclassified database states, and  $DB(S(w))$  can be intuitively interpreted as the set of states of a database which do not allow to derive any secret sentence.

Relation  $D_u$  represents, in each world  $w$ , the ideal belief states (in regard to the given regulation in the world  $w$ ) of a user who plays the role of a user [Cup94] cleared to know the unclassified database content.

Constraint  $C1$  imposes to have in  $W$  at least one world for each possible interpretation of the language  $L$ . Constraint  $C2$  imposes to doxastic modalities to obey the KD45 logic. In the context of  $C3$  and  $C4$  relation  $R_{xu}$  represents the fact that in an ideal world [CJ94] an unclassified user believes no more information than what is represented by some possible state of an unclassified database, that is information represented by some  $db$  in  $DB(U(w))$ . Then  $D_u$  and  $R_{xu}$  allow to represent the permission to believe some sentence (see figure 1).

Relation  $D_s$  represents, in each world  $w$ , the ideal belief states (in regard to the given regulation in the world  $w$ ) of a user who plays the role of a user not cleared to know the secret database content. In the context of  $C5$  and  $C6$  relation  $R_{xs}$  represents the fact that in an ideal world an unclassified user believes no more information than what is represented by a database which do not allow to derive any secret sentence, that is information represented by some  $db$  in  $DB(S(w))$ . Notice that from a  $db$  in  $DB(S(w))$  it cannot be derived any secret sentence. Then  $D_s$  and  $R_{xs}$  allow to represent the prohibition to believe some sentence (see figure 2).

Functions  $U$  and  $S$  define the regulation in each world, that is the set of sentences which are respectively assigned the unclassified level or the secret level.

Finally it has to be noticed that in this semantics we have not imposed the sentence  $PB_u p \rightarrow PB_u(PB_u p)$  to be a valid sentence. That means that it may happen that a user  $u$  has the permission to believe  $p$  while it is not the case that  $u$  has the permission to believe that he has this permission.

In the same way  $FB_u p \rightarrow PB_u(FB_u p)$  is not a valid sentence. That means that it may happen that a user  $u$  is forbidden to believe the sentence  $p$  while it is not the case that  $u$  has the permission to believe that he has this prohibition.

## Satisfiability Conditions

$M, w \models p$  means that the sentence  $p$  is true in the world  $w$  of the model  $M$ . The satisfiability conditions are:

- If  $p \in L$  :  $M, w \models p$  iff  $w \in P(p)$ .
- $M, w \models \Box p$  iff  $\forall w' \in W$   $M, w' \models p$ .
- $M, w \models [U](db_1, \dots, db_n)$  iff  $db_1, \dots, db_n$  is the set of the consistent subsets of  $U(w)$ .
- $M, w \models [S](db_1, \dots, db_n)$  iff  $db_1, \dots, db_n$  is the set of the consistent subsets of  $S(w)$ .
- $M, w \models B_u p$  iff  $\forall w' (wR_u w' \Rightarrow M, w' \models p)$ .
- $M, w \models B_s p$  iff  $\forall w' (wR_s w' \Rightarrow M, w' \models p)$ .
- $M, w \models PB_u p$  iff  $\exists w' (wD_u w' \text{ et } M, w' \models B_{xu} p)$ .
- $M, w \models FB_u p$  iff  $\forall w' (wD_s w' \Rightarrow M, w' \models \neg B_{xs} p)$ .

where  $w' \models B_{xu}p$  is an abbreviation of:  $\forall w''(w'R_{xu}w'' \Rightarrow w'' \models p)$ , and  $w' \models \neg B_{xs}p$  is an abbreviation of:  $\exists w''(w'R_{xs}w'' \text{ and } w'' \models \neg p)$ .

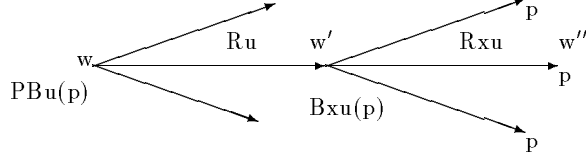


Figure 1: Permission to believe.

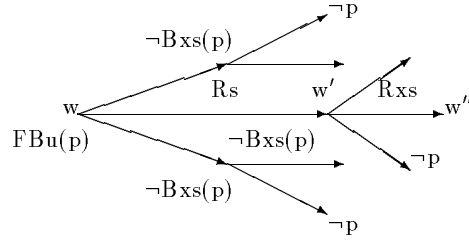


Figure 2: Prohibition to believe.

### Valid Sentences

$M \models p$  denotes the fact that  $p$  is true in the model  $M$ . We have:  $M \models p$  iff  $\forall w(M, w \models p)$ .

$\models p$  denotes the fact  $p$  is a valid sentence. We have:  $\models p$  iff  $\forall M(M \models p)$ .

### Properties of the $\Sigma$ Models

We adopt the following notations:

$$U'(w) = \{ p : \exists db(db \in DB(U(w)) \text{ and } db \vdash p) \}$$

$$S'(w) = \{ p : \forall db(db \in DB(S(w)) \Rightarrow db \not\vdash p) \}$$

$U'(w)$  represents the set of sentences that can be derived from an unclassified database. We have:  $U(w) \subseteq U'(w)$ .

$S'(w)$  represents the set of sentences that cannot be derived from any database which is not a secret database, that is a database  $db$  such that  $db \in DB(S(w))$ . We have:  $S(w) \subseteq S'(w)$ .

The following properties formally show that  $U'$  and  $S'$  respectively represent the set of sentences which are permitted or forbidden to an unclassified user to believe.

**Property 1:**  $M, w \models PB_u p$  iff  $p \in U'(w)$ .

**Property 2:**  $M, w \models \text{FB}_u p$  iff  $p \in S'(w)$ .

The following properties show how the concepts of permission to believe and prohibition to believe are structured by the concept of logical consequence.

**Property 3:**  $\models p \rightarrow q \Rightarrow \models \text{PB}_u p \rightarrow \text{PB}_u q$ .

**Property 4:**  $\models p \rightarrow q \Rightarrow \models \text{FB}_u q \rightarrow \text{FB}_u p$ .

We have the following corollaries of properties 3 and 4:

**Corollary 1:**  $\models \text{PB}_u(p \wedge q) \rightarrow \text{PB}_u p \wedge \text{PB}_u q$ .

**Corollary 2:**  $\models \text{PB}_u(p) \rightarrow \text{PB}_u(p \vee q)$ .

**Corollary 3:**  $\models \text{FB}_u p \rightarrow \text{FB}_u(p \wedge q)$ .

**Corollary 4:**  $\models \text{FB}_u(p \vee q) \rightarrow \text{FB}_u p \wedge \text{FB}_u q$ .

**Corollary 5:**  $\models p \leftrightarrow q \Rightarrow \models \text{PB}_u p \leftrightarrow \text{PB}_u q$ .

**Corollary 6:**  $\models p \leftrightarrow q \Rightarrow \models \text{FB}_u p \leftrightarrow \text{FB}_u q$ .

Finally we have the following properties.

**Property 5:**  $\not\models \text{PB}_u p \wedge \text{PB}_u q \rightarrow \text{PB}_u(p \wedge q)$ .

**Property 6:**  $\not\models \text{FB}_u(p \wedge q) \rightarrow (\text{FB}_u p \vee \text{FB}_u q)$ .

The intuitive meaning of property 5 is that it may happen that  $p$  can be derived from a given unclassified database state and that  $q$  can be derived from another unclassified database state, while there is no unclassified database state that allows to derive  $p \wedge q$ .

For instance if  $U(w) = \{a, a \rightarrow p, \neg a, \neg a \rightarrow q\}$ , we have  $p \in U'(w)$  and  $q \in U'(w)$ , while we also have  $p \wedge q \notin U'(w)$ .

The intuitive meaning of property 6 is that it may happen that  $u$  is forbidden to believe  $p \wedge q$ , while  $u$  is neither forbidden to believe  $p$  nor forbidden to believe  $q$ . Indeed it may happen that neither  $p$  nor  $q$  independently allows to derive a secret sentence, while their conjunction allows to derive it.

For instance if  $S(w) = \{a \wedge b\}$  we have neither  $a \in S'(w)$  nor  $b \in S'(w)$ .

This corresponds to a regulation of type ‘‘Chinese wall’’ [BN89], where the conjunction of two sentences may have a classification level strictly greater than each term in the conjunction when they are separately considered.

### 3.1.3 Axiomatics

The axiomatics of the  $\Sigma$  logic is defined by the following axiom schemas and inference rules.

- Axiom schemas of CPC.
- Modus Ponens.
- For each modality  $B_i$ , where  $B_i = B_u$  or  $B_i = B_s$ , we have:
  - $\vdash B_i p \wedge B_i(p \rightarrow q) \rightarrow B_i q$
  - $\vdash \neg B_i(\text{false})$
  - $\vdash B_i p \rightarrow B_i B_i p$

$$\begin{array}{l}
- \vdash \neg B_i p \rightarrow B_i \neg B_i p \\
- \frac{\vdash p}{\vdash B_i p}
\end{array}$$

- for the modality  $PB_{u,p}$  we have:

$$- \vdash [U](db_1, \dots, db_n) \rightarrow ( (\Box(db_1 \rightarrow p) \vee \dots \vee \Box(db_n \rightarrow p) ) \leftrightarrow PB_{u,p} )$$

- for the modality  $FB_{u,p}$  we have:

$$- \vdash [S](db_1, \dots, db_n) \rightarrow ( (\Box(p \rightarrow db_1) \vee \dots \vee \Box(p \rightarrow db_n) ) \leftrightarrow FB_{u,p} )$$

- for the modality  $\Box$  we have:

$$- \frac{\vdash p}{\vdash \Box p} \quad \frac{\not\vdash p}{\vdash \neg \Box p}$$

Notice that  $\Box(p \rightarrow q) \rightarrow (PB_{u,p} \rightarrow PB_{u,q})$  and  $\Box(p \rightarrow q) \rightarrow (FB_{u,q} \rightarrow FB_{u,p})$  are theorems of the  $\Sigma$  logic.

## 4 Deductive Channels

In this section, and in the next section, are defined in the framework of the  $\Sigma$  logic, conditions which characterize acceptable states of a database in regard to a given regulation. These conditions depend on the choice of a security policy, and we do not pretend that they have to be accepted in every situations. Nevertheless they show the expressive power and the flexibility of the  $\Sigma$  logic to formalize a security policy.

In this section are defined the conditions which prevent situations where deductive channels exist, in the sense defined in section 2.1. These conditions are called in the following “security constraints”.

A given state is defined by a tuple  $\langle U, S, \text{udb}, \text{sdb} \rangle$ , where  $U$  represents the set of unclassified sentences,  $S$  represents the set of secret sentences,  $\text{udb}$  represents the set of sentences stored in the unclassified database, and  $\text{sdb}$  represents the set of sentences stored in the secret database.

In the language  $L'$  defined in the previous section a state  $\langle U, S, \text{udb}, \text{sdb} \rangle$  is represented by the four assumptions:

$$[U](db_1, \dots, db_n), [S](db'_1, \dots, db'_m), B_u(\text{udb}), B_s(\text{sdb})$$

where  $db_1, \dots, db_n$  is the set of consistent subsets of  $U$ , and  $db'_1, \dots, db'_m$  is the set of consistent subsets of  $S$ .

We shall use the notations:  $R = [U](db_1, \dots, db_n) \wedge [S](db'_1, \dots, db'_m)$  and  $DB = B_u(\text{udb}) \wedge B_s(\text{sdb})$ .  $R$  will be called the “regulation” and  $DB$  will be called the “Database”.

We first present a security constraint about the regulation. This constraint expresses that in the context of a given regulation there is no sentence  $p$  such that an unclassified user  $u$  is permitted to believe  $p$ , and  $u$  is also forbidden to believe  $p$ . The formal representation of the constraint is:

$$(CR) \quad \vdash_{\Sigma} R \rightarrow \neg(PB_u p \wedge FB_u p)$$

A sentence  $p$  is defined as a cover story iff the unclassified user  $u$  believes  $p$  while the secret user  $s$  believes  $\neg p$ , then we have:

$$Cstory(p) \stackrel{def}{=} B_u(p) \wedge B_s(\neg p)$$

The security constraint which has to be satisfied to prevent deductive channels expresses that for every sentence  $p$  there is no consistent state where the user  $u$  believes  $p$ , and it is not permitted to  $u$  to believe  $p$ , and  $p$  is not a cover story, that is the set of sentences  $\{R, DB, B_u p, \neg Cstory(p), \neg PB_u p\}$  is inconsistent in the  $\Sigma$  logic. Then this constraint is represented by:

$$(SCD) \quad \vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \rightarrow (Cstory(p) \vee PB_u p))$$

Notice that from (CR) and (SCD) we can infer:

$$\vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \rightarrow (Cstory(p) \vee \neg FB_u p))$$

however the converse does not hold. In the case where there exist sentences which are neither unclassified nor secret, the (SCD) constraint imposes a stronger constraint. That means that (SCD) formalizes a more conservative attitude than the attitude represented by the above formula.

Notice that situations where cover stories do not exist can be represented by situations where the set of beliefs of the user  $u$  is a subset of beliefs of the user  $s$ . This is formally represented in the  $\Sigma$  logic by adding to the logic the axiom schema:  $B_u p \rightarrow B_s p$ . Therefore, if we assume that there is no cover story the security constraint takes the simplified form:

$$(SCD') \quad \vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \rightarrow PB_u p)$$

## 5 Abductive Channels

In this section we investigate situations where abductive channels may exist.

From an informal point of view, the sentence  $q$  can be abduced from the sentence  $p$  via abductive reasoning if there exists an assumption  $h$  such that  $q$  can be deduced by a classical derivation from  $\{p, h\}$ . However this definition has to be refined to remove trivial abductions. There are two kinds of trivial abductions. The first kind is when  $\{p, h\}$  is an inconsistent set of sentences. Indeed, in that case any sentence can be derived from  $\{p, h\}$ , and we can derive  $q$  even if there is neither a logical link between  $p$  and  $q$ , nor a logical link between  $h$  and  $q$ . The second kind of trivial abduction is when  $h$  allows to derive  $q$  without using  $p$ . Here again there is no logical link between  $h$  and  $p$ .

In more formal terms we say that  $q$  can be derived from  $p$  by abductive reasoning using the assumption  $h$ , and this fact is denoted by  $ABD(p, h, q)$ , if we have:

$$\vdash h \rightarrow (p \rightarrow q) \quad \text{and} \quad \not\vdash h \rightarrow \neg p \quad \text{and} \quad \not\vdash h \rightarrow q$$

that is:

$$\text{ABD}(p, h, q) \stackrel{\text{def}}{=} \Box(h \rightarrow (p \rightarrow q)) \wedge \neg\Box(h \rightarrow \neg p) \wedge \neg\Box(h \rightarrow q)$$

If there exists a sentence  $h$  such that  $\text{ABD}(p, h, q)$ , then  $h$  can be viewed as a channel which connects  $p$  and  $q$ . Therefore, if  $p$  is unclassified and  $q$  is secret,  $h$  ought to be secret in order to “break” the channel from  $p$  to  $q$ .

Notice that for every sentences  $p$  and  $q$  we have  $\text{ABD}(p, p \rightarrow q, q)$  and the assumption  $h = p \rightarrow q$  is the weakest assumption  $h$  such that we have  $\text{ABD}(p, h, q)$ . Indeed, from the definition of  $\text{ABD}(p, h, q)$  we have  $\vdash h \rightarrow (p \rightarrow q)$ . Therefore, if  $p \rightarrow q$  is classified at secret then, according to property 4, all the assumptions  $h$  such that  $\text{ABD}(p, h, q)$  are also classified at secret. This means that it is necessary and sufficient to classify the sentence  $p \rightarrow q$  at secret to break the channels from  $p$  to  $q$ .

We are now defining the security constraint which enables abductive channels to be prevented. The security constraint is different whether it is assumed, or not, that all the sentences which are believed by the unclassified user  $u$  are consequences of the unclassified database  $udb$ . In other terms  $u$  only believes sentences which are consequences of  $udb$ .

This assumption is sensible if  $udb$  contains the overall information a standard unclassified user believes without accessing to  $udb$ . That is, information which corresponds to common sense knowledge. This information will be called the “user’s model”.

In the context of this assumption the security constraint has to express that if  $q$  can be derived from  $p$  by abductive reasoning using the assumption  $h$ , and  $u$  believes  $p$ , then  $p$  is a cover story, or it is not forbidden to  $u$  to believe  $q$ , or  $u$  is forbidden to believe  $h$ . That is, the constraint is formally represented by:

$$(SCA) \quad \vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \wedge \text{ABD}(p, h, q) \rightarrow (\text{Cstory}(p) \vee PB_u q \vee FB_u h))$$

According to this constraint there are acceptable states where  $u$  believes  $p$  even if  $p$  informs about the secret information  $q$ . These states are acceptable because user  $u$  is forbidden to get any sentence  $h$  which enables  $q$  to be inferred.

The prohibition to believe  $h$  cannot be enforced if  $h$  is part of the common sense knowledge. This might happen if there was no user’s model in  $udb$ , or if this model is incomplete. Since it is not realistic to have a complete user’s model in  $udb$  the above assumption is rather careless, and it is more careful to have a stronger security constraint. A stronger constraint is not to consider as acceptable states the states where  $u$  is forbidden to believe  $h$ , and  $h$  allows  $u$  to derive a secret sentence. Then we have the constraint:

$$(SCA') \quad \vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \wedge \text{ABD}(p, h, q) \rightarrow (\text{Cstory}(p) \vee PB_u q))$$

Notice that in most cases if the constraint (SCA) is satisfied the constraint (SCD) is satisfied too. Indeed if we have:  $\not\vdash \neg p$  and  $\not\vdash p$ , that is if  $p$  is neither a tautology nor an inconsistent sentence, we have:  $\text{ABD}(p, \text{true}, p)$ , and (SCA) implies:

$$\vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \rightarrow (\text{Cstory}(p) \vee PB_u p \vee FB_u(\text{true})))$$

Since we usually do not have  $\vdash_{\Sigma} R \rightarrow FB_u(\text{true})$ , because if it would be the case that every sentence would be secret, we have:

$$\vdash_{\Sigma} R \wedge DB \rightarrow (B_u p \rightarrow (\text{Cstory}(p) \vee PB_u p))$$

which is the (SCD) constraint.

## 6 Conclusion

In this paper, we propose a general formalism for modeling database security and show how to use this formalism to characterize acceptable states of a multilevel security policy and of a database designed to enforce this multilevel security policy. This formalism includes a formal definition of the deontic concepts of permission and prohibition to believe some information using modal logic and possible world semantics. We also propose several security conditions which apply to detect deductive inference channels and show how to extend these security conditions to the case of abductive inference channels. We think that these security conditions provide a better understanding of the confidentiality problem in a multilevel database.

In the near term, interactive tools for assisting database security administrators may be developed using our logical formalism. These tools could provide effective assistance for security administrators in their attempt to give a consistent specification of the multilevel security policy and of the multilevel database's contents.

Another related problem not addressed in this paper is what counter measures apply in case of inference channels. We guess that several counter measures are possible, for instance update the unclassified user's model if some information are actually common knowledge, update the security policy, downgrade or upgrade some information in the database. However, if we consider a particular state of the database which violates the security requirements, then certain counter measures may apply to restore these security requirements and other measures may not. It actually depends on the particular situation we consider. Hence, a possible direction for future research would be to develop a strategy to choose which counter measure is the best candidate to restore the security requirements. This would represent another interesting extension of our work.

## References

- [BC92] P. Bieber and F. Cuppens. A Logical View of Secure Dependencies. *Journal of Computer Security*, 1(1):99–129, 1992.
- [BN89] D. Brewer and M. Nash. The Chinese wall security policy. In *IEEE Symposium on Security and Privacy*, Oakland, 1989.
- [Che80] B. Chellas. *Modal Logic*. Cambridge University Press, 1980.
- [CJ94] J. Carmo and A. I. J. Jones. Deontic database constraints and the characterisation of recovery. In A. J. I. Jones and M. Sergot, editors, *Second International Workshop on Deontic Logic in Computer Science*, Oslo, Norway, 1994.
- [Cup93] F. Cuppens. A Logical Analysis of Authorized and Prohibited Information Flows. In *IEEE Symposium on Security and Privacy*, Oakland, 1993.
- [Cup94] F. Cuppens. Roles and Deontic Logic. In A. J. I. Jones and M. Sergot, editors, *Second International Workshop on Deontic Logic in Computer Science*, Oslo, Norway, 1994.
- [Den82] D. Denning. *Cryptography and Data Security*. Addison-Wesley, 1982.

- [GL92] T. D. Garvey and T. F. Lunt. Cover Stories for Database Security. In S. Jajodia and C. Landwehr, editors, *Database Security, 5: Status and Prospects*. North-Holland, 1992. Results of the IFIP WG 11.3 Workshop on Database Security.
- [GLQS92] T. Garvey, T. Lunt, X. Qian, and M. Stickel. Toward a Tool to Detect and Eliminate Inference Problems in the Design of Multilevel Databases. In *Proc. of the Sixth IFIP WG 11.3 Working Conference on Database Security*, Vancouver, 1992.
- [GLS91] T. D. Garvey, T. F. Lunt, and M. E. Stickel. Abductive and Approximate Reasoning Models for Characterizing Inference Channels. In *Proc. of the computer security foundations workshop*, Franconia, 1991.
- [Hin88] T. H. Hinke. Inference Aggregation Detection in Database Management Systems. In *IEEE Symposium on Security and Privacy*, Oakland, 1988.
- [Mor88] M. Morgenstern. Controlling Logical Inference in Multilevel Database Systems. In *IEEE Symposium on Security and Privacy*, Oakland, 1988.
- [SW63] C. Shannon and E. Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL, 1963.
- [TFCO93] B. Thuraisingham, W. Ford, M. Collins, and J. O'Keefe. Design and implementation of a database inference controller. *Data & Knowledge Engineering*, 11(3), December 1993.