A Lattice of Information

Jaisook Landauer Timothy Redmond
Trusted Information Systems
3060 Washington Road (Route 97)
Glenwood, MD 21738

1 Abstract

This paper presents a framework for describing information and information flow. We show that information can be represented as a lattice. We will motivate the idea that this framework is applicable for demonstrating security properties of systems. In particular, we show the relationship between the lattice representing information and the unwinding theorem. We will also demonstrate the relationship between properties of this lattice and the aggregation problem.

2 Introduction

Consider a system as a black box that allows users to query its internal state. For example, an airline database can be queried for different kinds of information, e.g., estimated time of arrival. The queries can be ordered by the amount of information returned. For example, if a query returns the complete flight information, then one can deduce the estimated time of arrival. In this example, the first query that requests ETA is "less" than the query that requests the flight information.

Users may also be ordered by the type of information that they can access. For example, the president of an airline company may be able to make more detailed queries than a random customer. In particular, a customer may not be able to obtain the passenger manifest whereas this information is available to the president.

In this paper, we formalize the notion of information as a complete lattice. The queries described in the above example will define elements of this lattice. The information determined by a query, q_1, is greater than the information determined by a query, q_2, if the result of q_2 can be explicitly determined from the result of the query q_1. The information obtained by making two queries at once will be the join of the information obtained by making each of the queries individually.

We will then provide a necessary and sufficient condition for non-interference in terms of the information lattice. The condition is the existence of a sensitivity labelling of the information lattice in such a way that

- instructions with a high sensitivity label do not modify information with a low sensitivity label
- the flow of information in the system is from information with low sensitivity labels to information with high sensitivity labels.
- the output at a sensitivity level can be determined by the information at that sensitivity level.

We will also show a possible connection of some properties of the lattice with the aggregation problem. This may point to a way of dealing with the aggregation problem in an algebraic manner.

3 The Information Lattice

For the purposes of this section, we will fix a set Σ, representing a state space of a system. In this paper we will show how information about elements of the set Σ can be regarded as a lattice. This information lattice can be described in two equivalent manners. First, we can view the lattice as the set of equivalence relations on the set Σ. The equivalence classes represent sets of states that cannot be distinguished with the information being described. Second, we can view information in terms of functions from Σ. The functions represent the information that they extract from the state.

The information lattice is first defined in terms of equivalence relations. This definition is then restated in terms of functions in theorem 1. We will use either form as convenient. Where possible, we will describe constructions on the lattice using both representations.

Since this paper uses equivalence relations heavily, we need to make a convention on their use. If $\sim$ is an equivalence relation then we will always denote the equivalence of two elements σ_1 and σ_2 with the infix notation $(\sigma_1 \sim \sigma_2)$. This is standard practice, but in this document we make special mention of this notation because we will not always use the standard symbols for equivalence relations. For example, if the term $||f||$ represents an equivalence relation, then $\sigma_1 ||f|| \sigma_2$ means that σ_1 and σ_2 are equivalent via the equivalence relation $||f||$.

We will now construct of the lattice. We first define a set, $\mathcal{I}(\Sigma)$, to be the set of all equivalence relations on the set Σ. We will define an ordering on this set that makes it a complete lattice. The ordering on $\mathcal{I}(\Sigma)$ is defined as follows

$$\approx \leq \sim \;\leftrightarrow\; \forall \sigma_1, \sigma_2 \;\; (\sigma_1 \sim \sigma_2 \Rightarrow \sigma_1 \approx \sigma_2) \tag{1}$$

where $\approx$ and $\sim$ are elements of the set $\mathcal{I}$.

For example, suppose that the state space can be partitioned in to two components, an unclassified component and a classified component. The state space can be represented as a cartesian product $\Sigma = U \times C$, where U is the unclassified component and C is the classified component. We introduce two equivalence relationships $\sim$ and $\approx$ where

$$< u_1, c_1 > \sim < u_2, c_2 > \;\leftrightarrow\; u_1 = u_2$$

$$< u_1, c_1 > \approx < u_2, c_2 > \;\leftrightarrow\; (u_1 = u_2 \wedge c_1 = c_2)$$

The $\sim$ equivalence relationship represents the information that can be deduced by an unclassified user and the $\approx$ represents the information that can be deduced by a classified user. The unclassified user will be able to deduce a smaller amount of information than the classified user, which is represented by the inequality $\sim \leq \approx$.

We will now demonstrate why the ordering (1) makes the information set on Σ into a complete lattice. It is sufficient to show that for any set, $P \subseteq \mathcal{I}(\Sigma)$, there exists a least upper bound for that set [1, 2]. It follows from lattice theory that this is enough to guarantee that the information set is a lattice. It is not difficult to see that the least upper bound of the set P is the the equivalence relation, $\sim$, given by

$$\forall x, y \in \Sigma \;\; (x \sim y \leftrightarrow \forall \approx \in P \;\; x \approx y)$$

We will now make the link from the above definition to the representation of the lattice $\mathcal{I}(\Sigma)$ in terms of functions whose domain is Σ. The construction works as follows: for any function, $f : \Sigma \to X$, we will define $||f||$ to be the element of $\mathcal{I}(\Sigma)$ for which

$$\forall \sigma, \sigma' \in \Sigma \;\; (\sigma \; ||f|| \; \sigma' \;\leftrightarrow\; f(\sigma) = f(\sigma'))$$

The following theorem shows how the lattice relationships can be defined using functions.

Theorem 1 *For any set Σ, the following properties hold:*

- *any element of $\mathcal{I}(\Sigma)$ can be represented as $||f||$ for some set, X, and some function $f : \Sigma \to X$.*
- *$||f|| = ||g||$ iff there exists a set isomorphism, ϕ, from the range of f to the range of g such that $g = \phi \circ f$.*
- *$||g|| \leq ||f||$ iff there exists a function, ϕ, such that $g = \phi \circ f$.*
- *if $f : \Sigma \to X$ and $g : \Sigma \to Y$ then*

  $$||f|| \vee ||g|| = ||h||$$

  where $h : \Sigma \to X \times Y$ is defined by

  $$\forall \sigma \in \Sigma \quad h(\sigma) = (f(\sigma), g(\sigma))$$

4 Some Lattice Properties

In this section, we will describe some of the mathematical properties of the information lattice. Where possible, we will indicate the relevance of the properties to practical applications.

The most basic property of the lattice is the manner in which a function $f : \Sigma_1 \to \Sigma_2$ induces a function $f_\# : \mathcal{I}(\Sigma_2) \to \mathcal{I}(\Sigma_1)$. The function $f_\#$ can be defined by the equation

$$f_\#(||g||) = ||g \circ f||$$

Equivalently, if $\sim \in \mathcal{I}(\Sigma_2)$, then $f_\#(\sim)$ is the equivalence relation given by

$$x \quad f_\#(\sim) \quad y \qquad \leftrightarrow \qquad f(x) \quad \sim \quad f(y)$$

An important property of this induced function is that for $f : \Sigma_1 \to \Sigma_2$ and $g : \Sigma_2 \to \Sigma_3$, we have,

$$f_\# \circ g_\# = (g \circ f)_\#$$

Also if $id : \Sigma_1 \to \Sigma_1$ is the identity map, then $id_\#$ denotes the identity map on $\mathcal{I}(\Sigma_1)$.

This structure can be embedding into a category theory framework. $\mathcal{I}$ is a contravariant functor from the category of sets to the category of ordered sets[1].

The practical significance of the induced function $f_{\#}$ is that it provides a formalism for determining the source of updated information after a state change. To elaborate, we formalize the notion of state change. Let $R : \Sigma \longrightarrow \Sigma$ be a transition function. Let $f : \Sigma \longrightarrow X$ be a view of the state space. If σ is the state before the transition, then the value of f after the transition is $f \circ R(\sigma)$. Thus the information in f after the transition can be determined from knowing the information in

$$\|f \circ R\| = R_{\#}(\|f\|)$$

before the transition.

A second important concept is the notion of a function leaving certain information invariant. If $R : \Sigma \longrightarrow \Sigma$ is a function then we define $\mathbf{fix}(R)$ to be the greatest element of $\mathcal{I}(\Sigma)$ such that

$$\forall \sigma \in \Sigma \;\; \sigma \quad \mathbf{fix}(R) \quad R(\sigma)$$

The equivalence relation $\mathbf{fix}(R)$ can be formed by constructing the reflexive transitive closure of the symmetric relation that identifies σ and $R(\sigma)$ for all $\sigma \in \Sigma$.

This is important for expressing a requirement that a high process does not write down. If $\sim \in \mathcal{I}(\Sigma)$ represents information with a low sensitivity label and R represents a transition that is being executed by a high process, then we require that the high transition leave the low information invariant. Using the above notation this can be expressed as $\sim \leq \mathbf{fix}(R)$. This will represent one of the requirements of the unwinding theorem presented in section 5.

Next we will present a result that will be important later in this paper.

Theorem 2 *If the cardinality of Σ is greater or equal to three then the lattice $\mathcal{I}(\Sigma)$ is non-distributive.*

Proof: Let $\Sigma_1 = \{a, b, c\}$. There are three non-trivial elements of $\mathcal{I}(\Sigma_1)$ given by

$$y \quad i_x \quad z \;\leftrightarrow\; ((y \neq x \wedge z \neq x) \vee (y = z))$$

It is easy to show that

$$i_a = i_a \cap (i_b \cup i_c) \neq ((i_a \cap i_b) \cup (i_a \cap i_c)) = 0$$

If $f : \Sigma \longrightarrow \Sigma_1$ is any onto function then we also have

$$\begin{aligned} f_{\#}(i_a) &= f_{\#}(i_a) \cap (f_{\#}(i_b) \cup f_{\#}(i_c)) \\ &\neq ((f_{\#}(i_a) \cap f_{\#}(i_b)) \\ &\quad \cup (f_{\#}(i_a) \cap f_{\#}(i_c))) \\ &= 0 \end{aligned}$$

QED

Finally we define the notion of independence. Two items of information are independent if the value of one has no impact on the value of the other. This is formalized as follows:

$$\sim \perp \approx \;\leftrightarrow\; \forall \sigma_1, \sigma_2 \exists \sigma_3 \;\; \sigma_3 \sim \sigma_1 \wedge \sigma_3 \approx \sigma_2$$

Equivalently this can be expressed in terms of functions as follows

$$\begin{aligned} \|f\| \perp \|g\| \quad \leftrightarrow \quad & \forall x \in \mathbf{range}(f), y \in \mathbf{range}(g) \\ & \exists \sigma \;\; f(\sigma) = x \wedge g(\sigma) = y \end{aligned}$$

The independence of two elements, $\sim$ and $\approx$, of $\mathcal{I}(\Sigma)$ is a stronger concept than $\sim \cap \approx = 0$. The fact that

$$\sim \perp \approx \;\rightarrow\; \sim \cap \approx = 0$$

is easy to show. The fact that independence is strictly stronger can be shown by considering i_a and i_b from the proof of theorem 2. In practice the notion of independence is more useful than the notion that two items of information have no common information.

5 Non-interference and Information Flow

In this section we will describe a variant of the unwinding theorem using the information lattice. This theorem is very closely related to the Abstract SAT MLS Unwinding Theorem of Haigh and Young [3]. This unwinding theorem will provide a necessary and sufficient condition for the non-interference of a state machine.

We will suppose the existence of a distributive lattice, L, representing sensitivity levels. We will suppose that we have a state machine consisting of an initial state, $\sigma_0 \in \Sigma$, a transition function

$$R : \Sigma \times I \longrightarrow \Sigma$$

and output functions $o_\lambda : \Sigma \longrightarrow O_L$ for each sensitivity level $\lambda \in L$. We will assume also that the set of inputs I is partitioned into disjoint sets, I_λ, where $\lambda \in L$.

The transition function, R, can be used to define a function

$$R^\star : \Sigma \times I^\star \longrightarrow \Sigma$$

where $I^\star$ is the set of sequences of elements of I as follows:

$$R^\star(\sigma, ()) = \sigma$$

[1] It is not a contravariant functor from the category of sets to the category of lattices.

$$R^\star(\sigma,(i_0,\ldots,i_{n+1})) = R(R^\star(\sigma,(i_0,\ldots,i_n)),i_{n+1})$$

For each sensitivity level $\lambda \in L$ we will form a purge function, $p_\lambda : I^\star \rightarrow I^\star$, that takes a sequence of elements of I and returns the sequence formed by removing all the elements not in some $I_{\lambda'}$ where $\lambda' \leq \lambda$.

The non-interference property states that for all sensitivity levels, $\lambda \in L$, and all input sequences $(i_0,\ldots,i_n) \in I^\star$, we have

$$o_\lambda(R^\star(\sigma_0,(i_0,\ldots,i_n)) = o_\lambda(R^\star(\sigma_0,p_\lambda(i_0,\ldots,i_n))$$

Theorem 3 Haigh-Young Unwinding *Suppose that all states are reachable and let $R_i(\sigma) = R(\sigma,i)$ for all $\sigma \in \Sigma$ and $i \in I$. The non-interference property is satisfied if and only if there exists a function, $lvl : L \rightarrow \mathcal{I}(\Sigma)$ such that*

- *(Information flows up) For all $i \in I$*

  $$(R_i)_\#(lvl(\lambda)) \leq \bigcup_{\lambda' \leq \lambda} lvl(\lambda')$$

- *(Processes only write up) For all λ, λ' such that λ' is not greater than λ, and all $i \in I_\lambda$,*

  $$lvl(\lambda') \leq \mathbf{fix}((R_i)_\#)$$

- *(Output is determined by the information at a level) For all λ,*

  $$\|o_\lambda\| \leq lvl(\lambda)$$

Proof: We will start with the if direction.

We first claim that

$$\begin{gathered} o_\lambda(R^\star(\sigma_0,(i_0,\ldots,i_n)) \\ [\bigcup_{\lambda' \leq \lambda} lvl(\lambda')] \\ o_\lambda(R^\star(\sigma_0,p_\lambda(i_0,\ldots,i_n)) \end{gathered}$$

This follows by induction on the length of the sequence of inputs, $(i_0,\ldots,i_n)$.

We will divide the induction step into two cases: the case where the last instruction, i_n is purgeable and the case where the last instruction is not purgeable.

If the last instruction, i_n is purgeable, then there is a λ' with $i_n \in I_{\lambda'}$ and $\neg(\lambda' \leq \lambda)$. By the hypothesis that processes only write up, we have

$$(\bigcup_{\lambda'' \leq \lambda} lvl(\lambda'')) \leq \mathbf{fix}(R_{i_n})_\#$$

Using the definition of **fix**, we have

$$x \quad \bigcup_{\lambda'' \leq \lambda} lvl(\lambda'') \quad R_{i_n}(x)$$

This allows us to prove the induction claim from the induction hypothesis.

If the last instruction is not purgeable, then we use the inequality

$$\begin{aligned} (R_i)_\#(\bigcup_{\lambda' \leq \lambda} lvl(\lambda')) &= \bigcup_{\lambda' \leq \lambda} (R_i)_\#(lvl(\lambda')) \\ &\leq \bigcup_{\lambda' \leq \lambda} \bigcup_{\lambda'' \leq \lambda'} lvl(\lambda'') \\ &= \bigcup_{\lambda' \leq \lambda} lvl(\lambda') \end{aligned}$$

Using the definition of $\leq$ and $(R_i)_\#$ this means that

$$x \quad \bigcup_{\lambda' \leq \lambda} lvl(\lambda') \quad y \quad \rightarrow \quad R_i(x) \quad \bigcup_{\lambda' \leq \lambda} lvl(\lambda') \quad R_i(y)$$

This allows us to prove the induction claim from the induction hypothesis.

Now we prove the only if direction. We define $lvl(\lambda)$ to be the equivalence relation on Σ that equates two elements σ_1 and σ_2 if

$$o_\lambda \circ R^\star(\sigma_1,(i_0,\ldots,i_n)) = o_\lambda \circ R^\star(\sigma_2,(i_0,\ldots,i_n))$$

for all $(i_0,\ldots,i_n) \in I^\star$.

It follows vacuously that

$$(R_i)_\#(lvl(\lambda)) \leq lvl(\lambda) \leq \bigcup_{\lambda' \leq \lambda} lvl(\lambda')$$

$$\|o_\lambda\| \leq lvl(\lambda)$$

Suppose that λ' is not greater than λ and $i \in I_\lambda$. This means that the instruction i is λ purgeable. By the non-interference condition,

$$R_i(\sigma) \quad lvl(\lambda') \quad \sigma$$

But this is exactly the condition that

$$lvl(\lambda') \leq \mathbf{fix}((R_i)_\#)$$

QED

6 The Aggregation Problem

In this section we will discuss the aggregation problem. We believe that one of the aspects of information that makes the aggregation problem hard is the fact that the information lattice fails to be distributive. We hope that the ideas of this section will allow us to deal with aggregation problem using algebraic tools.

The aggregation problem is concerned with the problem of labelling the sensitivity of information. For simplicity, we will suppose that we have two sensitivity labels, $\{Lo, Hi\}$. We have an aggregation problem when separate items of information that are labelled at Lo can be combined to make up information that implies information labelled at Hi.

It seems reasonable to consider a labelling of information to be a partial function

$$lvl : I(\Sigma) \rightarrow \{Lo, Hi\}$$

An aggregation problem will occur if there exists elements i_1, i_2, and i_3 in $\mathcal{I}(\Sigma)$ such that

- $i_3 \leq i_1 \cup i_2$
- $lvl(i_1) = Lo$
- $lvl(i_2) = Lo$
- $lvl(i_3) = Hi$

The first evidence that we will provide for the connection between the aggregation problem and the non-distributivity of the information lattice is the following theorem.

Theorem 4 *A finite lattice, L, is distributive if and only there exists a set $I \subseteq L$ such that*

$$\forall i \in I \quad i \leq x \cup y \ \rightarrow \ [(i \leq x) \vee (i \leq y)]$$

$$\forall x \in L \quad x = \bigcup_{i \in I, i \leq x} i$$

This theorem follows easily from some of the results in [2].

If such a set I could be found for $\mathcal{I}(\Sigma)$, then we could use it to solve the aggregation problem. The elements of I could be regarded as atomic bits of information. Any item of information can be regarded as being precisely the conjunction of some bits of atomic information. The atomic bits of information contained in the conjunction of two items of information will consist of exactly the atomic bits of information contained in the conjuncts and nothing more.

It would therefore be reasonable to solve the aggregation problem by starting out by labelling the atomic bits of information. The label associated with any information can therefore be considered as the max of the information associated with the contained atomic information. The property that

$$\forall i \in I \quad i \leq x \cup y \ \rightarrow \ [(i \leq x) \vee (i \leq y)]$$

would guarantee that the information formed by joining two items of information contains only those atomic items of information that are contained in one or the other items of information. This would ensure that the label associated with the join of two items of information would be the join of the labels associated with the two items of information.

However, the lattice $\mathcal{I}(\Sigma)$ fails to have any subset like I when Σ has three elements or more. In fact, this lattice is pretty bad in this respect. It is possible to find elements i_1, i_2, and i_3 such that

$$i_3 \leq (i_1 \cup i_2)$$

$$i_3 \cap i_1 = 0$$

$$i_3 \cap i_2 = 0$$

That is i_3 can be deduced from the conjunction of i_1 and i_2, but it has nothing in common with either.

Example:

Suppose that a particular database holds information about the salaries of employees of a company. Suppose that information in this database is labelled as follows. The salaries of the non-officers is marked as unclassified. The number of employees of the company is unclassified. Any information that is averaged over the whole company is marked as unclassified. Any information that is explicitly about the salaries of officers is marked as classified. This labelling leads to an aggregation problem.

For example, a user might ask the salaries of the non-officers, the average salary of the employees of the company and the number of employees of the company. From this the user can determine the average salary of the officers. The user can then start asking a more refined set of queries to gain additional information about the officers salary (e.g. the standard deviation of the salaries at the company).

In fact, if we suppose that the number of employees at the company is a constant then this example turns out to be a good example of the situation described above. Suppose that *noff* is the list of salaries of the non-officers, n is the number of employees of the company, av is the average salary at the company and oav

is the average salary of the officers. These variables are related by the following equation:

$$\mathbf{Sum}(\mathit{noff}) + (n - \mathbf{Len}(\mathit{noff})) * \mathit{oav} = n * \mathit{av}$$

It is not hard to show that the value of oav is independent of the value of noff ($||oav|| \perp ||\mathit{noff}||$). Similarly the value of oav is independent of the value of n ($||oav|| \perp ||n||$). Finally the value of oav is independent of the value of av ($||oav|| \perp ||av||$). However, once given the values of noff, n and av we can deduce the value of oav ($||oav|| \leq ||\mathit{noff}|| \cup ||n|| \cup ||av||$).

Note that this does not imply that any database which has at least three distinct states will have an aggregation problem. For any Σ with cardinality greater than three, the lattice $\mathcal{I}(\Sigma)$ will have many distributive sublattices. It is possible for a database to restrict the information that it provides to such a distributive lattice. Such a database would avoid the aggregation problem as described in this section.

7 Conclusions

In this paper we have shown how information can be viewed as a lattice. This provides us with an algebraic way of understanding critical concepts concerning information flow. For example, state transitions give rise to order preserving functions that describe the flow of information resulting from the state change. State transitions also give rise to the set of information that is left fixed by the information. These concepts can be used to define algebraic necessary and sufficient conditions for the non-interference property. The algebraic approach will provide a new set of tools with which we can tackle the non-interference problem. An example that utilizes this approach can be found in [6].

In addition, we have indicated a connection between the aggregation problem and some algebraic properties of the lattice of information. This allows us to use algebraic techniques to gain additional insight into the aggregation problem.

References

[1] Garrett Birkhoff. *Lattice Theory*, volume XXV of *American Mathematical Society Colloquim Publications*. American Mathematical Society, 1967.

[2] P. Crawley and R.P. Dilworth. *Algebraic Theory of Lattices*. Prentice Hall, 1973.

[3] Will Harkness. The General LOCK Model and Unwinding Theorem. R21 informal technical report, Department of Defense, October 1990.

[4] T. H. Hinke. Inference Aggregation Detection in Database Management Systems. In *Proceedings of the 1988 IEEE Symposium on Security and Privacy*. IEEE, April 1988.

[5] T. Lunt. Aggregation and inference: Facts and fallacies. In *IEEE Symposium on Security and Privacy*, pages 102–109, Oakland, CA, May 1989.

[6] Sylvan Pinsky. An Algebraic Approach To Non-Interference. In Robert Werner, editor, *Proceedings of the Computer Security Foundations Workshop V*, pages 34–47. IEEE Computer Society Press, June 1992.

[7] M. Weiser. Program Slicing. *IEEE Transactions on Software Engineering*, pages 352–357, July 1984.