ORA-TM-98-030

A Test for Non-Disclosure in Security Level Translations[1]

David Rosenthal and Francis Fung

Odyssey Research Associates
33 Thornwood Dr, Suite 500
Ithaca, NY 14850

Abstract

Two security domains that want to exchange information securely may need to agree on translations of Mandatory Access Control (MAC) labels of their information, if their MAC labels have a different syntax or semantics. It is desirable that these translations do not introduce any confidentiality violations. In this paper we present a property, the Security Level Translation Property (*SLTP)*, which must hold if the security level translation functions satisfy MAC confidentiality. This property is in some sense the best possible test of the level translations in the absence of a "common domain" that gives the real relationships among the levels of the two domains.

1 Introduction

The need for constructing translations between MAC security levels arises when two security domains need to communicate, but the representation of the levels of those domains is not the same. Each domain may have its own syntax for assigning labels to objects and clearances to users; these labels and clearances do not necessarily have the same meaning in both domains. In order to securely send a message from one domain to the other, the two domains must agree on some method of translating the levels of one domain into those of the other domain, so that a user in the second domain can interpret the first domain's level appropriately. The translations can either be done on object labels, which are then compared with untranslated clearances, or the translations can be done on clearances, which are then compared to labels. The methods and analysis, although not identical, are essentially the same, and we examine only the object label translation method.

The Multilevel Information System Security Initiative (MISSI) [MISSI KPCMP] architecture, developed by the NSA, provides support for translations between security policies. MISSI is an architecture that enables efficient and secure communications across insecure channels, such as the Internet. The MISSI architecture is structured around the use of hierarchical domains, in which trust is propagated from a top-level root authority by the use of certificates. Part of this architecture supports communication between different domains. When two domains want to

[1] This work was supported by the National Security Agency under the direction of the Air Force Research Laboratory at Rome for contract F30602-96-C-0348

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2005		**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Test for Non-Disclosure in Security Level Translations	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Defense Advanced Research Projects Agency,3701 N. Fairfax Dr,Arlington,VA,22203**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Two security domains that want to exchange information securely may need to agree on translations of Mandatory Access Control (MAC) labels of their information, if their MAC labels have a different syntax or semantics. It is desirable that these translations do not introduce any confidentiality violations. In this paper we present a property, the Security Level Translation Property (SLTP), which must hold if the security level translation functions satisfy MAC confidentiality. This property is in some sense the best possible test of the level translations in the absence of a "common domain" that gives the real relationships among the levels of the two domains.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**16**	

interoperate, they use the mechanism of *cross-certificates*. A *cross-certificate* is a certificate issued by a Certificate Authority (CA) in a certain domain, whose subject lies in some other domain. Cross-certificates enable domains to convey trust in each other, without forcing them to commit their trust to a higher-level root. The MISSI level translations associated with cross-certified domains are not contained in the cross-certificate itself. Instead, the translations are contained in a file called the Security Policy Information File (SPIF), which is used to compare object labels to authorizations. The translations allow object labels from other domains to be compared with local authorizations, and local object labels to be compared with another domain's authorizations.

A translation of security labels should not result in a security leak. In particular, messages should only be sent to users that have the proper authorization to view that information. (The Bell-LaPadula Simple Security Property [BLP] should be satisfied.) In this paper, we describe a security check that can be applied to given pair of level translation functions. A complete non-disclosure analysis of translation functions requires a "common domain" that describes the "real" relationships between the levels. Such an analysis would require more information than is available in the representation of the functions and domain level orderings, i.e. in the SPIF. However, in the absence of full information about the common domain, we can still perform a partial analysis. We formulate the Security Level Translation Property (*SLTP*) for a pair of translations, which must be satisfied for the translations to be secure. *SLTP* is in some sense the best possible check for non-disclosure that can be done without knowledge of the real relationships among the labels. If *SLTP* is satisfied, then there exists a "comparison domain" into which both label orderings embed, such that non-disclosure is satisfied with respect to the comparison domain. There is, however, no guarantee that the constructed comparison domain will reflect the actual relationships among the levels of the two domains.

The rest of this paper is structured as follows. In section 2, we consider the representation of object labels as a partially ordered set and translations as partial functions between these orderings, and we discuss the notion of a common domain. Then, in section 3, we formulate the Security Level Translation Property (*SLTP*) and show that a pair of functions satisfies *SLTP* if and only if there exists some partially ordered set with the formal properties of a common domain in which the translation functions do not downgrade data. We also consider a simplified form that *SLTP* takes on when the functions are total and "order-compatible." In section 4, we formulate an equivalent form of *SLTP* for SDN.801 [MISSI SDN.801] military message labels.

2 Modeling the Security Level Translations

For the analysis that we describe here, we assume that the orderings on each domain are available, but that no information about the real relationships among the levels of the two domains is known. We will introduce the term *comparison domain* to mean a partially ordered set of levels containing copies of the two domains. Our analysis will use comparison domains in establishing the appropriateness of the desired security properties.

Suppose $(A,<)$ and $(B,<)$ are the partial orders of the security levels for the two domains. Let f represent the translation function from A to B and let f' represent the translation function from B to A. Note that f and f' might be only partial functions. We want to analyze the appropriateness of f and f'.

A *comparison domain* is a partially ordered set $(C,<)$ with $(A,<)$ and $(B,<)$ *properly embedded* into $(C,<)$, i.e., we have maps p_A and p_B from A to C, and B to C respectively, that satisfy:

For $x, y \in A$, $x < y \Leftrightarrow p_A(x) < p_A(y)$ and for $x, y \in B$, $x < y \Leftrightarrow p_B(x) < p_B(y)$ and
for $x, y \in A$, $x = y \Leftrightarrow p_A(x) = p_A(y)$ and for $x, y \in B$, $x = y \Leftrightarrow p_B(x) = p_B(y)$.

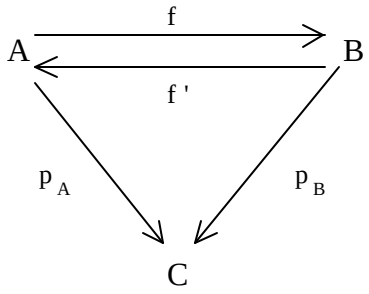

Figure 1: Relationship of translation functions

Any proper embedding of A and B into some C expresses an ordering relationship between levels in A and B. It may or may not represent the relationships of the common domain, i.e., the real relationships between A and B.

What security properties should the translation functions, f and f ', have with respect to a comparison domain for A and B? The translation functions should not allow a downgrade to occur. Hence we want

if x is in the domain of f, $p_A(x) = p_B(f(x))$ and
if y is in the domain of f ', $p_B(y) = p_A(f'(y))$.

This formula merely states that the translation functions can only raise levels. So, given a comparison domain C, we call f and f ' *level increasing* (relative to C), if the above two conditions hold for the proper embeddings p_A and p_B of A and B into C. If f and f ' are level increasing with respect to the common domain of A and B, then translating a label cannot cause a security leak, since a translated label is at least as a restrictive in who can view the object as the untranslated one.

3 Analysis of the Translation Functions

3.1 Analysis of the security level translation functions

3.1.1 Need for a common domain

As noted above, a complete analysis of the f and f ' translation functions requires using the common domain that captures the actual relationships between the levels of A and B. Analysis of just the translation functions in the SPIF (i.e., looking at just A, B, f, and f ') will not produce a total answer.

Consider the following example. Suppose A={S,TS} with the natural ordering and B={*protect*}. Let f(S)=*protect*, f(TS)=*protect* and f ' (*protect*)=TS. This completely specifies A, B, f and f ', but does not provide enough information to determine if the translation functions are in fact secure (i.e., whether f and f ' are level increasing in the common domain). If *protect* is equivalent to S (in the common domain) then TS information might be viewed by the equivalent of an S user. On the other hand, if *protect* is equivalent to TS then the level of the information may increase when translated, but there is no non-disclosure problem. Without a means to check the real relationship between the security levels (that is, using the common domain), we cannot guarantee that there is no non-disclosure violation.

3.1.2 Security Level Translation Property

Even though analysis without the common domain is only partial, it provides a way to identify non-disclosure violations with the information that is available.

We define the *Security Level Translation Property* (*SLTP*) as the following pair of conditions:

Condition 1: If x is in domain of f, y is in the domain of f ', and f(x) = y , then x = f ' (y).
Condition 2: If y is in domain of f ', x is in the domain of f, and f ' (y) = x, then y = f(x).

The two conditions represent the same property applied to the two different directions of the translation functions.

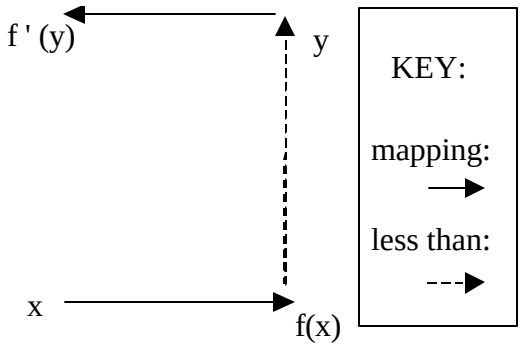

Figure 2: *SLTP* rule, condition 1

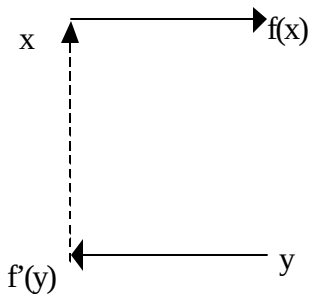

Figure 3: *SLTP* rule, condition 2

3.1.3 Main theorem for *SLTP*

Theorem:
For any two partially ordered sets (A,<), (B,<) , and functions f, f ' between them:

SLTP holds if and only if there exists a partially ordered set (C,<), with A and B properly embedded into C (i.e., a comparison domain C) and f, f ' satisfying the level increasing property with respect to C.

This theorem shows that *SLTP* is necessary for non-disclosure; whenever a comparison domain C exists for which f and f ' are level increasing with respect to C, then *SLTP* holds. The other direction shows that, with only the limited information of A, B, f, and f ', *SLTP* is the best possible sufficiency condition for non-disclosure. If *SLTP* is satisfied, there is a possible common interpretation of levels, namely C, for which the functions have the correct non-disclosure property (that is, they are level increasing). Hence, in the absence of additional criteria about the common domain (i.e., what are the real relationships between A and B), the translation functions are plausibly secure.

Note, however, that *SLTP* is not a sufficient condition to ensure that there is no non-disclosure problem, since the comparison domain C that is shown to exist in the theorem may not provide the real ordering relationships between the security levels in A and B (as shown in section 3.1.1).

Proof of Theorem:
<=
Let A and B be properly embedded into C, with f and f ' satisfying the level increasing property. We will show *SLTP*.

Suppose x is in the domain of f, y is in the domain of f ', and f(x) = y. By the level increasing property, we have $p_A(x) = p_B(f(x))$ and $p_B(y) = p_A(f'(y))$. Also, since p_B is a proper embedding, we have $p_B(f(x)) = p_B(y)$. Hence,

$p_A(x) = p_B(f(x)) = p_B(y) = p_A(f'(y))$ and thus $p_A(x) = p_A(f'(y))$.

Since p_A is a proper embedding, we see that $x = f'(y)$. This proves *SLTP* condition 1. A similar proof holds for condition 2.

=>
Assuming *SLTP*, we construct a comparison domain C satisfying the level increasing property, with A and B properly embedded into C.

Let C' be the disjoint union of A and B with the relation $=_C$ defined as follows:

If x in A and y in A, then $x =_C y$ iff $x =_A y$.

If x in B and y in B, then $x =_C y$ iff $x =_B y$.

If x in A and y in B, then $x =_C y$ iff there is some z in A such that z is in the domain of f and $x =_A z$ and $f(z) =_B y$ (see Figure 4).

If x in B and y in A, then $x =_C y$ iff there is some z in B such that z is in the domain of f ' and $x =_B z$ and $f(z) =_A y$.

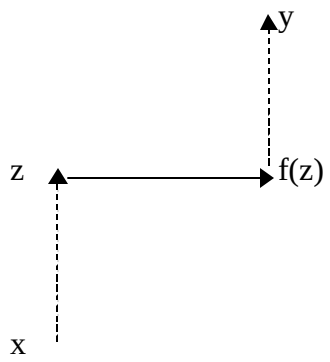

Figure 4: Part of inequality definition

Because the orderings $=_{A,} =_{C}$ agree on elements of A and $=_{B,} =_{C}$ agree on element of B, we can unambiguously drop the subscripts.

We will eventually construct a partially ordered set C that satisfies the conditions of the theorem by identifying elements of C'. First, we show some properties of C'.

Reflexive: It is easy to see that x = x for any x in C'.

Transitive: We next show transitivity of the partial order =.

There are several cases to consider. Suppose we have x, y, z such that x = y and y = z. We must show that x = z.

First, suppose x is in A.

Case: y is in A and z is in A:
Then x = z by transitivity of = in A.

Case: y is in A and z is in B:
Expanding the definition of y = z, we can find a q in A such that y = q and
f(q) = z. By transitivity in A and the hypotheses that x = y and y = q, we obtain x = q. Since we have f(q) = z, by the definition of = in C, we conclude that x = z.

Case: y is in B and z is in B:
Expanding the definition of x = y, we can find a q in A such that x = q and f(q) = y. Since y = z, we have f(q) = z by transitivity in B. So, by the definition of = in C, x = z.

Case: y is B and z is in A:
Expanding the definition of x = y, we can find a q in A such that x = q and f(q) = y.
Expanding the definition of y = z, we obtain an r in B such that y = r and f '(r) = z.

(See Figure 5.) By transitivity in B, we conclude that f(q) = r. By the property *SLTP*, we have q = f '(r). Then, since x = q and f '(r) = z, we find that x = z by transitivity in A.

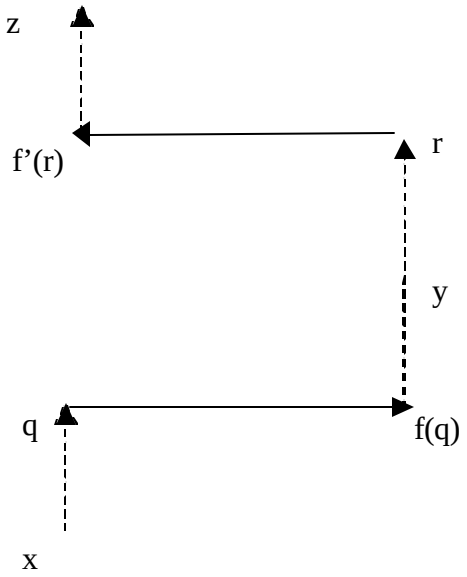

Figure 5: Relationship of x, y, and z

This completes the cases for x in A. The cases for x in B are similar. Hence the relation = is transitive.

Now, define x ~ y iff x = y and y = x. It is easy to check that this is an equivalence relation. We denote the equivalence class of x by [x].

Let C = (C'/~,<) be the set of equivalence classes of C' with respect to ~, where [a] < [b] iff (a = b in C' and not [a] = [b] in C). Note that this is well defined.

First, observe that C is indeed a partially ordered set. We now show C satisfies proper embedding and the level increasing property.

Proper Embedding:
We show that the maps $p_A(a) = [a]$ and $p_B(b) = [b]$ are proper embeddings of A and B into C.

If x, y in A, then ($p_A(x) = p_A(y)$) iff ([x] = [y]) iff (x = y and y = x in C') iff (x = y and y = x in A) iff (x = y in A).

If x, y in A then ($p_A(x) < p_A(y)$) iff ([x] < [y]) iff (x = y in C' and not [x] = [y] in C) iff (x = y in A and not x = y in A) iff (x < y in A).

A similar argument holds for the map p_B from B to C.

So, A and B are properly embedded in C.

Level Increasing:
For all a in the domain of f, a = f(a) in C by the definition of =. Similarly, b = f ' (b) in C. So, C satisfies the level increasing property.

Thus we have constructed a comparison domain C with the desired properties.
This proves the theorem.
?

We now note that the comparison domain C constructed in the theorem satisfies the property that an element of A is identified with an element of B exactly when the functions translate them to each other.

Proposition:
Let C be a comparison domain as constructed in the theorem. For any x in A and y in B, [x] = [y] iff (y = f(x) and x = f '(y)).

<=: If y = f(x) then x = x, x maps to f(x), and f(x) = y. Hence, by the definition of the partial order in C', we find that x = y. Similarly, y = x. Hence, [x] = [y].

=>: If [x] = [y] in C, then x = y and y = x in C'. So, there exists q in A such that x = q and f(q) = y and r in B such that y = r and f '(r) = y. We combine these inequalities to conclude that x = q = f '(r) and y = f(q) = r (see Figure 6). This produces the desired result.

?

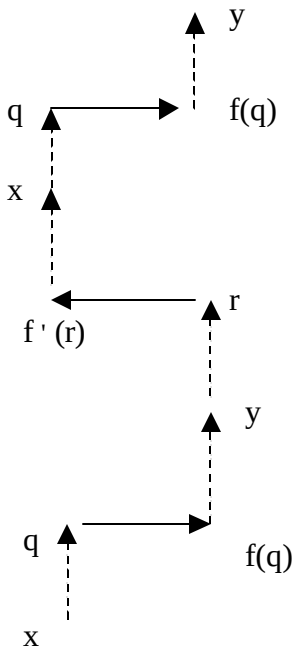

Figure 6: x = f(y) and y = f ' (x)

3.1.4 Total Functions

The natural interpretation of a partial translation function is that when an element is not in the domain of the function, the message is not sent. (There may, or may not, be some notification to the sender.)

We can convert such partial functions to total functions by:

1) Introducing a new level in A and B which is higher than any possible clearance.
2) Extending the functions to take this value whenever they are partial.

If the modified functions are used, the effect is that any message with a label not in the domain of the partial function will still not be sent, since it will fail the access constraint. So from an abstract point of view, the security analysis of the translation functions could assume only total functions. However, there are some practical differences and, to avoid confusion, we have used partial functions.

3.1.5 Simplification of *SLTP* under order compatibility

In some cases, it may be reasonable to impose some other constraints on the translation functions that are not strictly necessary for non-disclosure.

Consider the following property, which we call *order compatibility:*

For x and y in the domain of f, $x < y \Rightarrow f(x) = f(y)$, and
for x and y in the domain of f ', $x < y \Rightarrow f'(x) = f'(y)$.

Order compatibility says that the B's view of the ordering on A is compatible with A's view, although there may be some collapse. Similarly, this holds for A's view of B's ordering.

Order compatibility is not strictly necessary to satisfy non-disclosure (and hence not inferable from the property *SLTP*). For example, A could be {U, C}, B could be {S, TS} and C could be {U, C, S, TS} with the natural orderings. The translation function f(U) = TS and f(C) = S would not cause a leak, even though it violates order compatibility.

However, it is desirable to construct translation functions which are as faithful as possible. In particular, translation functions should generally not send a level to a level that is higher than necessary. If order compatibility is not satisfied, then this principle is violated. We show that if a level increasing function f does not satisfy order compatibility, then there is another function that is still level increasing but does not increase the level as much.

Suppose there are elements x and y such that $x < y$, $f(x) > f(y)$, where f is level increasing. Let g be the same as f, but with $g(x) = f(y)$. Then g is still level increasing, by the following argument. For all other elements except x, it follows because g agrees with f. And for x, since $p_A(y) = p_B(f(y))$ and $p_A(x) < p_A(y)$, we see that $p_A(x) < p_B(f(y))$. Hence, $p_A(x) < p_B(g(x))$, so g is still level increasing. Further, for any y in the domain of f, $g(y) = f(y)$ and $g(x) < f(x)$. Thus g is a more faithful alternative to f.

If the translation functions are total and they satisfy order compatibility, then *SLTP* can be expressed in a simple form.

Let *Alt_SLTP* (Alternate *SLTP)* be the formula: for x in A, x = f ' (f (x)); and for y in B, y = f (f '(y)).

Claim:
If f and f ' are total and order compatible, then f and f ' satisfy *SLTP* iff they satisfy *Alt_SLTP*.

=>: Suppose *SLTP* holds for f and f '. Pick any x in A. Since f and f ' are total, we can apply f to x, and then apply f ' to f(x). By *SLTP*, we see that x = f '(f(x)). Similarly, for all y in B, we have y = f (f '(y)). Thus, we have *Alt_SLTP*.

<=: Suppose *Alt_SLTP* holds for f and f '. Pick any x in A. Suppose we have y in B with f(x) = y. Since f ' is total and order compatible, we conclude that f '(f(x)) = f '(y). By *Alt_SLTP*, we find that x = f '(f(x)). By transitivity, we see that x = f '(y). This establishes the first condition of *SLTP*. The other condition follows similarly.
?

4 Military Message Levels

We now describe how *SLTP* relates to a military message level format from SDN.801.

4.1 A definition of military message levels

The SDN.801 representation for a military message level consists of

- a hierarchical security level,
- possibly some restrictive categories, and
- possibly some permissive categories of one or more types (permissive tag sets).

Not all combinations of hierarchical levels and categories are necessarily "valid" SDN.801 levels. For this discussion, we do not distinguish between valid and invalid levels.

If an object label contains restrictive categories, then a user must possess the authorization for all listed categories to access the object. Also, if an object label contains several permissive categories of a given type, then a user must possess clearance for at least one of the categories of that type. Note that if an object label contains permissive categories of several different types, then a user must possess clearance for at least one of the categories of each of those types in order to access the object.

We now formalize the ordering = on the set $L \times \mathbf{P}(R) \times \mathbf{P}(P)$, where L is a set of hierarchical levels (linearly ordered), R is a set of restrictive categories, and P is a disjoint union of sets P_i of permissive categories, and **P** means the power set (i.e., the set of all subsets of the set). Then (l1,r1,p1) = (l2,r2,p2) iff

1) l1= l2 and
2) r1 is a subset of r2 and
3) for every P_i, either p1 $\cap$ P_i is empty, or p1 $\cap$ P_i contains p2 $\cap$ P_i and p2 $\cap$ P_i is not empty. (The added complexity for the empty set is necessary because the absence of any permissive category in some type adds no access restriction for that type. On the other hand, for nonempty sets, fewer categories gives a tighter restriction.)

A security level translation in SDN.801 is defined in terms of how it maps hierarchical levels, restrictive categories, and permissive categories independently. For example, if the hierarchical level of f(h,r,s) is l2 then the hierarchical level of f(h,r2,s2) must also be l2 for any choice of category sets r2 and s2. In addition, a translation function is defined elementwise on a set of categories. In particular, for a level l = (h,R,P), the set of restrictive categories of the level f(l) is the union of the sets of restrictive categories of the levels f(h,{x},P) as x ranges over R; however, if some x in R is not in the domain of f, then the map f is not defined on the level containing the set R either. A similar discussion holds for permissive categories.

There are some SDN.801 restrictions on allowable security level translation functions:

- For any category x in the domain of f, the set f(x) must be non-empty, and similarly for f '.

- Mappings respect the "typing" of the categories. In particular, restrictive categories map to restrictive categories, and permissive categories map to permissive categories. Additionally, permissive categories within a single type never map to permissive security categories of different types.

For this presentation we also add an additional typing restriction on translation functions for permissive categories.

- Two permissive categories of different types do not map to permissive categories of the same type.

In some sense this is an extension of the constraint that translation function should respect the typing on categories.

Because the translation functions are defined on each of the parts, we overload the meaning of the functions as follows:

If x is one of the parts of a level l, f(x) means the corresponding part of the level f(l). This is unambiguous for any security level translation function. We also introduce the convention that if a is a category then f(a) is short for f({a}), i.e., the set of categories to which the set {a} is mapped by f.

4.2 Analysis of *SLTP* and military message labels

In this section, we show how the property *SLTP* can be interpreted for the military message labels defined above. Since the translation functions are defined independently on each part of a label, it suffices to examine the requirements on the translation functions for each part.

4.2.1 Definition of *Military-label-SLTP*

We define a property *military-label-SLTP* for a pair of functions f and f ' on military message labels, and then we will show that this property is equivalent to *SLTP* on military message labels.

As in the presentation of *SLTP*, the definition for *military-label-SLTP* is given as a pair of conditions. The conditions are the same property but applied to the forward and reverse mapping directions. Each condition is split into two parts, one for the hierarchical part and one for the categories.

Condition 1:

A. Hierarchical levels

If l1 is a hierarchical level in the domain of f, and l2 is a hierarchical level in the domain of f ' such that f(l1) = l2, then l1 = f '(l2).

B. Categories

It is convenient to handle a special case separately. This is the situation where the hypotheses for the first (forward) condition of *SLTP* are never met, because of the hierarchical levels.

Vacuous case:

There does not exist a hierarchical level l1 in the domain of f, and a hierarchical level l2 in the domain of f ', with f(l1) = l2.

If the special case holds, then *military-label-SLTP* Condition 1 puts no constraint on either restrictive or permissive categories.

Non-vacuous case:

Let l1 and l2 be levels such that l1 is in the domain of f, l2 is in the domain of f ', and f(l1) = l2. Then the restrictions for categories are as follows:

B1. Restrictive categories

Let a be a restrictive category in the domain of f. If the set f(a) is in the domain of f ', then the set f '(f(a)) contains a.

B2. Permissive categories

Let c be a permissive category in the domain of f. For every permissive category d in f(c), either d is not in the domain of f ', or f '(d)={c}.

Condition 2:
Same as Condition 1, but applied in the reverse direction.

Less formally, the non-vacuous conditions for restrictive and permissive categories are as follows:

If a restrictive category a in one domain maps to one or more categories in the other domain, and all of those categories map back to some categories in the first domain, then at least one of them maps back to a set that contains a.

If a permissive category c in one domain maps to one or more permissive categories in the other domain, then each of those categories may only map back to c (if they map back at all).

Note that if the vacuous cases do not hold, then the *military-label-SLTP* permissive category requirement says that if c is a permissive category in the domain of f, and some element of f(c) is

in the domain of f ', then f(c) is of size 1. To see this, suppose d is in f(c) and in the domain of f '. By the permissive category requirement from Condition 1, f '(d)={c}. By the permissive requirement for Condition 2, f(c)=f(f '(d))={d}.

4.2.2 Equivalence of *SLTP* and *military-label-SLTP*

Claim*:*
SLTP iff *military-label-SLTP* for translation functions on military levels.

Proof:
Let f and f ' be translation functions on military levels. We show the equivalence for condition 1. Condition 2 follows similarly.

=>: First, we show that *SLTP* implies *military-label-SLTP*.

We analyze each of the parts.

A. For hierarchical levels:

Suppose l1 is a hierarchical level in the domain of f, and l2 is a hierarchical level in the domain of f ' such that f(l1) = l2. Since the image of f is determined by the images on its parts, f(l1,{},{}) = (l2,{},{}), and (l2,{},{}) is in the domain of f ', *SLTP* says that (l1,{},{}) = f '(l2,({},{}). Hence, by definition of the partial order on the set of levels, we see that l1 = f '(l2).

B. For categories:

Vacuous case:

There is nothing to show for either restrictive or permissive categories, as *military-label-SLTP* holds vacuously.

Non-vacuous case:

Since the special condition does not hold, let l1 and l2 be levels such that l1 is in the domain of f, l2 is in the domain of f ', and f(l1) = l2.

B1. Restrictive categories:

Suppose a is a restrictive category in the domain of f, and where f(a) is in the domain of f '. We must show that f '(f(a)) contains a. By hypothesis, (l2,f(a) ,{}) is in the domain of f '. So, by *SLTP*, we find that (l1,{a},{}) = f '(l2 ,f(a),{}). By the definition of the partial order, we get that a = f '(f(a)) and hence we conclude that
f '(f(a)) contains a.

B2. Permissive categories:

Suppose c is a permissive category in the domain of f. Suppose d is in f(c) and d is in the domain of f '. We want to show that f '(d)={c}. Since f(c) contains {d} and {d} is non-empty, we have that f(c) = {d} (remember that the ordering for non-empty subsets of permissive categories is the reverse of subset inclusion). Hence f(l1,{},{c}) = (l2,{},{d}). Then, by *SLTP*, we have

(l1,{},{c}) = f '(l2,{},{d}). In particular, {c}= f '(d). By the definition of the partial order, f '(d) is non-empty and is contained in {c}. Hence we must have f '(d)={c}.

<=: We now show *military-label-SLTP* implies *SLTP*.

Suppose (h1, A, C) is in the domain of f, (h2, B, D) is in the domain of f ', and f (h1, A, C) = (h2, B, D). We want to show (h1, A, C) = f '(h2, B, D). To do this, we show that h1 = f '(h2), A = f '(B), and C = f '(D).

First, we note that h1 is in the domain of f, h2 is in the domain of f ' and f(h1) = h2. Hence, by the hierarchical condition of *military-label-SLTP* we conclude that h1 = f '(h2). This proves *SLTP* for hierarchical levels.

By definition of the partial ordering, we know that f(A) is contained in B. So, for each element a of A, f({a}) is contained in B and hence is in the domain of f '. Thus f '(f({a}) contains a by *military-label-SLTP*. Also, since f({a}) is contained in B, and the elementwise definition of translation functio ns, f '(f({a}) is contained in f '(B). So, {a} is contained in f '(B) for each a in A, and thus A is contained in f ' (B). Therefore, A = f '(B). This proves *SLTP* for restrictive categories.

Finally, pick any d in D. The set f(C) contains D by the definition of the partial order. Hence there is some c in C such that d is in f(c). By hypothesis, D is in the domain of f ', hence d is also in the domain of f ', since f ' is defined elementwise. So, by *military-label-SLTP,* we have f '(d)={c}. Since for each d in D, we can find a c in C with f '(d) = c, we conclude that f '(D) is contained in C.

However, to obtain C = f '(D), we must also show that for every category type P_i, if $C \cap P_i$ is non-empty then $f'(D) \cap P_i$ is also non-empty. To show this, suppose $C \cap P_i$ is non-empty. Let T_i be the permissive category type of $f(P_i)$. (T_i is well-defined by the constraints on allowable translation fucntions.) By hypothesis, there is an element of type T_i in f(C). Since f(C) = D, from the definition of the inequality, we find that $D \cap T_i$ is not empty. Pick an element d in $D \cap T_i$. Since d is in D, which is inside f(C), we know that d = f(x) for some x in C. Since D is also in the domain of f ', by *military-label-SLTP,* we conclude that f '(d) = x. Now x maps to an element of type T_i and so does any element of type P_i. As our constraint on allowable translation functions does not allow two different permissive category types to map to the same permissive category type, x must have type P_i. So, x is in $f'(D) \cap P_i$ and hence this set is non-empty. Thus, for every type P_i, if
$C \cap P_i$ is non-empty, then $f'(D) \cap P_i$ is also non-empty and is contained in $C \cap P_i$. Thus
C = f '(D).

Putting these together, we conclude that when f (h1, A, C) = (h2, B, D), we have that (h1, A, C) = f '(h2, B, D), which shows condition 1 of *SLTP*. Condition 2 follows similarly.
?

4.3 Order compatibility for military message level functions

Although order compatibility is not required for non-disclosure, it is a desired property. Here we describe what it means for military message levels

Proposition: For SDN.801 military message levels and translation functions, order compatibility holds if and only if it holds for hierarchical levels.

Proof:

=>
If order compatibility holds, then it clearly holds for the hierarchical parts.

<=
For the restrictive category part of a level, order compatibility follows from the definition of the ordering. To see this suppose x and y are sets of restrictive categories in the domain of f and $x < y$. By definition of <, every restrictive category in x appears in y. By the structure of the mapping f, the restrictive categories of f(y) are the union of the restrictive categories f(a) where a is in y. Thus the restrictive categories of f(y) clearly contain those of f(x).

For permissive categories, a similar argument applies, except that we must be more careful about handling the empty set cases. Suppose x and y are sets of permissive categories in the domain of f and $x < y$. We need to show that $f(x) = f(y)$. In particular, we show that for every type P, either $f(x) \cap P$ is empty, or $f(x) \cap P$ contains $f(y) \cap P$ and $f(y) \cap P$ is not empty. Suppose $f(x) \cap P$ is non-empty. Then f(x) contains f(y) by the elementwise construction of the translation function. Hence, $f(x) \cap P$ contains $f(y) \cap P$. Now let a be some element in x, such that f(x) is of type P. Say a is of type T. Since $x < y$, there must be some b in y of type T. By the assumption that all elements of T map to the same type, f(b) is of type P. Hence $f(y) \cap P$ is not empty.
?

Therefore, to establish order compatibility, we only need to check it on the hierarchical parts of levels.

5 Conclusion

The property *SLTP* provides a strong and conveniently checked requirement that a pair of partial functions must satisfy in order to be secure translations. Given two domains and translation functions between them, we prove that *SLTP* is equivalent to the existence of some comparison domain in which both of the domains embed, such that the functions are level increasing. In other words, *SLTP* holds exactly when there is some interpretation of the levels of the two domains for which the given translation functions have the proper non-disclosure property. Therefore, checking *SLTP* is the strongest possible check on the translations that involves only the partially ordered sets of levels and the translation functions.

Bibliography

[BLP] Bell, D. E., and LaPadula, L. J., "Secure Computer Systems: Mathematical Foundations and Model," MITRE Corp., Technical report M74-244, 1973.

[MISSI KPCMP] MISSI Key, Privilege, and Certificate Management Working Group Plan, MISSI Key, Privilege and Certificate Management Working Group, Draft 4.15, Feb. 27, 199

[MISSI SDN.706] X.509 Certificate and Certificate Revocation List Profiles and Certification Path Processing Rules for MISSI, Revision 3.0, May 30, 1997.

[MISSI SDN.801] MISSI Access Control Concept and Mechanisms, MISSI Key, Privilege, and Certificate Management Working Group, Draft 3.0, May 30, 1997.

[ORA] Rosenthal, D., and Fung, F., "Cross-Certification Modeling and Analysis," ORA Technical Report 97-0049, April 1998.

[TCSEC] Department of Defense Trusted Computer System Evaluation Criteria, DOD 5200-28-STD, Dec. 1985.

[X.509] ITU-T Recommendation X.509, Data Networks and Open Systems Communications Directory, Draft, June 1997.