Computational Issues in Secure Interoperation

Li Gong and Xiaolei Qian

Abstract— **Advances in distributed systems and networking technology have made interoperation not only feasible but also increasingly popular. We define the interoperation of secure systems and its security, and prove complexity and composability results on obtaining optimal and secure interoperation. Most problems are NP-complete even for systems with very simple access control structures, while for a general setting the problem is undecidable. Nevertheless, composability reduces complexity in that secure global interoperation can be obtained incrementally by composing secure local interoperation. We illustrate, through an application in secure database interoperation, how these theoretical results can help system designers in practice.**

Keywords— **Computer security, database systems, interoperability, secure composability, algorithms, computational complexity.**

I. Introduction

Recent advances in distributed systems and networking technology have made interoperation not only feasible but also increasingly popular. For example, heterogeneous databases can be linked by high-speed networks that consist of heterogeneous networks connected by gateways. In such an application environment, heterogeneity (such as in data semantics, data representation, and communication protocols) among system components must be reconciled properly. Some research efforts are under way to deal with these problems [1].

One attribute of interoperation that needs reconciliation but has not been closely studied is security with regard to access control. Consider an application involving multiple systems dealing with commerce (e.g., national credit databases), finance (e.g., stock market information systems), medicine (e.g., patient records), and defense, each having a distinct access control structure. To facilitate information exchange among such systems, some mapping between the heterogeneous security attributes must be introduced, for example, by the system administrators. Current practices show that these mappings, even if chosen carefully, can result in security breaches that previously did not exist in any individual system (e.g., [2], [3]).

Secure interoperation is a serious concern for military systems[1] as well as commercial ones. For example, consider the information system of a major research organization where Alice, being a project supervisor, is allowed access to Bob's files, but not vice versa. Suppose that this organization has just become a subsidiary of a corporation where Charles is Vice President for Research and Diana, being his secretary, has access to his files. After the merger, it seems natural to permit Charles to access Alice's project papers. But if Bob should be allowed access to Diana's file cabinet, there would be a security violation because now Bob would potentially have access (indirectly via Diana and Charles) to Alice's files to which he should be denied access.

Although the security violation in this example may not be too difficult to spot and remove, a real-world system could have hundreds or thousands of entries in its access control list so that choosing a secure yet satisfactory (e.g., with maximum data sharing) mapping between many such access control lists is a daunting task. In other words, interoperation of systems with heterogeneous access control structures poses the following new challenges: what is the definition of secure interoperation? How can security violations be detected? And how can these violations be removed while a maximum amount of information exchange is still facilitated? This paper attempts to answer some of these questions. First we turn to what we think are the fundamental requirements in secure interoperation.

II. Principles of Secure Interoperation

One essential feature in federated systems is the autonomy of an individual system – that is, each system may be administrated independently [4], [1]. To preserve this feature in secure interoperation, autonomy in security should be guaranteed.

> **Principle of Autonomy.** Any access permitted within an individual system must also be permitted under secure interoperation.

On the other hand, interoperation should not violate the security of an individual system.

> **Principle of Security.** Any access not permitted within an individual system must also be denied under secure interoperation.

Any other new access introduced by interoperation should be permitted unless explicitly denied by the specification of secure interoperation. Note that, unless stated otherwise, by access we mean direct or indirect access. Moreover, we assume that access rules in one system do not conflict with rules in another system. In practice, this assumption is generally satisfied by the fact that the set of one system's entities (e.g., users and files) is typically disjoint from that of another system. Extensions to our approach can handle situations when this assumption does not hold.

It is conceivable that under some circumstances a system may be willing to sacrifice some of its autonomy.

A preliminary version of this paper was presented at the IEEE Symposium on Research in Security and Privacy, Oakland, California, May 1994.

The authors are with SRI International, Computer Science Laboratory, 333 Ravenswood Avenue, Menlo Park, California 94025 U.S.A. Email addresses are {gong,qian}@csl.sri.com.

[1]The (U.S.) Defense Information Systems Agency's *Defense Information System Network Technology Requirements Document* (version 0, August 3, 1993) estimated that the U.S. DoD enterprise has more than 10,000 networks worldwide, most of which are not interoperable with each other and do not adequately support information sharing.

III. General Undecidability

In our discussion, the security attributes of a system are expressed with an access control list (ACL) [5]. We view a system as a collection of users, machines, data objects, and others, each being a distinct unit with regard to security.

The task we are facing is the following: given a set of access control lists, define what secure interoperation is and investigate the complexity of detecting security violations in the global system and that of removing security violations while maintaining a reasonable level of interoperation.

It has been previously shown that the security (or safety, as it is sometimes called in the literature) of any given set of access rights and commands is in general undecidable [6], and some variations of the decision problem are at best NP-complete [7]. Here, we also prove that the general secure interoperation problem is undecidable.

Informally, the security aspect of an interoperation is represented by access rights across systems. That is, given access control lists for individual systems, an interoperation F is a set of access control entries where, for each entry, the subject and the object belong to different systems. To satisfy the principle of security, the general problem is to decide if any access that is not permitted in one system is permitted as a result of interoperation.

We formulate our problem following the general definitions by Harrison, Ruzzo, and Ullman [6, Theorem 2, p.469]. A system in the general HRU model consists of a set of access rights and a set of commands. For brevity, we do not repeat the details here, except by noting that the set of access rights – subjects' actions on objects – include **create, delete, enter right, remove right**, and others. Therefore, informally, an interoperation consists of two individual systems and an additional set of commands that refer to symbols (subjects, objects, and access rights) in both systems.

Problem 1: **(General Secure Interoperation)** Given the access control lists and commands of two systems G_1 and G_2, an interoperation F, and an access right r in G_1. Is there a command sequence that will add r to an entry in G_1, where r previously does not exist and cannot be added by commands in G_1 alone?

Theorem 1: General secure interoperation is undecidable.

Proof: Given a safety problem, as defined by Harrison, Ruzzo, and Ullman, of the form: "Can access right r appear in entry (i, j)?" we can permute the lines and columns of the access control matrix so that entry (i, j) resides at the upper-left corner of the matrix.

We also draw a horizontal line and a vertical one to divide the access matrix into four sections, where the upper left section containing the single entry of (i, j) represents G_1, the lower right section represents G_2, and the other two sections represent the interoperation F (see Figure 1 where $F_1 \cup F_2 = F$).

We then accordingly reassign the commands in the original system as follows. Those commands that do not refer to symbols in G_1 are assigned to belong to system G_2. All other commands belong to F. Note that, if there is such a command that refers to only symbols in G_1, we can easily rewrite them so they now refer to symbols in both G_1 and G_2 and yet their functionalities remain unchanged.

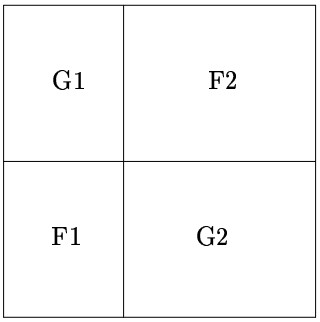


Fig. 1. Proof of General Undecidability

Clearly r does not exist in G_1, and cannot be added by commands in G_1 alone because there is no command in G_1 at all. Now, if the general secure interoperation problem is decidable, then we can decide whether r can be entered into G_1. Thus we can decide if r can be entered in entry (i, j) in the safety problem. In other words, the general safety problem is also decidable, which is a contradiction. ■

IV. System Model and Terminology

Given Theorem 1, we expect at best to obtain NP-completeness results for secure interoperation of systems with more restricted access control structures. We thus follow the usual proof method for NP-completeness to investigate only a restricted problem where in each ACL: (1) each subject owns exactly one file, with read and write access; (2) a subject can have only read access to a file owned by someone else; (3) if a subject can read another's file, the latter cannot read the former's file; (4) an ACL is static, and in particular, read and write are the only types of access specified.

Our NP-completeness results should imply similar NP-completeness results for formations of the problem using more general access control lists. In addition, given the particular restrictions on ACL, our results should also imply NP-completeness results for the interoperation of Bell-LaPadula type multilevel secure systems [8], [9], although our study is not specially aimed at multilevel security either in the sense of Bell-LaPadula or that of noninterference (e.g., [10]).

In our discussion, we use the following terminology, notations, and definitions. Because one subject owns exactly one file, there is no need to distinguish between a subject and its file. For example, instead of saying that Alice has access to Bob's file, we can simply say that Alice has access to Bob. We refer to this combination of a subject and its file an entity. Moreover, it is obvious that one entity has access to oneself (i.e., one's own file), and if Alice can access Bob, and Bob can access Charles, then Alice can access Charles indirectly. Recall that one restriction on the

ACL is that if Alice can access Bob then Bob cannot access Alice, we arrive at the following definition of a secure system as specified with a restricted ACL.

Definition 1: **(Secure System)** A secure system is an ACL in the form of $G = \langle V, A \rangle$ where V is a set of entities and A is a binary relation "access" on V that is reflexive, transitive, and antisymmetric.

Graphically, we can view a system as an acyclic directed graph. V is the set of vertices and A is the set of arcs – there is an arc leading from vertex u to v, denoted by (u, v), if and only if A contains the binary relation "u access v". The direction of the arc is then the direction of the permitted access. Thus, for the merger example, we have `Research = <{Alice, Bob, Eve}, {(Alice, Bob), (Eve, Alice)}>`, and `Corporation =<{Charles, Diana, Fred}, {(Charles, Fred), (Diana, Charles)}>`. The graphical representation of both systems is in Figure 2, where each person is represented by his or her initial.

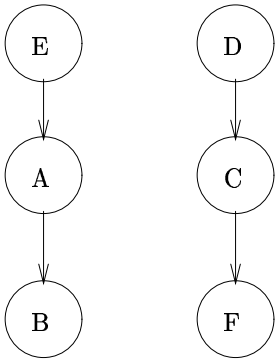


Fig. 2. Two Separate Systems

For convenience, we sometimes do not distinguish between an ACL and its graphical representation if no confusion can arise.

We say that an access (u, v) is *legal* in G (or in A) if and only if there is a directed path in (the graphical representation of) G leading from u to v. We denote this with $(u, v) \propto G$.

Suppose we have n secure systems, $G_i = \langle V_i, A_i \rangle$, $i = 1, 2, \ldots, n$, and for simplicity, we assume that all entities are distinctly named – that is, $V_i \cap V_j = \emptyset, i \neq j$. (Here $\emptyset$ denotes the empty set.) To facilitate interoperation, mappings between entities of different systems must be introduced to reflect the desired data sharing through interoperation. Such mappings can be represented by a set of cross-system access relations F, which is chosen possibly by an administrator with global security responsibility or by a select committee in charge of the individual systems.

Definition 2: **(Permitted Access)** Permitted access is a binary relation F on $\cup_{i=1}^{n} V_i$ where $\forall (u, v) \in F$, $u \in V_i$, $v \in V_j$, and $i \neq j$.

The fact that $(u, v) \in F$ indicates that it is thought that entity u (in system G_i) should be allowed to access entity v (in system G_j). Note that it is possible to have both $(u, v) \in F$ and $(v, u) \in F$.

In our example, suppose that it is decided that interoperation should allow Bob to access Fred (i.e., his file) and Charles to access Alice. Then the global system is in Figure 3 where arcs belonging to F are represented as dotted lines.

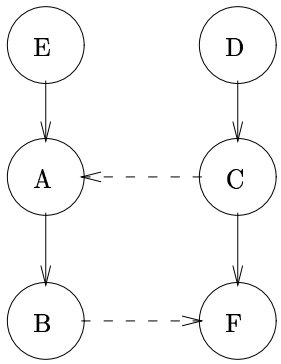


Fig. 3. Interoperation of Two Systems

The interoperation may also mandate a set of restricted access R, as follows.

Definition 3: **(Restricted Access)** Restricted access is a binary relation R on $\cup_{i=1}^{n} V_i$ such that $\forall (u, v) \in R$, $u \in V_i$, $v \in V_j$, and $i \neq j$.

This is similar to a negative entry in an access control list [11]. The purpose is to explicitly safeguard certain parts of the system when the potential implications of introducing F are unclear. In our example, we may forbid access `(Diana, Eve)`. R takes precedence over F.

To give the definition of secure interoperation for a federated system $Q = \langle V', A' \rangle$ consisting of the n subsystems, where $V' = \cup_{i=1}^{n} V_i$ and $A' = (\cup_{i=1}^{n} A_i \cup F) - R$, recall that the autonomy principle requires that a legal access in A_i remain legal in A', i.e., if $(u, v) \propto A_i$ then $(u, v) \propto A'$. On the other hand, the security principle requires that an illegal access in A_i remain illegal in the interoperation, i.e., if $(u, v) \not\propto A_i$ then $(u, v) \not\propto A'$. In addition, all access in R should be explicitly restricted – that is, $A' \cap R = \emptyset$.

Definition 4: **(Secure Interoperation)** Given $G_i = \langle V_i, A_i \rangle, n = 1, \ldots, n$. $Q = \langle \cup_{i=1}^{n} V_i, B \rangle$ is a secure interoperation if $B \cap R = \emptyset$, and $\forall u, v \in V_i$, $(u, v) \propto A_i$ if and only if $(u, v) \propto B$.

F and R may contradict each other, and other security violations can also occur as a result of interoperation. For example, with a different F, as illustrated in Figure 4, Bob can access Alice indirectly through Diana, even though this access is illegal within the research organization.

In situations like this, F may need to be changed or reduced to remove security violations (recall that R takes precedence over F). Thus, given $G_i, i = 1, \ldots, n$, F, and R, our aim is to find a federated system $Q = \langle W, B \rangle$, where $W = \cup_{i=1}^{n} V_i$ and $B \subseteq (\cup_{i=1}^{n} A_i \cup F) - R$, such that Q is a secure interoperation.

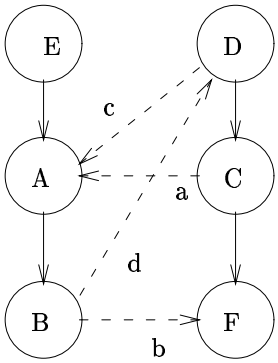


Fig. 4. Security Violation Caused by Interoperation

V. Complexity

For convenient discussion, we mark all arcs belonging to $G_i, i = 1, \ldots, n$, green, mark all arcs in the permitted access set F purple, and mark all arcs in the restricted access set R red.

The first problem we encounter is to decide if a given interoperation is secure.

Problem 2: **(Security Evaluation)** Given $G_i = \langle V_i, A_i \rangle, i = 1, \ldots, n$, permitted access F, and restricted access R. Is $\langle \cup_{i=1}^{n} V_i, (\cup_{i=1}^{n} A_i \cup F) - R \rangle$ a secure interoperation?

Theorem 2: Security evaluation is in P.

Proof: We prove the theorem by giving a polynomial-time algorithm to detect security violations. Let A_i^+ denote the transitive closure of A_i, and B^+ denote the transitive closure of $B = (\cup_{i=1}^{n} A_i \cup F) - R$. The algorithm is as follows.

First, obviously $B \cap R$ is an empty set. Then, compute B^+ and $A_i^+, i = 1, \ldots, n$, and check that B^+ induced by (i.e., restricted to) V_i is a subset of A_i^+. If any checking fails, report security violation; otherwise, report secure. The correctness of the algorithm is obvious, noting that the definition of B automatically satisfies the autonomy requirement.

The complexity of the algorithm is the complexity of calculating the transitive closures $O(| \cup_{i=1}^{n} V_i|^3)$ plus the complexity of the comparisons $O(| \cup_{i=1}^{n} V_i|^3)$, so an upper bound is $O(| \cup_{i=1}^{n} V_i|^3)$. ■

If $B = (\cup_{i=1}^{n} A_i \cup F) - R$ is insecure, we can remove the security violations by reducing F until the resulting interoperation is secure. In other words, we can look for $S \subseteq F$ such that $C = (\cup_{i=1}^{n} A_i \cup S) - R$ is secure. This is trivial because $S = \emptyset$ is definitely a secure solution.

To find nontrivial secure solutions, one choice is to look for a secure solution that includes all other secure solutions. In other words, find $S \subseteq F$ such that $C = (\cup_{i=1}^{n} A_i \cup S) - R$ is secure and, for any secure solution T, $T \subseteq S$. Unfortunately, such solutions do not always exist, as is shown by the following counterexample.

Consider the interoperation of systems $G_1 = \langle \{a1, a2, a3\}, \{(a1, a2), (a2, a3)\} \rangle$ and $G_2 = \langle \{b1, b2, b3\}, \{(b1, b2), (b2, b3)\} \rangle$, as illustrated by Figure 5. Suppose $F = \{(b3, a2), (a3, b2)\}$, which obviously causes a security violation because access $(a3, a2)$ is legal in the federated system but illegal in G_1. One secure solution is $S_1 = \{(a3, b2)\}$. Another secure solution is $S_2 = \{(b3, a2)\}$. But any solution containing both S_1 and S_2 contains F, which causes a security violation.

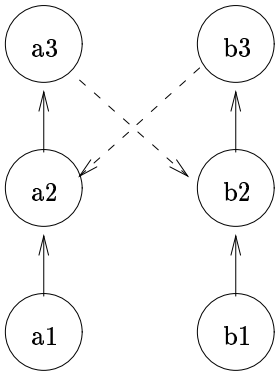


Fig. 5. All-Inclusive Solutions May Not Exist

An alternative in finding nontrivial secure solutions is to look for solutions that cannot be expanded any further. In other words, find a secure solution $S \subseteq F$ such that, for any secure solution T, $S \not\subseteq T$. This problem is in P, as the following polynomial-time algorithm demonstrates: start with an empty solution S; add elements in F to S one by one, and only if the addition will not cause a security violation (recall that security evaluation is in P); repeat this process until no more elements can be added. The correctness of this algorithm is obvious.

The three choices described so far do not give natural optimality measures. For example, a solution may turn out to contain just one arc from F although the exclusion of this single arc would allow the addition of two other arcs, with the latter intuitively facilitating more information exchange. Therefore, we propose two definitions that are more natural. From now on, we stipulate that $F \neq \emptyset$ because the secure interoperation problem disappears when $F = \emptyset$ (and thus $R = \emptyset$).

One natural optimality measure is to maximize direct data sharing. Take the interoperation represented in Figure 4, for example. Arcs a and d (or c and d) cause a security violation. To reduce a minimum number of arcs from F, it is better to remove d so that both a and c can be preserved.

Problem 3: **(Maximum Secure Interoperation)** For any positive integer $K \leq |F|$, is there a secure solution S such that $S \subseteq F$ and $|S| \geq K$?

Theorem 3: Maximum secure interoperation is NP-complete.

Proof: The problem belongs to NP because a nondeterministic machine can guess a solution at random and

verify its autonomy and security properties in polynomial time (refer to Theorem 2 on security evaluation).

The rest of the proof is to reduce a known NP-complete problem, the Feedback Arc Set problem [12, p.192], to a restricted case of our problem at hand. We first review the Feedback Arc Set problem:

> Given a directed graph $G = \langle V, A\rangle$, positive integer $K \leq |A|$. Is there a subset $A' \subseteq A$ with $|A'| \leq K$ such that A' contains at least one arc from every directed cycle in G?

The restricted case of Problem 3 is when all individual systems are of the form $G_i = \langle\{u_i, v_i\}, \{(u_i, v_i)\}\rangle$, F contains no directed cycles, and $R = \emptyset$. Here, the only type of security violation is a directed cycle (in the federated system) containing a green arc (u_i, v_i), because access from v_i to u_i would become possible via such a cycle. Moreover, any cycle must contain at least a green arc because there are no red arcs and no all-purple cycles.

Our reduction, shown in Figure 6, is as follows. Given any $G = \langle V, A\rangle$, we define $G' = \langle V', A'\rangle$ as follows. V' is formed by splitting every vertex u in V into a pair of vertices u_1 and u_2. For each pair of such vertices, A' has an arc (u_1, u_2). Let $\langle\{u_1, u_2\}, \{(u_1, u_2)\}\rangle$ denote an individual system. For every arc in A that ends at u, there is a corresponding arc in A' that ends at u_1, and for every arc in A that departs from u, there is a corresponding arc in A' that departs from u_2. Let $F = A'$. Clearly F does not contain any cycle.

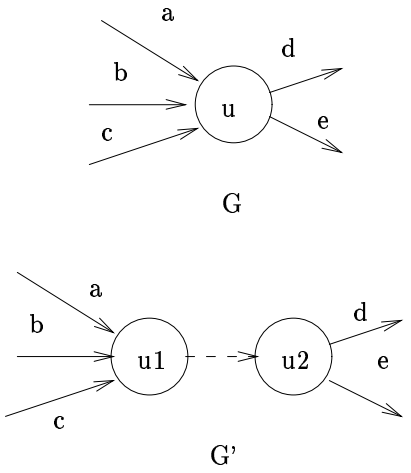


Fig. 6. Reduction

Next we show that this reduction is a one-to-one mapping between the two problems in that A'' is a solution for the Feedback Arc Set problem in G, with $|A''| \leq K$, if and only if $S = (A' - A'')$ is a solution of maximum secure interoperation in G', with $|S| \geq |A'| - K$.

Suppose A'' is a solution for the Feedback Arc Set problem in G, then $A'' \subseteq A'$ and $|A''| \leq K$. Let $S = A' - A''$. S does not contain directed cycles in G, thus it does not contain directed cycles in G' either because there is exactly one arc connecting each pair of splitted vertices. Therefore S does not cause a security violation. Moreover, since $A'' \subseteq A'$, we have $|S| = |A' - A''| = |A'| - |A''| \geq |A'| - K$. Thus, S is a solution to maximum secure interoperation in G'.

On the other hand, suppose S is a solution to maximum secure interoperation in G'. Since S does not contain directed cycles, $A'' = A' - S$ must contain at least one arc from each directed cycle in A'. Because $S \subseteq A'$ and $|S| \geq |A'| - K$, $|A''| = |A' - S| = |A'| - |S| \leq |A'| - (|A'| - K) = K$. Therefore, A'' is a solution to the Feedback Arc Set problem in G. ■

For an NP-complete problem, one naturally seeks good approximation algorithms. We now prove that finding certain approximate solutions is also NP-complete. Given a federated system G, we use $X(G)$ to denote a solution obtained by an approximation algorithm, of size $|X(G)|$, and $OPT(G)$ to denote the optimal solution, of size $|OPT(G)|$.

Corollary 1: If P $\neq$ NP, then no polynomial-time algorithm $X(G)$ for the maximum secure interoperation problem can guarantee $|OPT(G)| - |X(G)| \leq K$ for a fixed constant K.

Proof: Suppose to the contrary that X is indeed such an approximation algorithm with guarantees. We show that X can be used to construct a polynomial-time algorithm Y that solves the maximum secure interoperation problem, which contradicts the assumption that P $\neq$ NP.

Given G and a positive integer K, our algorithm Y constructs G' that consists of $K + 1$ isomorphic copies of G where the copies are not connected to each other. (The theorem is not more difficult to prove when connectivity is required.) It is easy to see that $|OPT(G')| = (K + 1) \times |OPT(G)|$. Furthermore, we can construct a solution for G with a size of at least $|X(G')|/(K+1)$ merely by taking the isomorphic copy of G that has the largest solution among the $(K + 1)$ copies. Thus, $(K + 1) \times |Y(G)| \geq |X(G')|$.

Because X guarantees that $|OPT(G')| - |X(G')| \leq K$, we have $|X(G')| \geq |OPT(G')| - K = (K+1) \times |OPT(G)| - K$. Thus, $(K + 1) \times |Y(G)| \geq |X(G')| \geq (K + 1) \times |OPT(G)| - K$. This is $(K+1) \times (|OPT(G)| - |Y(G)|) \leq K$, or $|OPT(G)| - |Y(G)| \leq K/(K + 1)$. This means $|Y(G)| = |OPT(G)|$ (because solutions must be integers) and thus Y is a polynomial-time algorithm for the maximum secure interoperation problem, a contradiction. ■

So far we have been working to find maximum subsets of F that result in secure interoperation, and Theorem 3 and its corollary suggest that this is hard.

Another natural measure of optimality is to maximize direct and indirect information sharing by working on the whole federated system. The aim is to find a secure interoperation with a maximum number of legal access, instead of looking for a secure solution F of a maximum size. That is, we change F as long as the new F does not introduce an access that is illegal under the initial set F.

Take the interoperation represented in Figure 4 again, for example. Arcs a and d (or c and d) cause a security violation. Previously, for a solution with maximum size, it was better to remove d so that both a and c could be preserved. Now to obtain maximum access, it is actually

better to remove both a and c to preserve d because the latter facilitates more (albeit indirect) information sharing.

Problem 4: **(Maximum-Access Secure Interoperation)** For any positive integer $K \leq |(\cup_{i=1}^{n} A_i \cup F)^+ - R|$, is there a secure interoperation $\langle \cup_{i=1}^{n} V_i, B \rangle$ such that $B \subseteq (\cup_{i=1}^{n} A_i \cup F)^+ - R$ and $|B| \geq K$?

Theorem 4: Maximum-access secure interoperation is NP-complete.

Proof: The problem obviously belongs to NP because a nondeterministic machine can guess a solution at random and verify its suitability in polynomial time (recall Theorem 2 that security evaluation is in P).

Again, we reduce the Feedback Arc Set problem to a subproblem when each individual system is of the form $G_i = \langle \{u_i, v_i\}, \{(u_i, v_i)\} \rangle$. Given any $G = \langle V, A \rangle$, our reduction is identical to that in the proof of Theorem 3, as shown in Figure 6. We compute the transitive closure of the new graph G' and call it $G'' = \langle V', (A')^+ \rangle$.

We aim to prove that A'' is a solution to the Feedback Arc Set problem in G if and only if $B = A' - A''$ forms a secure interoperation in G'', with $|B| \geq |A'| - K$.

In the set of arcs introduced by computing the transitive closure, namely $((A')^+ - A')$, if an arc is within one single system G_i, then it must be of the form (u_2, u_1), because (u_1, u_2) is already in G_i. Therefore, a new arc in G_i will cause a security violation and cannot be present in any secure interoperation. Define R to be the subset of $(A')^+ - A'$ that contains all arcs connecting two different systems. This definition of R effectively removes from any secure interoperation all arcs added when computing the transitive closure, thus the subproblem becomes the Maximum Secure Interoperation problem, and the rest of this proof is essentially the same as that of Theorem 3. ■

Corollary 2: If P $\neq$ NP, then no polynomial-time algorithm $X(G)$ for the maximum-access secure interoperation problem can guarantee $|OPT(G)| - |X(G)| \leq K$ for a fixed constant K.

Proof: Similar to the proof of Corollary 1. ■

The above results show that the problems we are investigating are mostly NP-complete. Nevertheless, we have found a special case where finding an optimal solution takes only polynomial time.

Problem 5: **(Simplified Maximum-Access Secure Interoperation)** Suppose that every G_i consists of a single directed path and graph $\langle \cup_{i=1}^{n} V_i, F \rangle$ is acyclic. For any positive integer $K \leq |(\cup_{i=1}^{n} A_i \cup F)^+|$, is there a secure interoperation $\langle \cup_{i=1}^{n} V_i, B \rangle$ such that $B \subseteq (\cup_{i=1}^{n} A_i \cup F)^+$ and $|B| \geq K$?

Theorem 5: Simplified maximum-access secure interoperation is in P.

Proof: We prove by constructing a polynomial-time algorithm to find the optimal solution. As before, we mark arcs in A_i green and arcs in F purple. We mark all the other arcs in $(\cup_{i=1}^{n} A_i \cup F)^+$ yellow. Since G_i consists of a single directed path and F does not contain directed cycles, a security violation occurs if and only if there is a directed cycle in the transitive closure $G^+ = \langle \cup_{i=1}^{n} V_i, (\cup_{i=1}^{n} A_i \cup F)^+ \rangle$. Thus, our objective is to find the maximum acyclic subgraph of G^+ that also contains all the green arcs (to preserve autonomy). In other words, we want to remove a minimum number of arcs in order to remove all cycles.

Note that in a transitive closure, the subgraph induced by all vertices on a cycle is a complete graph – where each pair of vertices is connected by arcs in both directions – so G^+ can be viewed as a collection of complete graphs plus arcs between them. These in-between arcs do not introduce cycles, thus any maximum interoperation must include them. Therefore, our task is reduced to finding in a complete graph the maximum subgraph that does not have cycles (since the number of such complete graphs in G^+ is polynomial).

By induction, we can easily prove a lemma that for a complete graph of m vertices, the maximum acyclic subgraph (denoted as $G(k)$) contains exactly $m(m-1)/2$ arcs. This is obviously true when $m = 2$ since $m(m-1)/2 = 1$. Suppose the lemma is true for $m = k$. For $m = k+1$, we argue that, after adding one more vertex to $G(k)$, we can add exactly k arcs to form $G(k+1)$ without introducing cycles. First, we can add k arcs without introducing cycles: each new arc departs from an existing vertex and arrives at the new vertex. Second, if we add $k+1$ arcs, then since $G(k)$ contains only k vertices, there are at least two arcs connecting the new vertex and an existing vertex. These two arcs are necessarily in opposite directions and therefore form a cycle. Therefore, $|G(k+1)| = |G(k)| + k = k(k-1)/2 + k = (k+1)k/2$.

Given the above lemma, we can arrange the vertices in a left-to-right line such that vertices in V_1 are grouped together first, from left to right in descending order, so that a vertex can always "access" the one on its right side. Then vertices in V_2 are similarly lined up, and so on. Under such an arrangement, all green arcs are in the direction of left to right. Therefore, we only need to delete all arcs pointing from right to left, which are either purple or yellow, and we have found a maximum acyclic subgraph that contains all the green arcs. The whole process is clearly in polynomial time. ■

Next we turn to another related problem. Suppose that the initial interoperation is already secure, or that an approximate or optimal solution has been found. Here the set F may contain some arcs that are redundant in the sense that data sharing provided by them is already provided by other permitted access. Therefore, it is quite natural to consider reducing the size of F as much as possible.

Problem 6: **(Minimum Representation)** For any positive integer $K \leq |F|$, is there a subset $F' \subseteq F$ such that $|F'| \leq K$ and that the set of legal access remains unchanged when F is replaced by F'?

Theorem 6: Minimum representation is NP-complete.

Proof: The subproblem when all $A_i, i = 1, \ldots, n$, and R are empty sets is identical to the known NP-complete problem of Minimum Equivalent Digraph [12, p.198]. ■

This result implies that, unless P=NP, any polynomial-time algorithm for finding a secure interoperation cannot guarantee to result in a minimum representation.

Nevertheless, if we remove the constraint that reduction

can take place only within F – that is, we ask if there is a $F' \subseteq \cup_{i=1,j=1,i\neq j}^{n} V_i \times V_j$ such that $|F'| \leq K$ and that the set of legal access remains unchanged when F is replaced by F' – then the problem is equivalent to Transitive Reduction [12, p.198], which is solvable in polynomial time. This type of reduction may be useful in a preprocessing step to reduce the problem space of any algorithm used subsequently. However, such a measure means that the setup of individual systems may be changed, which may not be desirable for other reasons.

VI. Composability

To reduce the complexity of finding maximum secure interoperation, one area for exploration is the topology of system interoperation. In some federated systems, for example, interoperation is accomplished by having a master system interacting with other systems in local interoperation [1]. We now prove that in such a configuration, the global interoperation is secure if and only if each local interoperation is secure.

Given systems $G_i = \langle V_i, A_i \rangle, i = 0, 1, \ldots, n$, where G_0 is the master system, let $G_{0-i} = \langle G_0, G_i, F_i \rangle$ denote the local interoperation between G_0 and G_i with permitted access set F_i, $i = 1, \ldots, n$. The global system is thus $G' = \langle \cup_{i=0}^{n} V_i, (\cup_{i=0}^{n} A_i) \cup (\cup_{i=1}^{n} F_i) \rangle$.

Problem 7: **(Federated Secure Interoperation)** Given secure G_{0-i}, $i = 1, \ldots, n$. Is G' secure?

Theorem 7: G' is secure if and only if G_{0-i} is secure, $i = 1, \ldots, n$.

Proof: Any local interoperation of a secure global interoperation is automatically secure. Thus we need only to show that the security of all the local interoperation guarantees the security of the global federated system. By two case studies, we show that, in the federated system, there cannot be a legal access that is illegal in either the master system G_0 or a satellite system G_i.

Suppose there is an access (a, b) that is legal in the federation but is illegal in the master system G_0. Because any local interoperation is secure, the chain of access from a to b involves G_0 and at least two other systems. As depicted in Figure 7, there must be an access chain from a in G_0 to outside, which reenters G_0 at some vertex c, goes out to G_i, and reenters G_0 again leading to b.

Clearly access (a, c) must be illegal in G_0 because otherwise access (a, b) would be legal in G_{0-i}, which contradicts the assumption that G_{0-i} is a secure local interoperation. But apparently (a, c) is legal in the federated system excluding G_i, thus after excluding G_i, the rest of the federation must still be insecure. By induction we can see that this implies that there exists an insecure local interoperation G_{0-j}, a contradiction.

Similarly, suppose there is a access (a, b) that is legal in the federation but is illegal in some satellite system $G_i, i \neq 0$. Again, the chain of access from a to b involves G_0, G_i, and at least another system. As depicted in Figure 8, there must be an access chain from a in G_i to c in G_0, which eventually leaves G_0 at some vertex d to enter G_i at vertex b.

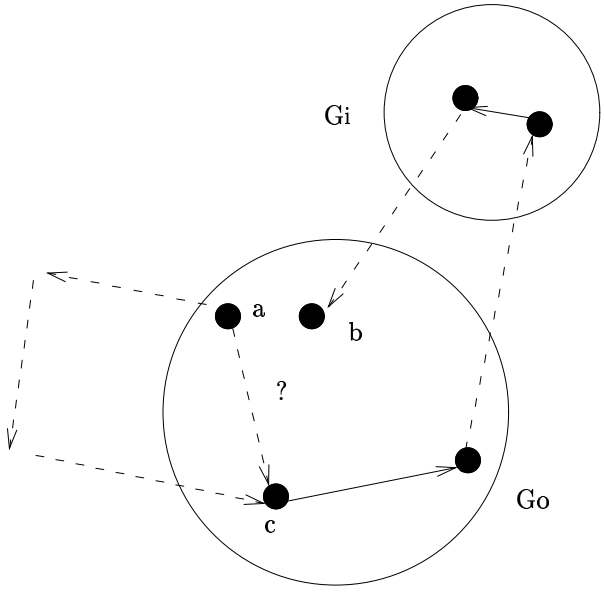


Fig. 7. Security Violation in G_0

Clearly access (c, d) must be illegal in G_0 because otherwise access (a, b) would be legal in G_{0-i}, which contradicts the assumption that G_{0-i} is a secure local interoperation. But apparently (c, d) is legal in the federated system excluding G_i. Thus, after excluding G_i, the rest of the federation must still be insecure. By induction we can see that this implies that there exists an insecure local interoperation G_{0-j}, a contradiction. ■

The above theorem implies that local secure interoperation, and thus local maximization, can be computed independently and in parallel.

Corollary 3: **(Maximum Federated Secure Interoperation)** G' is a maximum secure interoperation if and only if G_{0-i} is a maximum secure interoperation, $i = 1, \ldots, n$.

The two very positive results above indicate that in a star-like configuration, global (maximum) secure interoperation can be achieved in a distributed fashion, locally, and *incrementally* as more systems join the interoperation. We can thus say that (maximum) secure interoperation is *composable*. Note that these results do not necessarily imply that maximum-access secure interoperation is composable.

The proofs in Theorem 7 clearly extend to any configuration of a tree structure in that if all local interoperation between neighboring systems are secure (and maximum), then the global interoperation is also secure (and maximum).

Corollary 4: **(General Federated Secure Interoperation)** Secure interoperation and maximum secure interoperation are composable in any tree-structure configuration.

In a ring-structure configuration (or any configuration containing a ring), the composability theorem does not always hold. A simple counterexample is when each F_i con-

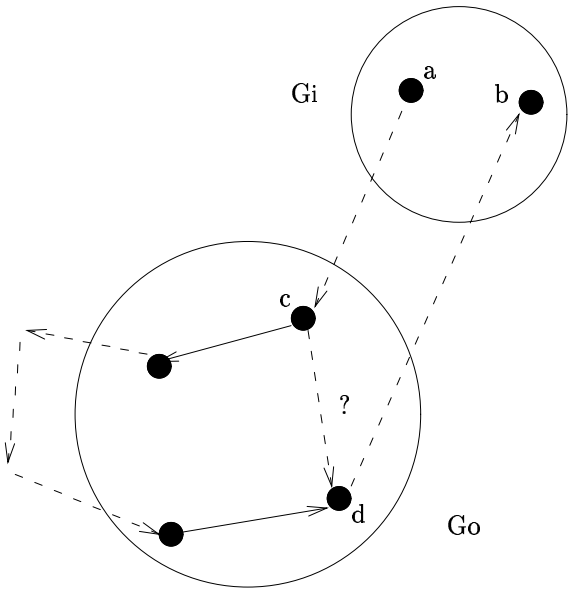


Fig. 8. Security Violation outside G_0

tains only one arc; thus, each local interoperation is secure, but the collection of these plus a green arc forms a cycle and permits an illegal access. The implication is that secure interoperation can be joined together as long as no ring is formed.

From the proof details, we expect that the above composability results generalize beyond the simple access control structure we have assumed in our current discussion.

VII. An Application in Databases

An obvious application area is the interoperation of heterogeneous databases. As more secure databases are built and connected through computer networks, a wide variety of secure data sources is becoming accessible. One of the biggest challenges presented by this technology is the secure interoperation of databases containing data with mismatched access control structures [13]. Providing secure interoperation not only makes it possible to reliably share data in isolated military and civilian databases, but also increases users' confidence and willingness in such sharing.

A key requirement in the interoperation of heterogeneous and especially legacy databases is autonomy [1]. Since these databases were often independently designed to each serve the needs of a single organization, and significant investment has already been made into them, the interoperation must respect their autonomy. Our definition of secure interoperation properly captures the autonomy requirement in security.

The interoperation of secure databases presents new requirements. While the concern in the interoperation of databases with homogeneous access control structures is how to maximize data sharing between databases, such maximization in the interoperation of databases with heterogeneous access control structures has to be tempered by security considerations. In other words, the data sharing caused by database interoperation should not compromise the security of individual databases. This requirement is also properly captured by our definition of secure interoperation.

Applying our complexity analysis to the autonomous and secure interoperation of heterogeneous databases with mismatched access control structures, we can see that the detection of security breaches in interoperation is easy, and the hard problem is the elimination of security breaches while maximizing data sharing. Although the general problem is not tractable, our results provide useful guidelines in solving this problem in practice. Below are some example guidelines.

- Although the general problem is NP-complete, the most common case in multilevel secure databases, where access control structures form total orders, is polynomial-time solvable, as is shown by Theorem 5.
- Although solving the general problem involves examining globally all the interoperating databases and links between them, for the widely adopted case of federated database systems [1], in which all data sharing is carried out through the federated schema, the problem can be solved in a pairwise manner, as is shown by Theorem 7. This implies that the problem can be solved *incrementally* as new databases join the federation.

In addition, the interoperation of secure databases suggests other natural optimality measures, whose computational complexity we are studying now. For example, we might want to maximize the number of databases interoperable, thus an optimal solution might link as many databases as possible, even if the amount of data sharing is not necessarily maximized.

VIII. Related Work

Secure interoperation can in some sense also be viewed as composing secure systems. A number of composition methods have been proposed for building a large system out of secure components (e.g., [14]). These previous results are mostly focused on composing systems with identical or compatible security attributes or policies, and tend to treat the avoidance of covert channels as the most important requirement. We deal with secure interoperation of systems with heterogeneous security attributes, and the composition method we examine is a very natural one that has been used frequently in practice.

Another related work is a study of interoperation of multilevel secure databases [15], where the problem is security label translation. Like us, these researchers recognize that naive interoperation may cause security violations. They define a notion called relation consistency and propose a label insertion algorithm to achieve that. But unlike us, they do not provide any complexity or composability result.

A canonical security model was proposed for federated databases [13], where the main concern is the integration of heterogeneous security policies and the specification of security constraints in a federated schema. However, the

problem of detecting and eliminating security breaches in a federated schema is not considered.

IX. Summary and Future Work

We have studied the problem of secure interoperation of systems with heterogeneous access control structures. We formed the definition of secure interoperation on the following basic notions: *autonomy*, which dictates that legal access in one system should remain legal in the global system, and *security*, which says that illegal access within one system should remain illegal in the global system. We proved that, while the security of a general interoperation is undecidable, finding a secure solution with some optimality is NP-complete even for a very simple type of access control list. Thus, finding similar optimal solutions for more general access control lists can only be harder. Nevertheless, composability reduces complexity in that secure global interoperation can be obtained incrementally by composing secure local interoperation. These results, as shown by the database application discussed, can help steer system design effort to searching for approximation algorithms and partial optimization, for example, by using heuristic algorithms.

For future work, one direction is to improve the theoretical results. This includes obtaining results on the hardness of obtaining percentage-wise approximation solutions, where some recent work [16] may be helpful, and investigating other optimality measurements that are applicable to particular environments. We have so far assumed that R represents direct access that are undesirable, such as a negative entry in an access control list. This means that an indirect access may still be possible, as in the case of a typical discretionary access control scheme. If we interpret restricted access as banning both direct and indirect access, then similar theorems might be obtained. For example, Theorem 2 trivially holds. Theorem 3 (and its corollary) should also hold because its proof is about the subcase when $R = \emptyset$. Developing near-optimal algorithms, possibly probabilistic algorithms, to obtain good average-case performance is also desirable.

Another direction is to examine ways to distribute the process of removing security violations from a central control point to individual systems, for example, by defining interfaces that preserve security. This is analogous to the development of distributed concurrency control. We can also explore various topologies of system interoperation, as in Theorem 7. Another possibility is to divide the overall task of finding maximum secure interoperation into preprocessing and run-time processing, because the latter on average probably does not involve a large number of separate systems. This idea of delaying the decision to run time can have other benefits. For example, given two permitted access that together will violate security, instead of deciding a priori (and somewhat arbitrarily) to remove one of them, we can decide to keep the one that is used first during run time. This is similar to the Chinese Wall policy (e.g., [17]) where one access will automatically prohibit future access of another kind, but which access to prohibit is not decided in advance.

Acknowledgment

Our colleagues at SRI, Pat Lincoln, Teresa Lunt, and Peter Neumann provided valuable comments on earlier versions of this paper. We are grateful to John McLean of the Naval Research Laboratory for spotting a serious technical error introduced in a more recent draft.

This work was supported in part by the U.S. Department of Defense Advanced Research Projects Agency and U.S. Air Force Rome Laboratory under Contract F30602-92-C-0140.

References

[1] A. Sheth and J. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", *ACM Computing Surveys*, vol. 22, no. 3, pp. 183–236, September 1990.

[2] U.S. National Computer Security Center, *Trusted Network Interpretation*, July 1987, NCSC-TG-005 version-1.

[3] NCSC, *Trusted Network Interpretation Environments Guideline*, (U.S.) National Computer Security Center, August 1990, NCSC-TG-011 version-1.

[4] J.A. Bull, L. Gong, and K.R. Sollins, "Towards Security in an Open Systems Federation", in *Proceedings of the European Symposium on Research in Computer Security*, Toulouse, France, November 1992, vol. 648 of *Lecture Notes in Computer Science*, pp. 3–20, Springer-Verlag.

[5] B.W. Lampson, "Protection", in *Proceedings of the 5th Princeton Symposium on Information Sciences and Systems*, Princeton University, March 1971, Reprinted in ACM Operating Systems Review, 8(1):18–24, January, 1974.

[6] M.A. Harrison, W.L. Ruzzo, and J.D. Ullman, "Protection in Operating Systems", *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, August 1976.

[7] R.S. Sandhu, "The Typed Access Matrix Model", in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, California, May 1992, pp. 122–136.

[8] D.E. Bell and L.J. La Padula, "Secure Computer System: Unified Exposition and Multics Interpretation", Tech. Rep. ESD-TR-75-306, The MITRE Corporation, Bedford, Massachusetts, March 1976.

[9] C.E. Landwehr, "Formal Models for Computer Security", *ACM Computing Survey*, vol. 13, no. 3, pp. 247–278, September 1981.

[10] J.A. Goguen and J. Meseguer, "Security Polices and Security Models", in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, April 1982, pp. 11–20.

[11] M. Satyanarayanan, "Integrating Security in a Large Distributed System", *ACM Transactions on Computer System*, vol. 7, no. 3, pp. 247–280, August 1989.

[12] M.R. Garey and D.S. Johnson, *Computers and Intractability*, W.H. Freeman and Co., New York, 1979, Paperback edition 1991.

[13] G. Pernul, "Canonical Security Modeling for Federated Databases", in *Proceedings of the IFIP TC2/WG2.6 Conference on Semantics of Interoperable Database Systems*, November 1992.

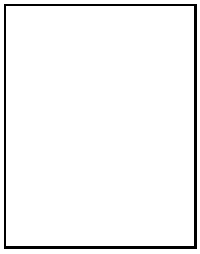
[14] D. McCullough, "A Hookup Theorem for Multilevel Security", *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 563–568, June 1990.

[15] V.E. Jones and M. Winslett, "Secure Database Interoperation via Role Translation", in *Security for Object-Oriented Systems*, B. Thuraisingham, R. Sandhu, and T. C. Ting, Eds. Springer-Verlag, London, 1994, A previous version appeared as Technical Report, Department of Computer Science, University of Illinois at Urbana-Champaign, April 1993.

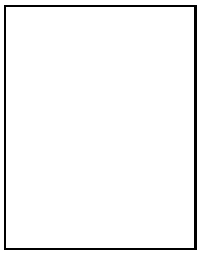
[16] S. Arora, G. Lund, R. Motwani, M. Sudan, and M. Szegedy, "Proof Verification and Hardness of Approximation Problems", in *Proceedings of the IEEE 33rd Annual Symposium on Foundations of Computer Science*, Pittsburgh, Pennsylvania, October 1992, pp. 14–23.

[17] D.F.C. Brewer and M.J. Nash, "The Chinese Wall Security Policy", in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, April 1989, pp. 206–214.

[18] L. Gong and X. Qian, "The Complexity and Composability of Secure Interoperation", in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, California, May 1994, pp. 190–200.



Li Gong was born in Beijing, China, and was educated at Tsinghua University, Beijing (B.E. with honors in 1985 and M.S. in 1987), and the University of Cambridge, England (Jesus College, Ph.D. in 1990). He is a Computer Scientist at SRI, researching in distributed systems and communication networks, particularly in issues of fault tolerance and security. He is Program Co-Chair of the Third ACM Conference on Computer and Communications Security (1996), and served as Program Chair of the 7th and 8th IEEE Computer Security Foundations Workshops (1994 and 1995) and as program committee member of various ACM, IEEE, and IFIP conferences. He is also on the editorial board of the *Journal of Computer Security*. He received the IEEE Communications Society Leonard G. Abraham Prize Paper Award in 1994 and the IEEE Symposium on Security and Privacy Outstanding Paper Award in 1989.



Xiaolei Qian received the B.Sc. degree from Xian Jiao Tong University, Xian, China, in 1982, and the M.Sc. and Ph.D. degrees from Stanford University, Stanford, California, in 1984 and 1989, respectively, all in computer science.

She is a Senior Computer Scientist in the Computer Science Laboratory at SRI International. Her research interests include database security, semantic interoperation and integration of heterogeneous databases, and software architectures. She is also interested in constraint management, database programming languages, and formal methods.