To appear, *Proc.10th Annual Computer Security Applications Conference*, Orlando, FL, Dec. 1994

A Practical Approach to High Assurance Multilevel Secure Computing Service

J.N. Froscher*, M. Kang*, J.McDermott*,
O. Costich†, and C. E. Landwehr*
*Naval Research Laboratory, Code 5542
Washington, DC 20375-5337
†Independent Consultant
e-mail: {froscher | mkang | mcdermott | costich | landwehr}@itd.nrl.navy.mil

Abstract

Current projects aimed at providing MLS computing services rarely seem to exploit advances in related fields. Specifically, the concepts of data distribution, replication, and interoperation are currently receiving much attention in the commercial database system sector but have yet to be applied to the delivery of MLS computing services. This paper explains how these concepts might help deliver MLS computing services relatively quickly and cheaply, and how they can ease integration of legacy systems and new technology into future MLS cooperative, distributed computing environments.

Researchers, system developers, and system integrators have long sought ways to provide acceptable, affordable multilevel secure (MLS) computing services. By "multilevel secure computing service" we mean a single service that permits users with different clearances to have access to computerized data and programs for which they are authorized and prevents them from gaining access to those for which they aren't.

Although the history of efforts to provide MLS computing service is entering its third decade, current projects often ignore the lessons that history provides. A practical approach is needed that exploits evolving commercial developments, instead of competing with them.

This paper documents an approach that attempts to take computing history and trends into account. It arose in the context of MLS database systems; the architecture and algorithms it depends on have been documented in the database security literature. Recently, we have realized that this approach can be extended to deal with security problems posed by the integration of a wide variety of legacy systems (i.e., obsolescent systems still in operational use) into a cooperative, distributed information system. The following sections place our approach in historical perspective and explain both how it meshes with current commercial developments in database systems and how it can be used to engineer an MLS cooperative, distributed computing environment.

1. A Brief History of MLS Computing

In the 1960's and 1970's, computation was expensive. Computer systems tended to be large and monolithic; many interactive terminals might be attached to them, they might even have several CPUs and many disk and tape drives, but the basic architecture was centralized. The goal then was to build or shore up time-sharing operating systems so that they could provide users with different clearance levels access to files holding different levels of classified data without compromise. Probably the Multics system came closest to achieving this goal ([Orga72],[NCSC85]) but there were other efforts as well, including the earlier ADEPT-50 and later KVM-370 and Keykos [Land83] projects.

A principal motive for trying to provide MLS computing service at that time was the high cost of providing duplicate services at each security level, which was the main alternative. Not only were such duplicate systems expensive, they couldn't communicate with each other securely. Because in many cases such communication was essential, risky and awkward work-arounds such as "air-gaps" and "sneaker-nets" were tolerated. Centralized MLS computing service naturally seemed highly desirable, and it seemed only a matter of time before it would be achieved and MLS services would abound, yielding both lower costs and lower risks.

Toward the end of the 1970's and into the 1980's, computing power began to migrate away from the monolithic central processor and toward users' terminals. The computer security world shifted its attention from securing large scale time-sharing systems to securing a small-scale one: UNIX. As the machines providing

UNIX service moved from the computing center into the user's office, the goal for MLS computing service shifted to providing an MLS workstation to the user in his office that could support electronic mail at different security levels and access to MLS file services, probably remotely located. Attempts to build a "secure UNIX" probably began at UCLA [Popek79]. An early British effort [Rush83] proposed to produce an MLS UNIX computing service using untrusted hosts sharing a multilevel file server via trusted network interface units. By 1987 at least half a dozen projects were underway [NRL87], and similar ones have continued to the present, for example in the TMach, DTMach, and Synergy efforts [Bran89] [Fine93][Sayd94]. Developments during the 1980's underlined the cost of developing software to meet criteria for high assurance. IBM reported, for example, that 80% of the resources used to modify Xenix to meet TCSEC class B2 requirements went toward satisfying the assurance requirements; meeting the feature requirements required only 20% of the project resources[Glig87].

But the focus on MLS computing service via MLS operating systems remained; the goal was still to build an operating system that could be trusted to support applications running at distinct security levels. It was unclear how an application that itself needed to span security levels would be layered on such an operating system. Further, a significant part of the increased computing power available to each user was used to improve the human - computer interface: from a black and white "glass teletype", user interfaces advanced to full color window and icon formats.

It was easy for users to imagine a system that would provide them with windows at different security levels -- much easier than it was to secure the complex application-level software required to support such processing. None of the systems being developed for high TCSEC levels of assurance (B2 and up) could support this kind of interface. The Defense Intelligence Agency (DIA) defined requirements for the Compartmented Mode Workstation (CMW) in order to generate commercial implementations of this vision, albeit at a relatively low level of assurance. Perhaps DIA could accept this level because in their operational environment all users were highly cleared and violations of mandatory need-to-know controls resulting from potential system flaws could be dealt with relatively easily[1].

Database applications also attracted increasing attention during this period. Since the days of Multics, an MLS database service had been seen as highly desirable. In 1982, the Air Force sponsored a National Research Council study [MDMS83] that considered potential near, medium, and long term approaches to providing an MLS database service. Over the next decade, several of these were explored through research projects. The National Computer Security Center developed a "Trusted Database Interpretation" of the TCSEC toward the end of the 1980's [TDI]. In parallel with this effort, database vendors began to consider improving security in their systems.

Theorists of MLS computing service focused increasingly on the problem of developing realistic and accurate security models [McLe88] and on the problem of finding and reducing covert channels. In 1987, McCullough [McCu87] published initial work concerning the security properties that might (or might not) hold when individual MLS components were hooked together to form a larger system. This came to be known as the composability problem: how to identify a security property desired of individual components that would also hold for a system of such components properly hooked together [McLe89] [McLe94]. Finding a composable security property that is also of practical interest has proven quite difficult, and the increasing prevalence of systems that are patched together from a variety of components has made this problem seem urgent [Lubb93] [Land93]. NATO chartered a research study group (NATO RSG-2) to investigate the question, "How are the assurances associated with the trustworthiness of a composite system to be derived from the assurances associated with the subsystems?" and though it convened a workshop in the fall of 1991, results were inconclusive and the group has been disbanded. The Trusted Network Interpretation of the TCSEC [TNI87] attempted to provide some practical guidance to system composition, but left many problems unsolved.

From the late 1970's through the early 1990's, then, several trends were evident: hardware was cheap and getting cheaper, software (at least high assurance software) was increasingly expensive relative to hardware, and understanding the security properties of the interconnections of hardware and software that were beginning to populate operational systems was exceedingly difficult. However, most research aimed at providing MLS computing service seemed to ignore these trends, continuing to focus on building MLS operating systems on which single level applications could be layered.

During the same period, commercial information technology products had become much more affordable and much more capable. However, the cost of migrating legacy systems to newer technology was high [Aike94]. Business and government were reluctant to change these

[1] For an approach to providing labeled windows with high assurance, see [Epst92]

fragile systems upon which their survival depended and found it difficult to modernize their business processes to become more efficient and productive. These legacy systems came to be known as "stovepipes," reflecting their architecture, in which data flow in one end of the pipe, are transformed by an application, and emerge from the other.

In the early 1990's, client-server architectures began to dominate data management systems. To avoid the mistakes of the past, these architectures emphasized the separation of concerns between data management and application processing. The idea was to make data, at any level of granularity, available for any application to access and to allow the cooperative distribution of processing capabilities [Brod94]. Distributed information management servers provide access by information processing systems to the required data and support the sharing of data by users through these servers. To ensure the availability and reliability of data in a cooperative, distributed computing environment, information management technology recently has turned to replication.

A project that proposed to exploit declining hardware costs to avoid expensive, high assurance software and provide cost-effective, high assurance MLS database service was proposed in 1988 and commenced in 1990 [Fros89]. This project, now known as the Secure Information Through Replicated Architecture (SINTRA) project, uses physical separation and data replication to achieve high performance and high assurance at a practical cost. It requires only one high assurance component, a replica controller, and only a small amount of high assurance software must be written for that. Further, the approach taken in the SINTRA project for databases can in fact be applied to a much broader class of systems than conventional DBMSs. While it is not a panacea, the SINTRA approach offers a practical alternative for interconnecting legacy systems without introducing unwanted risks. It offers not only a new paradigm for providing MLS data access, but it exploits synergy between a strong MLS protection strategy and current trends in data management technology, namely the distribution of data management servers and the use of replication.

The remainder of the paper describes the SINTRA approach to MLS database service, how it can be applied to legacy systems, the composition principle it uses, the problems it solves, and the problems it doesn't solve.

2. The SINTRA Approach to MLS Database Service

The SINTRA approach to providing MLS database service is based on physical separation and data replication. Rushby and Randell [Rush83] noted four kinds of separation that could be exploited to provide security: physical, temporal, logical, and cryptographic. SINTRA relies on physical separation (of backend database systems) with coordination provided by a trusted front end (TFE) or replica controller (RC). The TFE, because it must be connected to systems operating at different security levels, cannot rely on physical separation, so it relies on logical separation. Conceivably, it could employ cryptographic or temporal separation as well. The approach was developed to provide multilevel relational database service, and the current SINTRA prototypes demonstrate this service. The approach and the prototype are documented in numerous papers and reports ([Fros89], [McDe91], [Cost92a], [Cost92b], [Kang92], [Kang93a], [Kang93b], [McDe93],[Cost94],[Kang94]). We briefly summarize two possible SINTRA DBMS configurations: the SINTRA MLS DBMS and distributed SINTRA MLS DB servers.For a SINTRA MLS DBMS (Fig. 1), a TFE is connected directly to two or more backend databases. Each backend stores data (e.g. tuples) at its own security level and replicas of data at all lower security levels that it receives from the TFE. Users at different clearance levels gain access to the database service via the TFE, which both identifies and authenticates users and coordinates the upward flow of replicas among the backends.

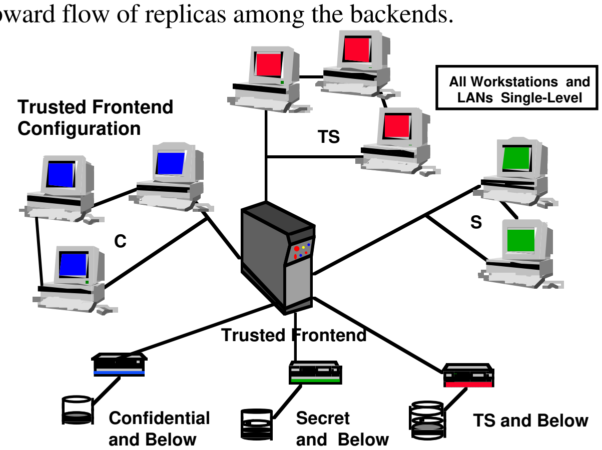


Fig. 1: SINTRA trusted front end configuration.

An authenticated user may initiate both retrievals and updates. Retrievals are directed to the backend corresponding to the user's authenticated security level and can be satisfied by that backend alone, since it contains all of the information that a user at that level is authorized to see. Updates requested by a user at this security level would be directed to the same backend, but they are also propagated to all backends operating at security levels that dominate the level of this one, so that the higher level backend databases remain consistent with the lower level ones. One of the fundamental contributions of the SINTRA project has been the development of algorithms to organize the propagation of updates so that the databases remain consistent without permitting a downward flow of information among the backends ([McDe91],[Cost92a],[Kang92]).

The second configuration (Fig. 2), distributed SINTRA, connects users directly to databases without interposing a TFE. A Replica Controller (RC) (essentially a TFE without identification/authentication and other user interface functions) connects the backend databases and propagates updates as in the first configuration. In this case, the backend databases may be responsible for passing on update operations to the replica controller.

Note that this configuration has much in common with current commercial architectures to support sharing among heterogeneous databases. Conventionally, when outside subscribers establish an agreement for routine sharing of certain information, the data provider agrees to send a copy and all updates of the data to the requester.

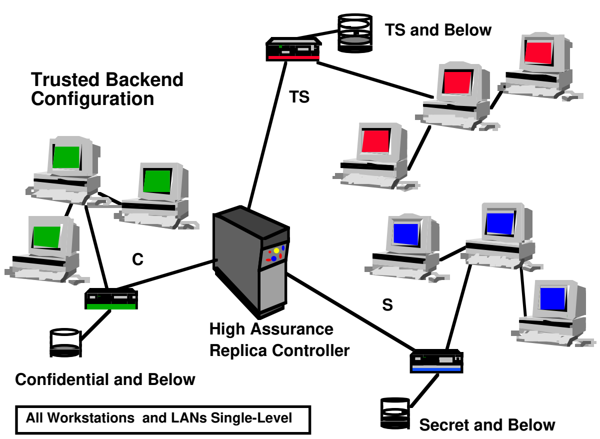


Fig. 2: SINTRA backend replica controller configuration.

The requester is responsible for managing and maintaining the copy for as long as access is required. Distributed SINTRA provides higher level users access to low data in much the same way. Not all low data must be replicated at higher levels -- only data that higher level users or processes acting on their behalf must read. In effect, the higher level users subscribe to data provided by the low level system. The RC handles all multilevel secure communication among single level DBMSs, which contain information at its level and below. If all DBMSs operate at the same security level, the high assurance SINTRA RC could be replaced by a low assurance commercial counterpart.

Distributed SINTRA resists malicious code attacks that could compromise information in other systems. A TCB in which physical resources are shared among processes at different security levels will inevitably contain covert channels. Malicious code inserted in an untrusted DBMS on such a system could exploit a variety of these channels to leak information. If such malicious code were inserted into one (or all) of the database systems in distributed SINTRA, however, it would be unable to exploit such channels, because the only physical component shared across security levels is the RC, and it is relied on only for the upward propagation of updates. Covert channels within the RC are a concern, but they cannot be directly exercised by software running on physically separate machines. Any indirect use of such channels would have to occur over the very limited update-propagation interface between the database systems and the RC, instead of via the direct, TCB system call interface that would typically be available to a database running on the same processor as the TCB. Thus, by physically separating the protection critical execution (on the RC) from the general purpose execution (on the databases), distributed SINTRA severely limits opportunities for exploiting vulnerabilities in the TCB on which the RC is built.

Variations on these configurations might differ in the precise RC and TFE functions required, the specific database functions relied upon for maintaining consistency, and the degree of continuous operation provided in the face of failures, either in the backend databases or the TFE/RC. The essentials of the approach are the use of physical separation and data replication to avoid relying on the database management systems to enforce confidentiality requirements.

3. The SINTRA Approach to Legacy Systems

Although the SINTRA approach was developed to provide a centralized MLS database service, the same approach can be generalized to provide MLS services for systems that are not conventional database systems. In fact, it can be applied to many existing application systems to provide users with different clearances access to their services without requiring major changes to those systems. As long as one can identify transactions for these legacy systems, the SINTRA approach can produce an MLS service.

As described above, there are two basic kinds of components in this approach: "system-high" components and replica controllers. "System high" is used in the conventional DoD sense [DoD 5200.28]-- a component operated so that all its users possess a security clearance or authorization, but not necessarily need-to-know, for all data handled by the component. Any data leaving a system-high component are considered to be at the level of the component, even if they may have originated at a lower level. Commercial, off-the-shelf systems can be operated in system-high mode without modification. Data may enter a system-high component either from networks or sensors operating at that security level or from a replica controller port designated at the security level of that system.

To participate in a SINTRA configuration, a system-high component need only be able to identify the data it stores that are under replica control and report to the replica controller whenever these data are updated. Even this degree of cooperation from the legacy system may be avoided if all updates enter via the RC, since the RC could be programmed to recognize and propagate the updates as they enter the system.

For example, suppose an existing command and control system receives data from sensors and messages classified up to the TOP SECRET (TS) level and, consequently, operates as a system-high TS system. Operationally, some TS users need to manipulate all of the data on the system, but another class of users, cleared only to the SECRET level, requires only access to information up to this level. Today, this operation would likely be accommodated by configuring a SECRET system-high system and a TS system-high system connected by a guard processor. The guard might permit data to move between the two systems after they were reviewed by a human operator (see Fig. 3).

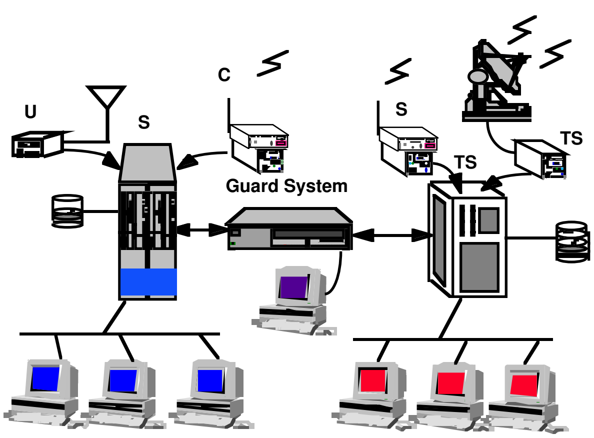


Figure 3. Conventional guard hookup for hypothetical C^4I system.

If the SINTRA approach were applied to this system, all of the system inputs classified through the SECRET level would be connected to a SECRET system-high system. This system would also be connected to a SECRET level port on a replica controller. The TS system-high system would require as inputs only those sensors and data links operated at the TOP SECRET level, together with a connection to a TOP SECRET level port of the replica controller. Since the replica controller doesn't permit downward flow of information, human review of the traffic passing from the SECRET system to the TOP-SECRET system would not be required. Both systems could run identical software, except that the SECRET system would have to pass its updates to the replica controller (see Fig. 4).

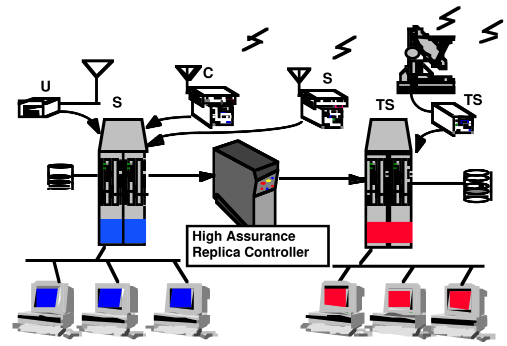


Figure 4. SINTRA approach for hypothetical C^4I legacy system.

Since in many applications updates are much less frequent than retrievals, a single replica controller might have enough capacity to serve several application systems. The single RC would handle updates for each application independently in this case. Figure 5 illustrates a possible configuration that permits the information from several enclaves producing different kinds of information to be brought together in a single system-high enclave with a single replica controller.

4. Building a Cooperative Distributed MLS Computing Service with SINTRA

The vision of cooperative, distributed computing depends on separating data access and management from application processing and user interfaces and on providing all users access to data in diverse locations. Users run complex applications on data and store the results in databases for other applications and users to access. Data sources will be many and diverse in location, structure, and sensitivity. The goal is to establish interface requirements and data dictionaries that allow access to these heterogeneous data stores. Information must be collected and stored at a granularity that permits its use by many applications and users.

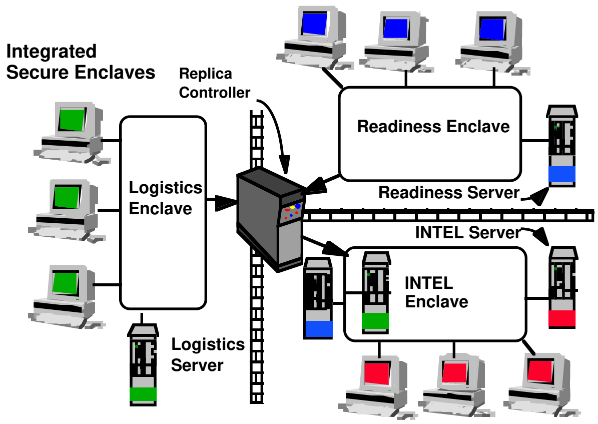


Figure 5. SINTRA approach for integrating legacy systems in secure enclaves.

Although the approach described in Section 3 allows us to interconnect legacy systems, thereby permitting useful communication across security levels, the approach cannot magically change the way legacy systems operate -- they will simply be interconnected "stovepipes" until they are re-engineered to separate data management and application processing functions. We now examine a security engineering approach that integrates cooperative, distributed computing with SINTRA.

The Approach

This approach begins by recognizing that in an MLS environment, most applications are untrustworthy; that is, it is difficult to establish confidence that an application can release data at any security level other than that of the most sensitive data it retrieves without leaking information. Consequently, in practice the application must execute at a high security level and be able to acquire data at that level and below, regardless of the details of the architecture employed.

The essence of the approach is to develop a database design that supports a variety of applications that use those data, or subsets of data restricted to a given security level and below. Applications that only require data up to a certain classification level (or that are to be run by users limited to a certain clearance level) operate on single level databases that include replicas of relevant data from all lower levels. Updates on a given database are propagated upward via replica controllers. (Note that the decision of which data to replicate is static, based on agreements among data owners -- higher level systems do not request replicas from lower level systems; this would introduce a covert channel.) Although this approach may seem a straightforward extrapolation of the ideas presented above, applying it successfully in a particular context to re-engineer existing stovepipe applications will require a thorough understanding both of the existing systems and of the fundamental operational requirements of the overall system. In fact, users' job descriptions may very well change as a result of this kind of re-engineering.

A Command and Control Example

Consider again the command and control (C^2) system example. What would migration to a cooperative, distributed computing environment mean for such an application? Today, information is communicated among diverse systems that contain C^2-related data through formatted messages concerning readiness, schedules, equipment failures and their impact on readiness, location of friendly forces, commercial traffic, enemy platforms, and more. Current systems must parse messages and run application programs to provide an updated situation assessment. Further, since each type of information may arrive in a different message type, and each message type may be processed by a different application, complex data fusion and decision support systems are required to integrate the information from diverse messages concerning a common situation.

An integrated database containing all the relevant information and using data replication for sharing could substantially simplify system operation, data fusion, and decision support. Ideally, each watch team would simply update its local database and these updates would be replicated to operational databases instead of sending messages containing the same information. The data management systems would ensure that the information is consistent and current. The key factor is to capture the information in a database design that supports the operational user. Formatted message traffic promotes the identification of related information, but achieving the most effective database design requires a solid understanding of how the data will be used to make decisions.

Discussion

With this approach, making systems interoperate becomes a problem that can be dealt with independent of security concerns, because each security level can be dealt with separately. SINTRA allows the copying of Low data to High enclaves or systems, so long as physical and procedural countermeasures ensure that only High logins are permitted there (i.e., only high users access this copy of low data). Thus at any given security level, all information at that level and all lower levels can be made available via replication. Developers seeking to join heterogeneous systems will still have to negotiate interfaces, but they need not add security concerns to their lists of potential incompatibilities.

Although the approach permits the use of low assurance COTS systems to process sensitive information, it does not provide opportunities for adversaries to exploit vulnerabilities in either high or low assurance systems, because (as noted in Section 2) the Replica Controller both prevents downward flow of information and prevents any but High users from invoking processes to be executed on High replicas of Low data.

Comparison With An Approach Based on Connecting Monolithic MLS Systems

Consider now an approach to distributed, cooperative MLS computing built on monolithic MLS systems [TNI87]. A straightforward approach would employ MLS databases (either TCB subset or trusted subject architecture) at diverse locations operating over different ranges of security levels according to local requirements. Both SINTRA and Monolithic MLS approaches require that the sensitivity of individual entities and data elements be defined. Once this is done, the database can be designed.

Some implications of the monolithic MLS approach are as follows:

- A user with an MLS workstation at one location who requires data from a different location will need to establish a connection at the security level of the data he requires. So, if he needs data from several sources, he may have to change security levels several times, and he may need to copy the data into another DBMS to perform the operations.
- Interoperation of heterogeneous systems will require solving problems in multilevel (rather than single-level) transaction management.
- If the security level ranges of two systems that need to communicate do not overlap, it may be necessary to extend the range of one of them artificially, requiring additional security management and possibly introducing vulnerabilities.
- To achieve reliability and availability goals, each system may still require a replication strategy. The SINTRA approach thus seems to avoid whole classes of problems that must be solved if monolithic MLS systems are used in this way.

The monolithic MLS approach also seems likely to cost more over the system life cycle than the SINTRA approach:

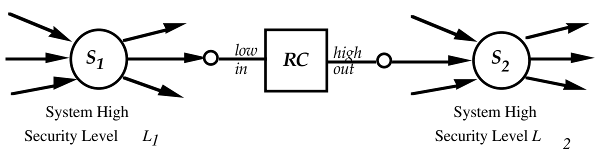
- Its cost will include development and certification of high assurance software (for the general purpose monolithic MLS system and for the MLS database).
- Because current high assurance MLS platforms do not generally conform to standard interfaces, little off-the-shelf software can be used, and any inconsistencies among various MLS products to be run on the MLS platform will have to be resolved.
- Operating costs will also be higher than for untrusted systems, because trusted software must be controlled under separate configuration management, audit records must be scrutinized for malicious use of covert channels, and operators will require special training and vetting for trustworthiness.
- Upgrades to untrusted data management systems or application software must be ported to the high assurance platform instead of merely being installed; new technology will probably be harder to incorporate.

The SINTRA approach alleviates many of these problems. SINTRA faces the one-time cost of developing the replica controller, which requires a relatively small amount of high assurance software with very limited functions. Commercial, off-the-shelf software can be used for the data management systems. Certification and accreditation reduces to an examination of the configuration of the components, including untrusted DBMSs, replica controllers, identification and authentication mechanisms, and encryption devices, so it is easy-to-understand, repeatable, scalable, and affordable. Replica controllers are analogous to encryption devices for communication systems in the sense that they can provide "add-on" protection to any transaction system. The replica controllers must be maintained as MLS devices, not systems. Only properly cleared individuals may be permitted to have access to the data and the system; however, those are the owners of the system. Other than general maintenance, no special MLS requirements are levied. The RCs that serve as the MLS connectors in a MLS, cooperative, distributed computing environment must be maintained and kept under configuration management much as encryption devices are today. Upgrades to hardware, data management systems, and application systems are simply installed because they do not affect system security.

5. SINTRA's Composition Principle

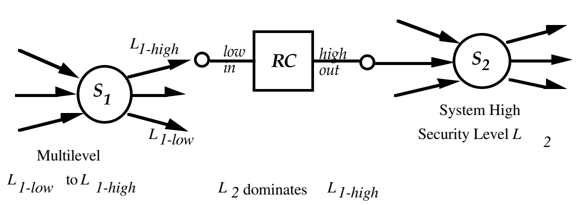
SINTRA's composition principle is simple: if a higher level system requires access to data originally classified at a lower level, then it may access replicas of those data only at its own level. In a world of SINTRA TFE's, RC's, and system-high systems, each operating at a particular security level, two systems can be connected via a TFE/RC without compromising confidentiality, because the SINTRA component only permits the upward flow of data.

Put slightly more formally, given a replica controller RC with a low input port and a high output port, two systems S_1 operating at security level L_1 and S_2 operating at L_2 (where L_2 dominates L_1), the connection

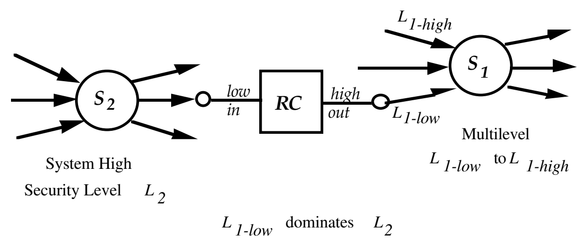


(i.e., connecting one of S_1 's outputs to RC's low input and RC's high output to one of S_2 's inputs) does not compromise confidentiality of either S_1 or S_2. The resulting composed system is multilevel: it can serve users cleared to either L_1 or L_2 and provide each user all (and only) the information for which he or she is cleared.

Now suppose we wish to connect a multilevel system with a system-high system. Let S_1 be a multilevel system operating over a lattice of security levels, say with glb L_1*-low* and lub L_1*-high*, and S_2 is as before. If L_2 dominates L_1*-high*, it suffices to connect an L_1*-high* output port to an RC input and connect an RC output to an S_2 input port. S_1 and the replica controller would be configured to propagate replicas of data classified at L_1*-high* or lower up to S_2.



If L_2 is within the lattice covered by S_1, it could be connected directly to L_1 at its own level. If L_2 is dominated by L_1*-low*, an S_2 output would be connected to a low input port on the RC and the RC output would be connected to an S_2 input port, presumably at L_1*-low*.



Finally, if L_2 is not comparable with any of the security levels in the set processed by S_1, the two systems must not be directly connected, but access to the data in the two systems can be achieved through replication of data in each system via a RC to a third system accessible only by those users having both L_2 and L_1*-high* clearances.

Although we have not addressed the direct connection of two monolithic multilevel systems, we have highlighted some of the difficulties involved in achieving interoperability between two such systems.

6. How Many Replicas Do You Need?

An explicit feature of this approach is the replication of backend databases or legacy systems. The simplest application of this approach would call for one backend machine for each level of the security lattice. For systems that include data classified at Confidential, Secret, and Top Secret, the cost of replication may be easy to justify, particularly as hardware costs continue to drop. For systems that process data from many different compartments, however, one backend per point in the classification lattice is unlikely to be practical. There is, however, an alternative approach: use one backend per clearance level, rather than per classification level. To understand why this approach is sensible, we need to consider the context of DoD clearance and classification procedures. Clearance procedures are designed to establish whether an individual can be trusted to safeguard sensitive information or not. Granting a clearance to an individual, however, does not automatically imply that she or he will be granted access to such information -- the individual must also have a job-related need to know the information in question. However, if an individual cleared for some particular level of information but lacking the need to know it somehow (perhaps through an accidental disclosure) learns it, this is not generally a major problem -- the individual's trustworthiness has previously been established, and only the need-to-know controls have been violated.

Considering the SINTRA approach in light of these observations suggests that the strong separation the approach provides between different backends corresponds most naturally to the differences in degrees of clearance. So, we would argue that it may often be adequate to rely on relatively low assurance (say TCSEC B1 level) controls within a specific backend to enforce the mandatory need-to-know constraints that compartment designations reflect, as long as all of the users with clearance for the backend in question share a clearance (though not necessarily the need-to-know) for all of the data on that backend. Proctor and Neumann made a similar observation [Proc92].

In the DoD at present, there are effectively three levels of background investigation applied to individuals: National Agency Check (NAC), National Agency Check with Inquiries (NACI), and Single Scope Background Investigation (SSBI) [JSC94]. In computer systems that will not be used by anyone with less than a Secret clearance and that include information up to TS with compartments, then, we would consider using only three backend machines even if several different compartments were processed. Secret-cleared users would have access only to the Secret backend, Top-Secret users would use the Top Secret backend, and users with clearance for TS-compartmented information would use that backend. The TFE/RC would be responsible for providing the strongest separation: assuring that users have access only to the backends for which they are cleared and assuring that information flows only upward (in security level) among the backends. To provide this separation, the TFE/RC must use the best available technology for assuring access control decisions are made correctly and covert channels among backends are removed or minimized. At present, the strongest commercially available systems for this purpose meet the TCSEC B3 level. Any backend responsible for mandatory need-to-know separation (i.e., keeping track of compartmented data and user's compartment authorizations) would require security features and assurance at least equivalent to those specified for Compartmented Mode Workstations. Backends responsible only for discretionary need-to-know separation (i.e., keeping track of access control lists for files within the same security level) would probably only require TCSEC C2 features and assurance, since the effects of security breaches are even more limited in this case.

7. Status

The basic algorithms for secure, consistent replica control have been developed. A proof-of-concept prototype, using HFSI's B3 XTS-300 as the TFE is operational with three backend databases [Kang94]. This prototype provides element-level labeling and guarantees the consistent updating of replicas without requiring the modification of the backend DBMSs. Several approaches have been investigated for propagating updates to higher security levels, including write-up and read-down. A reliable write-up algorithm, known as the Pump [Kang93b], is being implemented as a commercial product. The notion of a multilevel transaction has been defined, and several scheduling algorithms have been developed [Cost92b, Cost94]. Recovery and transaction management issues for distributed SINTRA are currently under investigation. Some details of the SINTRA data model and its application in the context of Section 4 remain to be worked out.

8. Summary and Conclusions

The SINTRA approach offers considerable promise for providing practical multilevel secure computing service. Although developed independently, it could be viewed as an extension of the Distributed Secure System work stimulated by Rushby and Randell [Rush83]. Proctor and Neumann's recent approach [Proc92] is consistent with SINTRA [Fros89]. SINTRA capitalizes on perhaps the most reliable trend in computing, which is declining storage and hardware costs, and avoids as much as possible the development of high assurance software. It permits the use of commercial database technology without changes, and it permits that technology to be upgraded as new commercial products become available. It provides a way to introduce multilevel security into the information technology mainstream, which is turning to replication as a solution to other problems in cooperative, distributed computing. It also offers a method for legacy systems to provide multilevel computing service.

It is not a panacea, however. SINTRA provides MLS data access and management services in the context of a client-server architecture. It aims to build security around existing systems, which are almost exclusively single-level, rather than trying to simplify the task of sending unclassified e-mail from a user's Top Secret workstation. Those who seek a multilevel secure workstation may not be satisfied with this approach since it does not address the MLS workstation problem (though we see some possibilities for exploiting replication within a workstation, too). For very large systems with extensive databases, or mobile systems with tight constraints on power consumption and weight, the cost of hardware replication may be difficult to bear, but necessary if availability and reliability are to be ensured.

Nevertheless, we are confident that there is a large class of systems that would benefit from this approach. It offers improvements not only in security and interoperability but also in function, by reducing the need for manned guard systems and increasing the opportunities for interconnecting systems so that timely, consistent information is available to those who need it.

Acknowledgments

We are grateful to Mike Ware for his consistent support of our research. We thank the referees for their thoughtful comments, which have improved the presentation.

References

[Aike94] Aikens, P., A. Muntz, and R. Richards, "DoD Legacy Systems; Reverse Engineering Data Requirements," *Comm. ACM, 37*, No. 5, pp. 26-41, May 1994.

[Bran89] Branstad, Martha, H. Tajalli, F. Mayer, D. Dalva, "Access Mediation in a Message Passing Kernel," *Proc. 1989 IEEE Computer Society Symposium on Security and Privacy*, Oakland, California, IEEE CS Press, 66-72.

[Brod94] Brodie, M. "The Promise of Distributed Computing and the Challenge of Legacy Information Systems," in *Advanced Database Systems: Proceedings of the 10th British National Conference on Databases*, P.M.D. Gray and R.J. Lucas (eds.), Springer-Verlag, New York/Heidelberg, 1992.

[Cost92a] Costich, O. "Transaction Processing Using an Untrusted Scheduler in a Multilevel Database with Replicated Architecture", in *Database Security V: Status and Prospects*, eds. S. Jajodia and C. Landwehr, North-Holland, 1992, pp. 173-190.

[Cost92b] Costich, O. and J. McDermott, "A Multilevel Transaction Problem for Multilevel Secure Database Systems and Its Solution for the Replicated Architecture", *Proc. 1992 IEEE Computer Society Symposium on Research in Security and Privac*y, Oakland, California, May 1992, pp. 192-203.

[Cost94] Costich, O., and M. Kang, "Maintaining Multilevel Transaction Atomicity in MLS Database Systems with Replicated Architecture," in *Database Security VII: Status and Prospects*, T. F. Keefe and C. E. Landwehr, eds,IFIP Trans. A-47, ISBN 0 444 81833 2, Elsevier Science B.V., Amsterdam, 1994, pp. 329-356.

[DoD 5200.28] DoD Directive 5200.28. Security Requirements for Automated Information Systems.21 Mar.,1988.

[Epst92] Epstein, J., *et. al*, "Evolution of a Trusted B3 Window System Prototype," *Proc. 1992 IEEE Comp. Soc. Symp. on Res. in Sec. and Privacy*, Oakland, CA, IEEE CS Press, ISBN: 0-8186-2825-1, pp.226-239.

[Fine93] Fine, Todd, S.E. Minear, "Assuring Distributed Trusted Mach," *Proc. 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, California. IEEE CS Press, ISBN: 0-8186-3370-0, pp. 206-218.

[Fros89] Froscher, J.N., and C.L. Meadows, "Achieving a Trusted Database Management System Using Parallelism," in *Database Security II: Status and Prospects*, C.E. Landwehr, ed., North Holland, 1989, 151-160.

[Glig87] Gligor, V. Verbal comments at NRL/NCSC Invitational Workshop on UNIX and Security, April, 1987.

[JSC93] Joint Security Commission. Redefining Security. A Report to the Secretary of Defense and the Director of Central Intelligence. February 28, 1994, Washington, DC 20505, p.44.

[Kang92] Kang, M.,O. Costich, and J.N. Froscher, "A Practical Transaction Model and Untrusted Transaction Manager for a Multilevel-Secure Database System" in *Database Security VI: Status and Prospects*, B.M. Thuraisingham and C.E. Landwehr, eds., North Holland, ISBN 0 444 89889 1, 1993, pp.285-300.

[Kang93a] Kang, M. H., and R. Peyton, "Design Documentation for the SINTRA Global Scheduler," NRL Memorandum Report #5542-93-7362, June 30, 1993.

[Kang93b] Kang, M., and I. Moskowitz, "A Pump for Rapid, Reliable, Secure Communication," *Proc. 1st ACM Conf. on Computer and Communications Security*, Fairfax, VA, Nov., 1993, pp. 119-129.

[Kang94] Kang, M.H., J. N. Froscher, J. P. McDermott, O. Costich, and R. Peyton, "Achieving Database Security through Data Replication: The SINTRA Prototype," *Proc. 17th Nat. Comp. Security Conf.*, Baltimore, MD, Sept., 1994.

[Land83] Landwehr, Carl E. The Best Available Technologies for Computer Security. 1983 Jul. *IEEE Computer 16*(7) pp.86-100

[Land93] Landwehr, Carl E., "How Far Can You Trust a Computer?" in *SAFECOMP'93, Proc. of the*

12th Int'l Conf. con Computer Safety, Reliability, and Security, Poznan-Kiekrz, Poland, Oct., 1993, Janusz Gorski, ed., ISBN 0-387-19838-5, Springer-Verlag, New York, 1993.

[Lubb93] Lubbes, H.O., "COMPUSEC: A Personal View," *Proc. 9th Ann. Computer Security. Applications Conf.* Orlando, FL, IEEE CS Press, ISBN 0-8186-4330-7, Dec., 1993.

[McCu87] McCullough, Daryl, "Specifications for Multilevel Security and a Hook-Up Property," *Proc. 1987 IEEE Computer Society Symposium on Security and Privacy*, pp. 161-166.

[McDe91] McDermott, J., S. Jajodia, and R. Sandhu, "A Single-Level Scheduler for the Replicated Architecture for Multilevel-Secure Databases", *Proc. 7th Annual Comp. Security Applications Conf.*, San Antonio, IEEE CS Press, December 1991, pp. 2-11.

[McDe93] McDermott, J., and R. Mukkamala, "Performance Analysis of Transaction Management Algorithms for the SINTRA Replicated-Architecture Database System," in *Database Security VII: Status and Prospects*, T.F. Keefe and C.E. Landwehr, eds., North Holland, ISBN 0 444 81833 2, 1994, pp.215-234.

[McLe88] McLean, John, "The Algebra of Security," *Proc. 1988 IEEE Symposium on Security and Privacy*, Oakland, CA, IEEE CS Press, ISBN 0-8186-0850-1, pp.2-7.

[McLe89] McLean, J. and C. Meadows, "Composable Security Properties," *IEEE Computer Security Foundations Workshop II*, reprinted in Cipher (newsletter of the IEEE CS TC on Security and Privacy), Fall 1989, pp. 27-36.

[McLe94] McLean, J. "A General Theory of Composition for Trace Sets Closed under Selective Interleaving Functions," *Proc. 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland, CA, IEEE CS Press, ISBN 0-8186-5675-1, pp. 79-95.

[MDMS83] *Multilevel Data Management Security*. Air Force Studies Board, Commission on Engineering and Technical Systems, National Research Council, National Academy Press, Washington, D.C., fall, 1983.

[NCSC85] Final Evaluation Report, Honeywell Inf. Sys. Multics MR11.0, Nat. Computer Security Center Rep. #CSC-EPL-85/003, Ft. Meade, MD, 1985.

[NRL87] NRL/NCSC Invitational Workshop on UNIX and Security, April 1-2, 1987, Naval Research Laboratory, Washington, D.C.

[Orga72] Organick, E. I. *The Multics System : An Examination of its Structure.* 1972. MIT Press, Cambridge, MA

[Popek79] Popek, G. J., M. Kampe, C.S. Kline, A. Stoughton, M. Urban; E. J. Walton, "UCLA Secure UNIX," *AFIPS Conf. Proc. 48* (1979 NCC) pp.355-64

[Proc92] Proctor, N. E., and P. G. Neumann, "Architectural Implications of Covert Channels," *Proc. 15th National Computer Security Conference*, Oct. 1992, pp. 28-43.

[Rush83] Rushby, J. M. and B. Randell, "A Distributed Secure System," *IEEE Computer 16*(7) pp.55-67.

[Sayd94] Saydjari, O. Sami, S. J. Turner, D. E. Peele, J. F. Farrell, P. A. Loscocco, W. Kutz, G. L. Bock. Synergy: A Distributed Microkernel-based Security Architecture, Version 1.0, National Security Agency, INFOSEC Research and Technology, Nov. 22, 1993.

[TNI87] Trusted Network Interpretation of the TCSEC. National Computer Security Center NCSC-TG-005, Version 1, July 1987, p. 229.

[TDI91] Trusted Database Management System Interpretation of the TCSEC. National Computer Security Center NCSC-TG-021, Version 1, April 1991.