IDA

INSTITUTE FOR DEFENSE ANALYSES

Recommendations and a Plan of Action for Standardized Security Labeling

Ron S. Ross, Task Leader

Stephen R. Welke
John M. Boone
Edward A. Feustel
W. T. Mayfield

July 1995

IDA Paper P-3044

Log: H 95-047033

This work was conducted under contract DASW01 94 C 0054, Task T-S5-1210, for the Defense Information Systems Agency. The publication of this IDA document does not indicate endorsement by the Department of Defense, nor should the contents be construed as reflecting the official position of that Agency.

PREFACE

This report was prepared by the Institute for Defense Analyses (IDA) under a task order, Strategy for Security Standards in Support of the Defense Information System Security Program (DISSP), for the Defense Information Systems Agency (DISA). The report contributes to an objective of the task, to develop a complete and integrated Department of Defense approach, in support of the DISSP, for developing standards to protect information being transported, processed, or stored.

The report was reviewed by IDA research staff members Dr. Alfred E. Brenner and Dr. Richard J. Ivanetich. The following IDA research staff members provided very useful and important technical contributions, especially on the abstract model and goal security architecture formalisms: Dr. Dennis W. Fife, Dr. Reginald N. Meeson, Dr. Asghar I. Noor, and Mr. Glen R. White.

The authors are also indebted to Dr. William Flanigan from DISA's Center for Standards, Dr. Dave Gomberg from the MITRE Corporation, Mr. William McAllister from the National Security Agency, and the members of the Information Systems Security Standards Working Group for providing important feedback on the early drafts of this report. Finally, a special note of thanks is given to our technical editor, Ms. Katydean Price, for her excellent editorial support.

Table of Contents

List of Figures

List of Tables

EXECUTIVE SUMMARY

The Department of Defense (DoD) recognizes that today's ever-increasing use of information technology to conduct routine business makes protecting automated information essential. Security labels are one type of computer security mechanism used to facilitate controlled access to information in a shared resource environment. The purpose of this report is to recommend how computer security label standards should be pursued in light of existing labeling technology and the new security architecture being developed for DoD. Label standards are necessary to facilitate the integration, interoperability, and cost-effective implementation of protection in information systems.

Manually marking paper documents with security labels, and the associated procedures for enforcing protection requirements represented by the label, are well understood. As security labels are adapted to protect electronic documents within a network of distributed information systems and multi-media devices, these marking and enforcement procedures are now being recast. To date, most efforts to standardize computer security labels address single information system components and leave system-level labeling interfaces undefined. This report uses the model illustrated in Figure ES-1 to discuss component and system-level label standardization issues in five components of an information system: network, operating systems, applications, human-readable devices, and storage media.

The authors of the report examined existing label implementations, leveraged and synthesized related work, and studied existing and emerging label standardization efforts to gain a better understanding of the successes and failures of labeling technologies and standards. As this work progressed, the authors discovered that the DoD Goal Security Architecture (DGSA) was an emerging architecture that could significantly change how information will be protected by the DoD in the foreseeable future. Therefore, the report presents a brief description of the DGSA and formalizes its fundamental security concepts and principles before making recommendations about pursuing security label standardization.

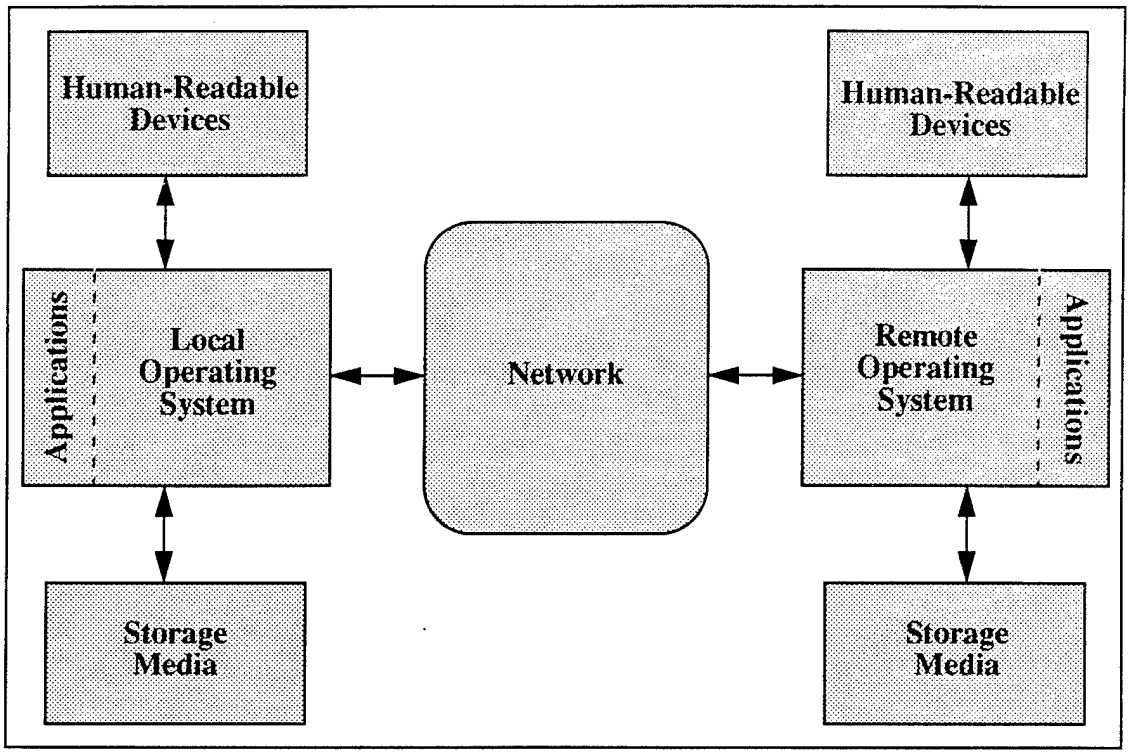


Figure ES-1. Information System Model

The body of the report consists of three assessment areas: security labeling standards, security labeling technical investigations, and future directions in security labeling. The primary factors involved in developing a useful security label standard include coordinating standardization efforts, capturing requirements, assessing market factors, allowing for technology development, pursuing interoperability, and registering labels. These factors set the context for examining label implementations. Five technical investigations (one for each component type in the information model in Figure ES-1) are presented as examples of the kinds of security label technologies that exist in state-of-the-art products. These investigations provide insights into the label structures and interface approaches that are used in existing commercial products, and into the trends of security label implementations in information systems. The future direction of security labeling is captured by describing the DGSA and its fundamental information management concepts: security policies, controlled entities, information domains, and information systems.

Findings

The three assessment areas led to five major findings.

The DGSA is a paradigm shift from Multi-Level Security (MLS). This finding is particularly important in light of a March 1995 memorandum from the Assistant Secretary of Defense for Command, Control, Communications, and Intelligence stating that new DoD systems development and modernization programs must conform to the Technical Architecture Framework for Information Management (TAFIM). The DGSA is an integral part of the TAFIM. The MLS paradigm inherently employs a lattice-based relationship between security labels. The DGSA paradigm employs security labels in information domains, where there are no implicit relationships between domains. The DGSA still requires that access to information be mediated, but the mediation is not the same as that in an MLS system.

The information domain concept is fundamental to the DGSA. The DGSA changes the multi-level approach to information protection and emphasizes a mission-based approach. Information domains consist of information objects and end users constrained by a domain security policy. Information objects must be uniquely identifiable, and explicit domain membership is required of all end users. Domain security policies are supported by security attributes, where every object in a domain has the same security attribute values, and there is no relationship between the attribute values in different domains. Explicit import-export rules must be developed as part of a domain security policy to control the flow of information between domains.

A security label has different representations (e.g., native, network, and human-readable) as it travels through the five components of an information system. Networks need an interoperable network representation of a label. Network labeling protocols should provide the syntax for needed attributes, including a field to indicate the semantics of the attribute values. Operating systems, applications, and storage media implement native representations to optimize label processing. Human-readable devices provide human-readable representations to help users understand what a label means when it is printed or displayed. There may be a need for products to support a minimum number of labels.

The DoD community will focus more on commercially developed standards in the future. This finding is based on a June 1994 memorandum from the Secretary of Defense. Thus, if the DGSA is to succeed, users must demand label-based products supporting the DGSA and vendors must build such products. As commercial solutions to labeling

standardization issues are pursued, however, it is important that those solutions are tempered by DoD policy. This goal could be accomplished by dividing security labeling specifications between a generic standard describing label structure and format and specific implementation guidance describing how DoD should use the standard. The Standard Security Label (SSL) and Common Security Label (CSL) efforts are prime candidates for this type of relationship.

There is no registry and associated organization for security label registration within the DoD. Although there has been considerable effort by DoD to develop the CSL, there has not been a parallel effort to develop an appropriate registration process and to assign an organization to carry out the necessary registration activities. Without a registry for security labeling, critical information cannot be catalogued for general use by the DoD community.

Conclusions

The successful use of security labels depends on customer *demand* and government and industry *supply*. Government and industry supply provides a means to satisfy the customer demand through the following: security labeling *standards*, administrative *organizations* to support security labeling activities, and the availability of affordable *technologies* to implement security labeling within enterprise information systems. The technical investigations show that network label standardization has strong support from commercial vendors, but that commercial vendors are much less willing to support security label standardization within an end system (specifically in the operating system).

The DGSA offers a basis for all future architectural decisions relating to the design, development, and implementation of secure information systems within DoD. Given the commitment to the DGSA as part of the Technical Architecture Framework for Information Management (TAFIM), continuing investment in new security label standards outside of the scope of the DGSA does not appear to be in the best interests of DoD or the Federal Government, and could prove to be counter-productive in the long term. Gaining commercial industry support for DGSA concepts is crucial. DoD should identify and promote the significant overlap of DoD's and commercial industry's need for security labels, and should articulate these common needs in the context of the DGSA. The Government must bear the costs and risks associated with developing and proving DGSA-supportive technology in the near term, and then provide the opportunities and paths to transfer this technology to the commercial world.

1. INTRODUCTION

As information technology advances at an unprecedented rate and becomes an integral part of how the world conducts business, protection of information and system assets becomes increasingly important. However, the complexity and common use of modern information systems[1] make their protection increasingly difficult. This report focuses on two important aspects of protecting information and automated system assets in the continually expanding network of information systems: security labels and label standardization. It provides recommendations about how security label standards should be pursued in light of existing label technology and the new security architecture that has been developed for the Department of Defense (DoD).[2]

Security labels are a necessary part of protection because they facilitate uniform, system-wide policy enforcement by separating information into distinct classes. Labels, by themselves, however, are not sufficient for achieving security. Underlying technical mechanisms (e.g., a reference validation mechanism) and administrative procedures (e.g., locking a document in a secure file cabinet) must be in place to enforce the protection requirements represented by the label. Label standardization is also a necessary part of protection because standardization facilitates the integration, interoperability, and cost-effective implementation of information system security architectures.

1.1 BACKGROUND

Historically, security labels have been used in the "paper world" to manually mark documents with required or useful security information. Whether their use has been motivated by governmental laws, national policies, industrial regulations, or personal choice, such security labels have served as a shorthand for expressing, either explicitly or

---

[1] Traditional boundaries between communications and computing systems are being replaced by the new paradigm of an *information system*, which is defined as any component or group of components that generates, collects, processes, stores, transfers, disseminates, or disposes of information [DISSP93].

[2] The report assumes the reader has some knowledge of computer security, in general, and security labeling, in particular. The reader is referred to Gasser[1988] and Pfleeger[1989] for general background and to the Bibliography for more detailed background.

implicitly, protection requirements about a document. The use of manual marking labels has resulted in institutionalized physical and administrative procedures to enforce the protection requirements represented by the label.

As security labels are adapted to protect electronic documents within a network of distributed information systems and multi-media devices, marking and enforcement procedures now must be recast. For example, electronic documents could be protected by physical separation. However, the need for connectivity and information/resource sharing to accomplish mission-related tasks makes physical separation less attractive. Thus, the need for security labels within the information system environment to facilitate proper identification and authorized sharing of electronic information becomes an increasingly important requirement. Implementing security labels in a paperless world while retaining the marking concepts at the human interface (e.g., monitor, printout) is a formidable challenge that the DoD must address.[3]

1.2 SCOPE

Many efforts have been made to standardize security labels and their use [DISA93]. Most of these efforts have been adequate for the needs of a particular system component, but have left the labeling interfaces for distributed information systems undefined. This approach to standardization has become less effective as the proliferation of distributed systems leads to greater emphasis on the system-level requirements for integration, interoperability, portability, and extensibility. As a result, this report examines label standardization from a system perspective.

1.3 APPROACH

Figure 1 identifies five components of an information system: network, operating systems, applications, human-readable devices, and storage media. It provides a framework for discussing specific and overall label standardization issues, and will also be used to walk through the concept of operations for how a label traverses through an information system. Existing label implementations were examined, work initiated or accomplished by others was leveraged and synthesized, and existing and emerging label standardization

[3] A recent report by the Joint Security Commission [JSC94] provides an excellent description of the new information systems security reality that exists within enterprises today. The report concludes that the DoD and intelligence communities must have capabilities such as managing and protecting information of any sensitivity on a global basis, transmitting classified information securely over public networks to authorized users, and maintaining confidentiality and integrity of sensitive but unclassified information (e.g., financial data, medical records, and personnel files).

efforts were studied. As this work progressed, it became evident that the DoD Goal Security Architecture (DGSA)[4] was an emerging architecture that could significantly change how information will be protected by the DoD in the foreseeable future. Therefore, a significant portion of this report is dedicated to analyzing security labeling within the context of the DGSA. The concepts and principles of the DGSA were studied before recommendations about security label standardization were made.[5]

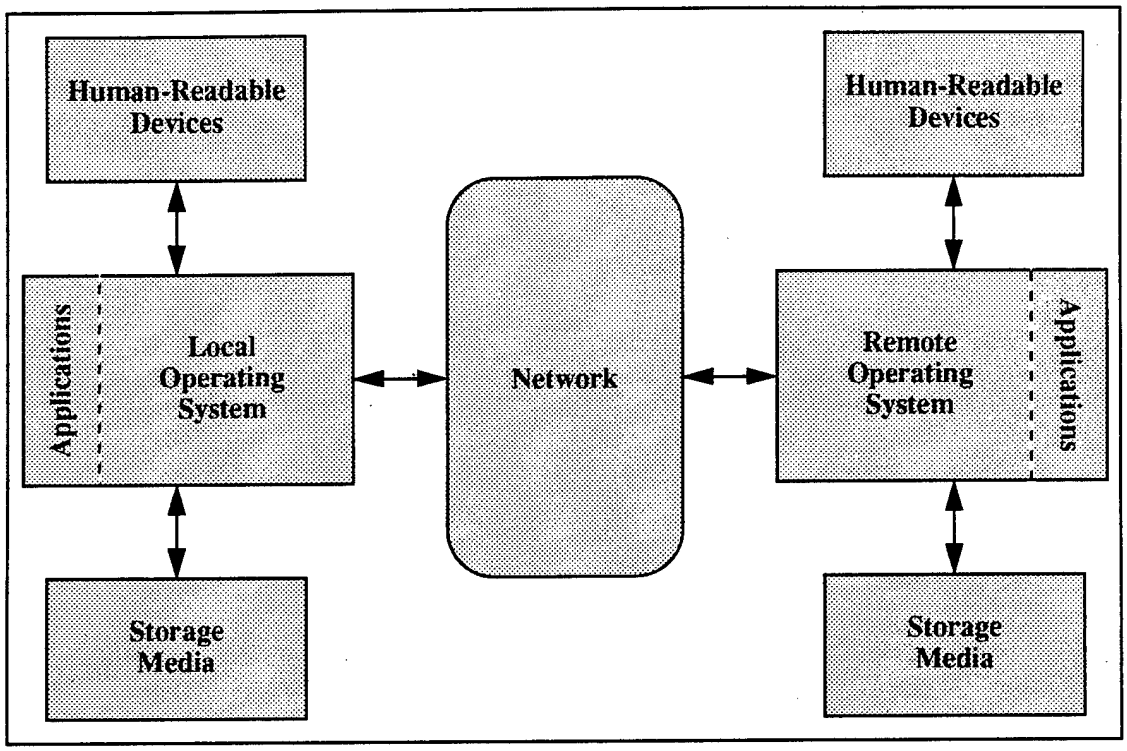


Figure 1. Information System Model

---

[4] A DoD enterprise strategy has been developed by the Defense Information Systems Agency (DISA) to take advantage of emerging information technologies and effectively manage the migration of information systems within the DoD. This strategy focuses on the definition of information architectures for the evolving Defense Information Infrastructure (DII) under a Technical Architecture Framework for Information Management (TAFIM) [TAFIM93]. The DGSA is an integral part of the TAFIM.

[5] Security labeling is inherently dependent on underlying architectures and associated implementations. The DGSA approach to information protection provides a contrast to the Multi-Level Security (MLS) approach employed in some of the current DoD information systems. Thus, a significant portion of this report is devoted to analyzing the DGSA in order to understand the effect it will have on future security label implementations. While a detailed discussion of MLS is not within the scope of this report, the distinction between the two paradigms and approaches to information protection will be noted. For additional details on MLS policies, architectures and principles, the reader is referred to Gasser[1988].

1.4 ORGANIZATION

In the process of researching label standardization with respect to the five components of an information system and the DGSA, it became useful to organize the results into three assessment areas as shown in Figure 2. A chapter has been devoted to each of these assessment areas. Chapter 2 presents the primary factors involved in the development of a useful security label standard: coordinating efforts, capturing requirements, market factors, technology development, interoperability, and the registration process. Chapter 3 presents five technical investigations (one investigation for each of the components identified in Figure 1) that examined examples of the kinds of security label technology that exist in state-of-the-art products. These investigations provide insights into the label structures and interface approaches that are used in existing commercial products, and the technology trends of implementing security labels in information systems. Chapter 4 presents the fundamental (DGSA-related) components of information management that are important for the future of security labeling: security policies, controlled entities, information domains, and information systems. Chapter 5 presents findings and conclusions based on the three assessment areas discussed in the previous chapters, and Chapter 6 presents recommendations and a suggested action plan.

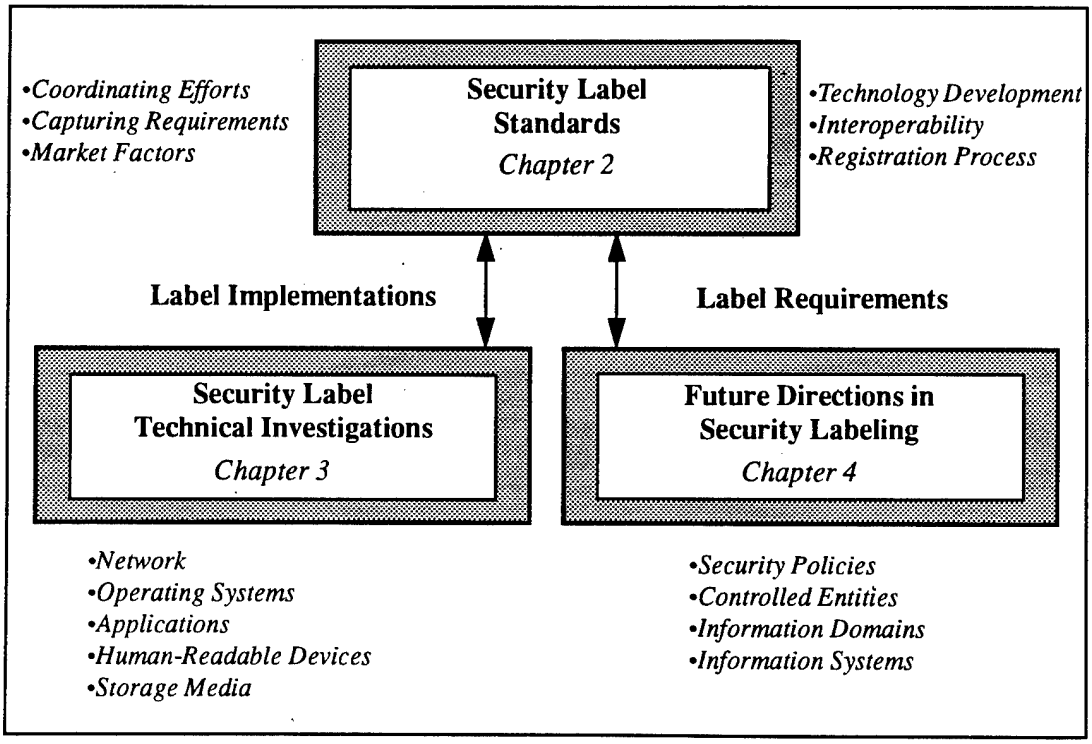


Figure 2. Organization of Analysis

2. SECURITY LABELING STANDARDS

This chapter describes the factors involved in developing a security label standard. These factors, presented without regard to order, are generally applicable to any type of standards effort. The chapter describes how the factors apply to the standardization of security labels.

Standards are the result of input from and interaction among standardization bodies (e.g., American National Standards Institute (ANSI) and International Organization for Standardization (ISO)), product developers (e.g., IBM and Microsoft), and product users (e.g., DoD, government agencies, banks, and health care providers). A standard

> ... represents a statement by its authors, who believe that their work will be understood, accepted, and implemented by the market. This belief is tempered by the understanding that the market will act in its own best interests, even if these do not coincide with the standard. A standard is also one of the agents used by the standardization process to bring about market change [Cargill89, p. 41].

Standards can be either *formal* or *de facto*. The term "formal" is used to denote explicit standardization efforts (e.g., government standards), whereas "de facto" is used to denote standardization driven by specific products and market forces. If formal standards are not accepted and implemented in products, then de facto standards may emerge through market domination by a proprietary product. De facto standards may affect the production of formal standards and are often adopted by formal standards producing bodies; therefore, de facto standards are an important consideration in the standardization process. Figure 3 shows the relationships that exist between the organizations, technology, and standards involved in the standardization process.

The goal of any effort to standardize computer security mechanisms should be to cost effectively protect information having varying sensitivities. The sensitivities could result from (1) national information security classification regulations, (2) regulations affecting the protection of Government-held unclassified information (e.g., Privacy Act of 1974), (3) proprietary restrictions, and/or (4) mission-critical protection needs (e.g.,

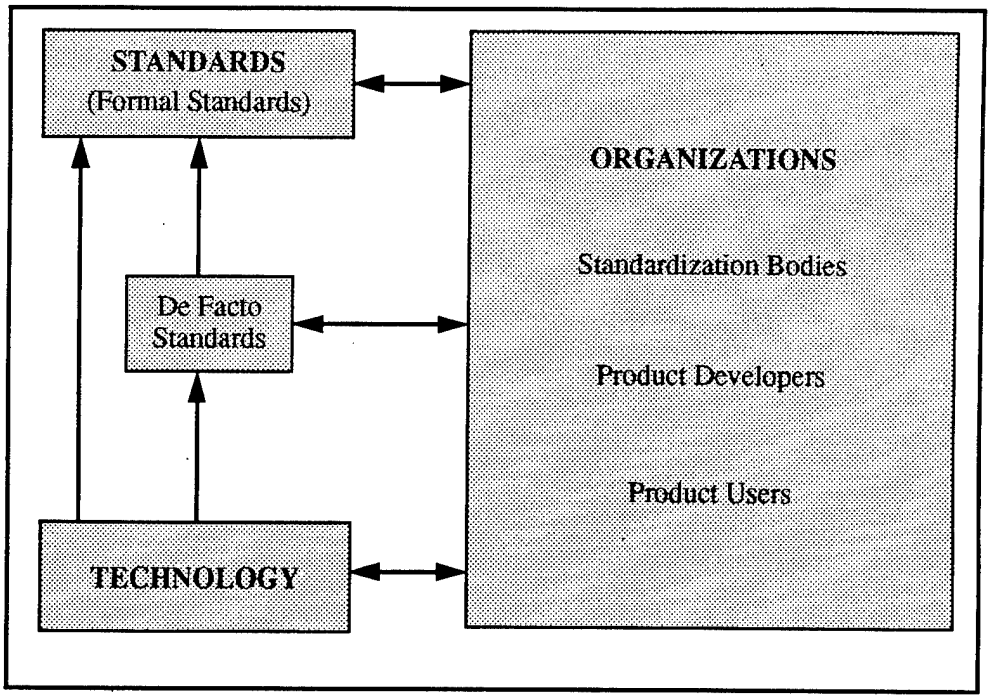


Figure 3. Standardization Environment Components

financial and process control data). Security labels are one type of computer security mechanism most applicable for access control operations in a shared resource environment, where uniform, mandatory protection is required. Other computer security mechanisms (e.g., cryptography, checksums, and digital signatures) could also be used for mandatory protection; however, these mechanisms are typically used to achieve other protection goals (e.g., authentication, integrity, and non-repudiation).

2.1 COORDINATING EFFORTS

Within DoD, the nature of standardization is undergoing change. Recent events have brought about a change in policy that must be taken into account when determining DoD standardization issues and their effects. The most important policy change is outlined in the Perry memorandum [OSD94], which calls for more reliance on the commercial marketplace as a source of products and standards. The Perry memorandum states

> [M]oving to greater use of performance and commercial specifications and standards is one of the most important actions that DoD must take to ensure we are able to meet our military, economic, and policy objectives in the future.

DISA[1993] identifies many of the efforts that have been undertaken, with varying degrees of success, to develop security label standards. The majority of these efforts are sponsored by formal standards bodies (e.g., National Institute of Standards and Technology (NIST), Defense Intelligence Agency (DIA), Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), ANSI, ISO, and DoD). However, consortia are emerging as a major factor in standards development. The Object Management Group (OMG), X/Open, the Open Software Foundation (OSF), and the Trusted Systems Interoperability Group (TSIG) serve as examples of consortia that are influencing the development of de facto standards for security labels. Consortia appear to be the response of industry to the reality that individual manufacturers and vendors are no longer capable of setting de facto standards unilaterally. De facto standardization is very powerful and often quicker than formal standards. The Perry memorandum points out that DoD must make greater use of these different efforts to ensure meeting DoD objectives in the future.

As an example of the potential for coordinating efforts, the Common Security Label (CSL) [DoD95] and the Standard Security Label (SSL) [NIST94] are protocol standards that are very similar to the Commercial Internet Protocol Security Option (CIPSO) [IETF93]. CSL was developed by the Data Communication Protocol Standards (DCPS) Technical Management Panel (DTMP) Working Group 3 in response to DoD operational needs, and SSL was developed by NIST for the Government Open Systems Interconnection Profile (GOSIP).[5] The CSL effort began first and has produced a product that is fully coordinated with SSL. The SSL effort is being considered at different layers of the Open Systems Interconnection Reference Model [ISO89]. There is virtually no difference in the syntax (i.e., structure) of CSL and SSL; the only notable difference is that SSL allows multiple tag sets per label while CSL allows just one tag set per label. Also, CSL uses the term "domain of interpretation" and SSL uses the term "named tag set" for the field that communicates the semantics (i.e., meaning) of associated security attribute values.

2.2 CAPTURING REQUIREMENTS

Standardization bodies must coordinate with product developers and users to ensure that standards capture label requirements in a way that is compatible with existing and evolving technology. When standards-producing bodies fail to capture the true

---

[5] TSIG is currently working with CSL to pursue the addition of three more tags to be used for authentication and digital signatures.

requirements of their constituent user communities, the result is the development of undesirable standards, the lack of necessary standards, or both. Inadequacy of standards can be assessed in many ways, including technical requirements, interoperability, or cost. This issue is particularly difficult in that the standardization process must be "forward-looking" to some extent (i.e., the relevance of a standard should persist over an appreciable time period). Hence, changing conditions and requirements may induce additional errors into an already uncertain process.

For security labels, a standard must provide for the implementation of products meeting the needs of users without unduly inhibiting advances within the state of the practice. Any security label standard must provide for sufficient "expressiveness" of the label. The standardized attributes carried by a security label must be expressive enough to allow the enforcement of a given community's security policy. Security label standardization must also consider changing security paradigms. In particular, limiting the scope of a security label standard to the current paradigm for security labeling (i.e., MLS) could inhibit necessary or desirable changes in technology.

2.3 MARKET FACTORS

Since DoD is downsizing and has limited resources, commercial market factors (i.e., costs) are becoming increasingly important. The viability of a standard can be measured by the degree to which end users are able to obtain affordable technology that meets their needs. If the technology implementing security label standards is expensive to develop and implement, users probably will not use it. If functional needs are not met, end users may either avoid the product or be forced to implement expensive customization. To address the cost issue, today's standards must appeal to the widest market possible (e.g., commercial, non-DoD Government, and DoD) to generate the most response in developing products. As discussed below, cost is driven by both supply and demand.

2.3.1 Supply

Supply issues address the willingness and capability of commercial vendors to produce the types of security labeling products that meet the standards required by the DoD. If it is not profitable for a producer to meet the requirements in a standard, products will not be available. Mandatory standards are no guarantee that products will be supplied by vendors. The typical example is the Trusted Computer System Evaluation Criteria (TCSEC), which has not produced a large supply of products supporting security labeling (as evidenced by the listing of evaluated products [NSA94]).

2.3.2 Demand

Demand issues address the willingness of DoD users to buy and use available security labeling products. Solutions that have a serious effect on efficiency will not be accepted by the market or by users. Technical solutions must minimize the overhead costs associated with using security labels. Overhead can be accumulated in the processing environment of systems using labels and in the administration of those systems. Since the use of labels implies extra processing (versus not using labels), this source of overhead could, depending on the design of the mechanisms, be prohibitively large. Similarly, administrative costs will increase due to the maintenance of security label attributes.[6]

2.4 TECHNOLOGY DEVELOPMENT

New technologies are making it increasingly difficult to standardize due to a rapidly shifting technology base. For instance, security labeling standardization must be considered within the context of distributed system architectures. These architectures pose different systems management and administration challenges than do mainframe (centralized) security architectures. Distributed system domains consist of a logical partition of physical systems and subsystems as opposed to a traditional partition along the lines of physical connectivity. Standards must be developed to support the management and administration of security labels across distributed systems.

2.5 INTEROPERABILITY

For standardized products to be effective, the focus of standardization efforts must be on systems rather than stand-alone products. Products that do not interoperate with other systems or system components will not be useful. Security label requirements are pervasive throughout the computing environment, appearing in all five information system components described in this report. This entire range of security labeling requirements must be considered when pursuing label standardization opportunities. Maximizing interoperability through the use of a single labeling standard or a potential family of standards while maintaining vendor flexibility in systems design is a challenging task.

2.6 REGISTRATION PROCESS

Registration, for the purposes of this report, is a formally controlled process by which technical specifications are globally named and managed. The registration process

---

[6] Although administrative overhead is not purely technical in nature, it can be exacerbated by the design of the mechanism.

maintains the name-space and associated specifications so that ambiguity is avoided. For security labeling, the registration of security policy and label attribute specifications will allow end users to communicate and share protected information while performing mission-oriented tasks.

The formal procedure for submitting a specification to a registration body must be determined, implemented, and tested. Any constraints (e.g., secrecy of specifications) must be determined and accounted for in the overall process structure. Some technical specifications may be short term or project related while others may persist over a long period. The current concept of DoD registration of security label specifications also calls for a technical review prior to registration. This review requirement exceeds the general requirements for registration given in NIST[1993]. The review process must be incorporated into the general registration procedure. The foci of the technical review (e.g., security policies, technical criteria, and user needs) must be determined.

Once the procedure for submitting a specification has been established, there must be a management structure in place to orchestrate its operations. Management structures will be involved in the formal acceptance of specifications into a registry. Control of the registered name-space can be managed in either a centralized or decentralized manner, or by some combination of the two. In centralized management, a single authority controls the entire name-space. For decentralized management, name-space values are arbitrated between the individual parties wishing to communicate. It is unclear whether NIST's object registry [NIST93] is sufficient to name security policy and label attribute specifications. It is also unclear who would take on the task of managing the registration of such specifications.

2.7 SUMMARY OF STANDARDIZATION FACTORS

This chapter has presented the underlying factors involved in developing a useful security label standard. Cooperation is required between standards bodies, product developers, and product users. These groups must work together to properly capture label requirements in the midst of the evolution of new technologies. They must coordinate standards efforts to support interoperability while allowing flexibility of market factors. Finally, a registration process must be established to globally name and manage technical specifications that support label standardization.

3. SECURITY LABELING TECHNICAL INVESTIGATIONS

Chapter 2 identified the primary factors involved in standardizing security labels, thus establishing a context to examine technology that exists for supporting labels. This chapter presents five technical investigations as examples of label implementations and label translations in information system components. Figure 4 provides an overlay of the flow of a label through the different components of an information system. A label in the local operating system has a local representation that must be translated by applications and device drivers to be recognized by human-readable devices, storage media, or the network. Once the local label has been translated into a network representation, the network label travels with its associated container of data over the network. This network label then must be translated into a remote representation when it arrives at the remote operating system. Thus, a label flowing through an information system has many different representations.

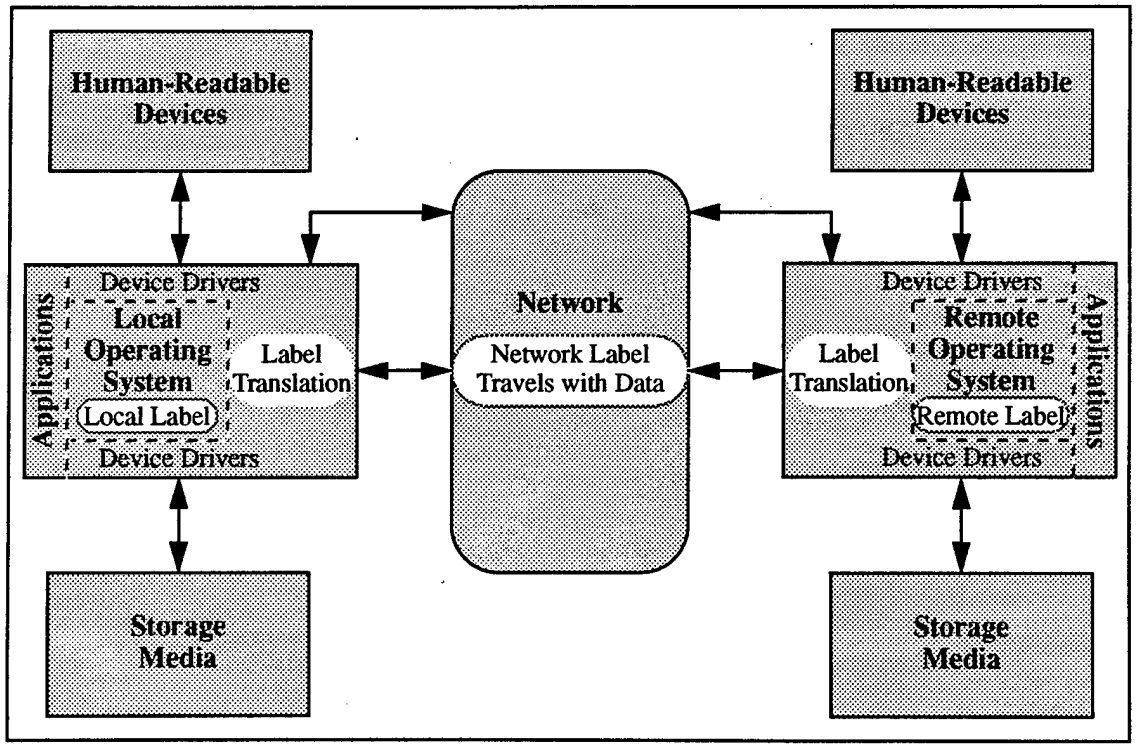


Figure 4. Label Flow Through an Information System

To serve as a baseline for the technical investigations in this chapter, it is important to identify the current availability of products supporting security labels. Table 1 summarizes the products published in the Evaluated Products List (EPL) for Trusted Computer Systems [NSA94] and a few products that are related to the EPL.[8] The products are listed in alphabetical order by vendor rather than in any priority order; the usefulness of any particular product is based on the security policy it is targeted to meet.

Note that the supply of trusted products is small. Furthermore, Table 1 shows that this small number of label-based products implement a wide variety of label name space capacities.[9] The variety of label name space capacities is due to the variety of security policies that different products are built to meet. For example, an organization with a security policy requiring only a small, fixed number of security levels and categories may choose a product with a small label name space for optimized performance. Security label name space capacity is a critical issue with respect to interoperability and efficiency of implementation. If a product supports labels but is not able to provide a sufficient number of labels to adequately communicate with other label-based products, labels will not be useful due to a lack of interoperability. This variety in label name space capacities supports the observation in Mayfield[1991] that the demand for label-based products is low.

The remainder of this chapter describes detailed technical investigations of security label implementations in the five components of an information system. It also describes the translation of label representations between the various components. Although many products were considered for inclusion in this set of investigations,[10] only one product per component type was chosen as an example to be investigated in detail. The products chosen should not be construed to imply any ranking. The information presented in this chapter is based on Final Evaluation Reports for products published in the EPL [NSA94], other open literature, and conversations with vendor representatives of each product.

---

[8] Sun Trusted Solaris and Sybase Secure SQL Server are currently undergoing evaluation. Gemini BLACKER and SecureWare `spax` were not evaluated, but they support Gemini GTNP and SecureWare CMW+, respectively. TSIG MaxSix was developed by a consortium of trusted product vendors, many of whom have products on the EPL. The rest of the products are listed on the EPL.

[9] The "Levels" and "Categories" columns in Table 1 indicate the number of hierarchical levels and non-hierarchical categories, respectively, that are supported by each product's label. The levels and categories represent a product's label name space capacity (unless noted differently under "Other").

[10] Four network products (Gemini BLACKER, Boeing MLS LAN SNSS, TSIG MaxSix, and Verdix VSLANE), six operating systems (AT&T System V/MLS, Cray Trusted UNICOS, Harris CX/SX, HFSI XTS-200, SecureWare CMW+, and Sun Trusted Solaris), two applications (Informix OnLine/Secure and Sybase Secure SQL Server), one human-readable device (SecureWare's trusted windowing for CMW+), and one storage media product (SecureWare's `spax` program) were considered.

Table 1. Commercial Products Supporting Labels

Product	Product Type	Label Name Space		
		Levels	Categories	Other
Amdahl UTS/MLS	OS	255	1,024	
AT&T System V/MLS	OS	256	1,024	16-bit tag points to label
Boeing MLS LAN SNSS	Network	8 sensitivity 8 integrity	256 sensitivity 256 integrity	
Cray Trusted UNICOS	Network	19	63	
DEC SEVMS VAX	OS			-- not available --
Gemini BLACKER	Network	8	16	
Gemini GTNP	Network	16 sensitivity 16 integrity	64 sensitivity 32 integrity	
Harris CX/SX	OS	256	1,024	16-bit tag points to label
Harris CX/SX with LAN/SX	Network	255	512	
Hewlett-Packard HP-UX BLS	OS			-- not available --
HFSI Multics	OS	8	18	
HFSI SCOMP	OS			-- not available --
HFSI XTS-200	OS	16 sensitivity 8 integrity	64 sensitivity 16 integrity	16 bits extra (for NATO caveats)
Informix OnLine/Secure	Database			name space of underlying OS
IBM MVS/ESA	OS	254	11,761	
Oracle Trusted Oracle7	Database			name space of underlying OS
SecureWare CMW+	OS			32-bit tag points to label
SecureWare CMW+ trusted windowing	Human-Readable			name space of underlying OS
SecureWare spax	Storage			name space of underlying OS
SGI Trusted IRIX/B	OS	256 sensitivity 256 integrity	2^{16} sensitivity 2^{16} integrity	
Sun Trusted Solaris	OS	256	128	
Sybase Secure SQL Server	Database			name space of underlying OS
TIS Trusted Xenix	OS	255	64	
TSIG MaxSix	Network			name space of hosts' Encodings files
Unisys OS 1100/2200	OS	64	30	
Verdix VSLANE	Network	16	64	

3.1 SECURITY LABELS IN NETWORKS

This technical investigation is based on MaxSix [SecureWare93], a product developed by a consortium of trusted product vendors called the Trusted Systems Interoperability Group (TSIG).[11] TSIG's MaxSix assumes that a standard compartmented mode workstation (CMW) Encodings file [DIA91a] exists on each host. The Encodings file contains the native (internal) representation of all of the security attributes (e.g., labels, privileges, identity) supported by the host along with their associated human-readable representations. Security label attributes in MaxSix are passed at two layers of the Open Systems Interconnection Reference Model [ISO89], network and session. Integral to the passing of security label attributes is the concept of a "domain." Network layer and session layer protocols give the syntax for communicating security attributes, and domains with their associated mappings give the semantics.[12]

MaxSix requires each host to have a database for the interfaces it provides and a database for the remote hosts to which it is allowed to connect. A database identifying domains and domain mappings may also be needed. System administrators make entries into these databases to establish what security attributes will be transferred with network datagrams to particular remote hosts and how such attributes will be transferred. System administrators must also perform the following configuration steps before a MaxSix host can communicate with a remote host:

- Identify the interface to be used in communicating with the remote host.
- Specify the Internet Protocol (IP) address of the remote host.
- Set the accreditation (sensitivity) range of the remote host.
- Select the security attributes that must be present when communicating with the remote host.
- Establish defaults for attributes received from a remote host.

---

[11] The TSIG was officially chartered in April 1990 to facilitate the design and development process of vendors who are interested in interoperability of their trusted systems. It is an interoperability and/or standards body that provides a forum where vendors can develop the necessary agreements in design and implementation to ensure interoperability among trusted systems. TSIG participants include Argus, Cray, Data General, Digital Equipment Corporation, Harris, Hewlett-Packard, HFSI, IBM, Oracle, Santa Cruz Operations, SecureWare, Sequent, and Sun. Project MAX is a consortium of vendors (having significant overlap in membership with TSIG) that is funding the common implementation of MaxSix.

[12] Evolving secure HyperMedia protocols — such as the Secure HyperText Transfer Protocol (S-HTTP), the Secure Socket Layer (SSL) protocol, Secure Mosaic, NetScape Navigator, and Intelink-S — have yet to address security labels. However, they each support primitive markings that could evolve in the future.

- Select the session label protocol.
- Select the network label protocol.
- Limit the privileges that can be used when operating on behalf of a remote client.

After these steps have been completed, all of this information must be updated in the MaxSix host's configuration data cache.

3.1.1 Labeling Protocols

Network and session layer protocols specify the "language" that a host "speaks" when sending datagrams or messages to a specified remote host, and the "language" that a host expects to "hear" for incoming datagrams and messages. For two systems to communicate, they must employ the same labeling protocol(s).

3.1.1.1 Network Layer Protocols

MaxSix provides six network layer protocols for passing security attributes: unlabeled, msix, msix_1.0, cipso, ripso, and dnsix. The security attributes that can be transmitted by each network layer protocol are shown in Table 2. The unlabeled network layer protocol allows a MaxSix host to communicate with remote hosts that do not support labels. Security attributes for an unlabeled remote host are assigned by default at the local MaxSix host.

Table 2. Security Attributes for MaxSix Network Protocols

Network Protocol[a]	Possible Security Attributes[b]
unlabeled	———
msix	sl, il, privs, uid, luid, gid, sid
msix_1.0	sl, il, privs, uid, luid, gid
cipso	sl, il, privs, uid, luid, gid
ripso	sl
dnsix	sl

a. Note that the authors use lower-case capitalization (e.g., cipso) to identify options in MaxSix, and upper-case capitalization (e.g., CIPSO) to identify specifications.

b. sl = sensitivity label; il = information label; privs = privileges; uid = user ID; luid = login user ID; gid = group ID; sid = session ID.

A host that uses either the msix or msix_1.0 network layer protocol is called a "token mapping host" because it uses a special token mapping protocol to negotiate security attributes that are associated with a token. A local host labels its network packets with the generated tokens, and the remote host translates the tokens into attribute values that it knows about. These two network layer protocols are being phased out of the MaxSix product because negotiating label formats is more appropriate at the session layer.[13]

The cipso network layer protocol labels datagrams according to the Common IP Security Option (CIPSO) [IETF93]. The ripso network layer protocol labels datagrams as defined in the "Revised IP Security Option" (RIPSO) specification [Kent91]. The dnsix network layer protocol is very similar to the ripso network layer protocol, but specific values it transmits are as defined in the DNSIX 2.1 specification [DIA91b].

A MaxSix host can speak multiple network layer protocols, which is why it can communicate with so many different types of remote hosts, as illustrated in Figure 5.

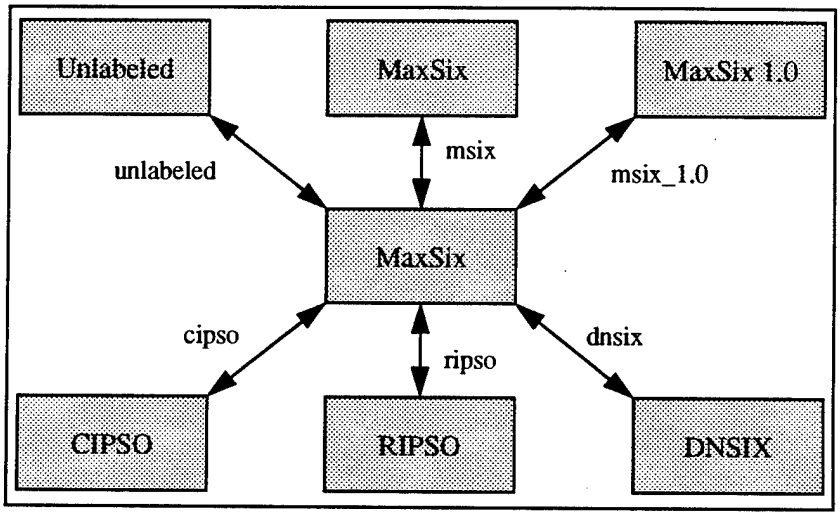


Figure 5. Host Types

Note that while the network layer protocol that two MaxSix hosts generally use is msix, they can speak to each other in any of the other supported protocols as well. This feature is useful if communications between the two pass through a gateway or router that does not understand msix, but may speak one of the other labeling protocols. For example, Figure 6 shows two MaxSix hosts communicating through a router. The router speaks

[13] The TSIG is working on a new approach where all negotiated security attributes are moved to the session layer. As a result, MaxSix is moving toward implementing only three, non-negotiated label formats at the network layer: unlabeled, cipso, and ripso.

RIPSO, so both MaxSix hosts are configured to speak the ripso network layer protocol for the benefit of the router.

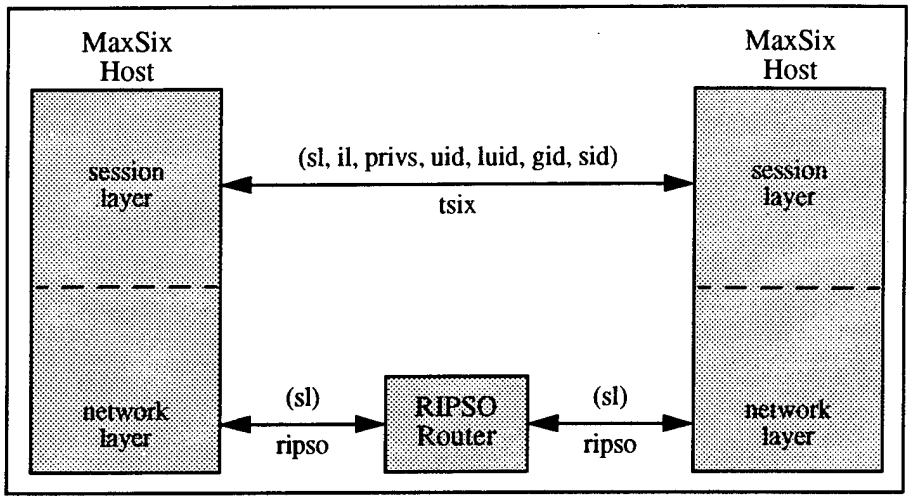


Figure 6. Communicating Through a RIPSO Router

This configuration limits the hosts to exchanging sensitivity labels at the network layer. However, since the hosts also understand a session layer attribute transport protocol, they can still communicate a full set of security attributes at the session layer. Session layer protocols are discussed in the next section.

3.1.1.2 Session Layer Protocols

Attributes can also be transported at the session layer. MaxSix provides three session layer protocols for passing security attributes: standard, dnsix, and tsix.[14] The standard option indicates that the host does not use or support session-layer attribute transport. The dnsix option indicates that the host performs the DNSIX in-band session management handshake, but does not transmit attributes at the session layer. The tsix option indicates that the host performs the DNSIX in-band session management handshake, and uses the Security Attribute Modulation Protocol (SAMP) to transmit security attributes at the message's session layer. Table 3 shows which session and network layer protocols can be combined.

[14] The tsix protocol is based on the TSIG's new Trusted System Interoperability for UNIX (TSIX) specification.

Table 3. Session and Network Layer Protocol Combinations

Session Protocol	Network Protocols
standard	unlabeled, msix, msix_1.0, cipso, ripso, dnsix
dnsix	dnsix
tsix	cipso, ripso, dnsix

3.1.2 Domains

While the protocols specify the formats and rules for exchanging security attributes, the systems using those protocols may actually use different "mappings" to represent the same attribute value. For example, Host A maps the value TOP SECRET to the number 5, while Host B maps the value TOP SECRET to the number 61. If they try to exchange these numbers, neither host can resolve the number into a meaningful classification. Even though two hosts use the same protocol, they cannot communicate if they have such mismatches. The principle is that a host must be able to resolve every attribute value it receives from another host, or it rejects packets with the values it does not understand. Thus, while the network and session layer protocols provide the "language" (syntax) for communicating security attributes, domains (with their associated mappings) provide the "vocabulary" (semantics) for meaningful communication.

A domain defines a specific set of "network representations." If a host understands a domain, then it knows how to communicate with other hosts in that domain by translating its attributes to the domain's network representations on export, and doing the opposite on import. If a host's "native representations" are the same as a domain's network representations, then the process of mapping to and from the domain is a null operation. For two hosts to communicate, they must share a domain that includes every attribute they intend to exchange.

MaxSix provides three types of domains: token_map, cipso, and ripso. The hosts that use a token_map type domain must be capable of doing token mapping (i.e., hosts that use the msix or msix_1.0 network layer protocols or the tsix session layer protocol). A token_map type domain is the most flexible because it can be defined and revised on an as-needed basis if new attributes are defined or new hosts that understand different attributes are added to the network. Attribute values in a token_map type domain can be represented as either integers or ASCII text and can be mapped ASCII-to-ASCII, ASCII-to-integer, or integer-to-integer.

A cipso type domain is called a "domain of interpretation" (DOI).[15] Hosts that use a cipso type domain must use the cipso network layer protocol. Like a token mapping domain, a DOI defines specific representations for a specific set of attributes. A cipso type domain is less dynamic than a token_map type domain because new or different attributes must be defined in a tag and agreed upon before they can be communicated. A cipso type domain is not dynamically chosen by two hosts as a token_map type domain may be. Attribute values are always represented in a cipso type domain as integers.

There is only one ripso type domain. Hosts that use the ripso type domain must use either the ripso or dnsix network layer protocol. The ripso type domain includes some extra fields for RIPSO-specific processing.

3.2 SECURITY LABELS IN OPERATING SYSTEMS

Once a label has been communicated over the network, it is translated into a native representation that is determined by the operating system on the particular host. The label representation used in the operating system can then be translated into component-dependent label representations in database applications, human-readable displays, and storage media. This technical investigation describes security labels in SecureWare's CMW+ operating system.[16] In particular, the translation of a MaxSix network label representation to a CMW+ native representation is described, and the implementation of the CMW+ native representation is also described. Label representations in the three other host components are described in subsequent sections.

3.2.1 Network-to-Operating System Label Translation

Once the network representation of a label has been successfully transmitted from a remote host to the local host, the local host must translate the network representation into a native representation that is understood by the local operating system. MaxSix network representations are translated to CMW+ native representations through a daemon process running on CMW+ that is trusted to not violate the CMW+ security policy. Based on the domain identified in the network and/or session layer protocol, this trusted daemon makes calls to CMW+ to establish the appropriate native representation of the label.

---

[15] "Domain of interpretation" is the best-known, most-accepted term in the trusted product community for a communications protocol domain. The terms "named tag set" or "domain of translation" mean the same thing.

[16] CMW+ is an enhancement of Apple Corporation's A/UX UNIX-based operating system.

3.2.2 Operating System Label Representation

In the CMW+ operating system, tags are 32-bit keys into a database of native representations of security policy attributes, where each policy (e.g., confidentiality, integrity) has its own 32-bit tag space. Each object in CMW+ has a tag pool of up to eight tags (one of which is reserved for tracking information label floating). Tag pools are referenced in one of four different ways: through in-memory and on-disk inodes (e.g., files, directories, special files, symbolic links, pipes and sockets), through tag tables (e.g., semaphores, message queues, and shared memory), through structures for both halves of a pseudo terminal, or through the per-process security data structure (e.g., processes). Each subject has four tags — clearance, current sensitivity label, information label, and user ID data. Each object has three tags — sensitivity label, information label, and access control list (ACL).[17] Permission bits are stored in inodes and in an interprocess communication (IPC) data structure.

Trusted software performs the mapping from tags to native and human-readable label representations. For example, a mandatory access control (MAC) policy daemon maintains a database of tag to native representation mappings. Untagged file systems can only be mounted at a single label; files and directories of untagged file systems are labeled with the sensitivity and information labels associated with the point where the file system is mounted.

There are more than 30 privileges defined in CMW+. Privileges are referenced through inodes (e.g., files) or through the per-process security data structure (e.g., processes). A privilege set consists of a bit vector, where each bit represents a privilege. There are two privilege sets per object (minimum and maximum) and three privilege sets per subject (minimum, maximum, and actual). Privileges only have relevance when an object is executed. Authorizations are very similar to privileges; typically, privileges are associated with files, and authorizations are associated with users and processes.

3.3 SECURITY LABELS IN APPLICATIONS

Since database management systems (DBMSs) are the most prominent trusted applications in current systems, this investigation is based on security labels in the Sybase Secure SQL Server. Sybase noticed that operating systems do two things very well: network label translation, and human-readable label to binary label translation. So, Sybase

[17] Information labels are provided in CMWs as a convenience to users, but are not used for access mediation; access control is enforced by CMWs based on the sensitivity label values.

decided to take advantage of this existing technology. The Sybase Secure SQL Server runs on top of the Sun Trusted Solaris operating system,[18] and depends on the label enforcement mechanisms of the underlying operating system to implement whatever security policy the operating system supports.[19]

3.3.1 Operating System-to-Database Label Translation

The Sun Trusted Solaris operating system[20] has 17-byte labels, provides a routine to translate the binary representation of labels to human-readable form, and provides a routine to perform MAC checks. The Secure SQL Server maps the Trusted Solaris 17-byte representation into its own internal 4-byte representation, called a sensitivity label ID (SLID), and makes calls to these two Trusted Solaris routines. The Sybase Server translates the SLID back into binary form to use the Trusted Solaris routine for MAC checks. The ASCII form of a label is obtained by translating the SLID to binary and the binary to ASCII. Note that the SLID is completely internal to the Sybase Server: no user or other application ever sees it.

3.3.2 Database Label Representation

The Sybase Secure SQL Server is only concerned with sensitivity labels for protecting its information. The server associates one sensitivity label with each row and each subject, three sensitivity labels with each table, and two sensitivity labels with each database. Each table has a high-water mark (maximum) and a low-water mark (minimum) label to limit the sensitivity of the data that may be entered in the table. Each table also has a "hurdle" label, which is the minimum user sensitivity required for the user to read data in the table. The hurdle is intended to address data aggregation. Each database has a high-water mark and a hurdle label.

From the Trusted Solaris perspective, there is only one process for the Sybase SQL Server. From the server perspective, each subject is implemented as a thread of the Sybase SQL Server process. This multi-threaded, single process approach is intended to reduce input/output overhead and deadlock conditions. The Sybase Server has its own engine (like a miniature operating system) to do scheduling and context switching of the subjects (threads).

---

[18] The Sybase Secure SQL Server also runs on top of the Hewlett-Packard HP-UX BLS operating system.

[19] Note, however, that Sybase does not support information labels due to their questionable usefulness.

[20] Sun's Trusted Solaris has an Encodings file for labeling, and a file for secure networking (either Sun 6.0 or MaxSix).

The Sybase Secure SQL Server must be booted at "database system high," and the Server process must be assigned an operating system privilege to perform multi-level IPC (for each of the threads) and a privilege to reset its label (depending on which thread is executing at the time). These privileges allow users at different levels to access the Server at the same time.

The mediation of possible query answers is actually handled on the Sybase Secure SQL Server. Caching is the key. The association of SLIDs to operating system native label representations is cached. Also, once SLID A and SLID B are compared for MAC (by the operating system routine), the result is cached for any future comparisons of SLID A and SLID B.

3.4 SECURITY LABELS ON HUMAN-READABLE DEVICES

This investigation examines the trusted windowing software that is part of SecureWare's CMW+. The X Server and Motif window manager on CMW+ have been enhanced to provide for the labeling of data (i.e., window objects) and the separation of clients. The server is responsible for associating labels with window objects and for enforcing access controls over these objects. The window manager is responsible for user authentication, trusted path enforcement, window labeling, and inter-window data move control. Both the server and the window manager are trusted processes.

The term "client" refers to an X application program that is invoked by a subject to perform some task. Clients connect to the server through sockets. When the server accepts a connection from a new client, the server creates a new client data structure to maintain information pertinent to that connection. This client data structure has been augmented to support three labels: a client sensitivity label, an output information label, and an input information label. Similarly, space has been allocated in the window data structure to support three fields: a sensitivity label, an information label, and an input information label. These labels are obtained from the socket used for making the connection to the window. MAC and DAC protection is provided for all window system objects through polyinstantiation.[21]

The window manager is a unique type of X client that enjoys a symbiotic relationship with other clients. While other clients are generally only concerned with the contents of their windows, the window manager is responsible for the maintenance of

[21] *Polyinstantiation* is a technique where "many instances" of a single object are created, one instance for each sensitivity level.

windows on the screen, but remains ignorant of those windows' contents. The window manager's primary responsibility is to display a sensitivity and information label above each window in a label bar that cannot be removed or altered by unprivileged clients. It also coordinates with the server to intercept and mediate cut-and-paste operations.

A cut-and-paste operation occurs in CMW+ as follows. A "cutting client" sends a cut request to the server. Another (usually different) "pasting client" then sends a paste request to the server. The server forwards the paste request to the window manager. The window manager retrieves the cutting client's sensitivity label, sets the sensitivity level of a special buffer (that it created at startup) to that sensitivity label, and copies the necessary data from the cutting client into the special buffer. The window manager then retrieves the pasting client's sensitivity label to perform the cut and paste mediation. If the cut and paste are allowed, the window manager copies the data from the special buffer to the pasting client's window.

3.5 SECURITY LABELS ON STORAGE MEDIA

This investigation is based on SecureWare's `spax` program, which can be used with CMW+ to backup and restore files on archive storage media, retaining all associated security attribute values for each file. The `spax` program stores security attributes in an extended archive format that is similar to, but not compatible with, ISO 1001 and POSIX 1003.1a. The `spax` format consists of an initial Security Configuration Entry, followed by zero or more archive entries, and terminated by an end-of-archive indicator.

The Security Configuration Entry is always the first entry on the archive media. It has the same format as regular archive entries, except that its data portion contains a description of the security configuration in effect on the system creating the archive. This security configuration indicates to the restoring system which security attributes may accompany the other entries in the archive.

Each archive entry contains a file header. Archive entries for regular files may also contain a data portion and, in some cases, a file trailer. The header and trailer consist of security-relevant information. The data portion consists of regular file data. Permission bits, user ID, and group ID must be recorded in each header. Depending on the security configuration identified in the Security Configuration Entry, additional security attributes can also be recorded in a file header or trailer, such as MAC label, information label, ACL, privilege vector, and whether the component is part of the trusted computing base (TCB).

The values associated with all security attributes are stored on the archive in their human-readable representation.

3.6 SUMMARY OF TECHNICAL INVESTIGATIONS

A label flowing through an information system has many different representations. To achieve interoperability and efficiency in the transfer of security labels between hosts, significant work has gone into developing widely accepted network label representations (e.g., CIPSO, RIPSO, DNSIX, CSL, and SSL). TSIG's MaxSix is an implementation that satisfies several different labeling protocol standards. On the other hand, no work has gone into developing standards for label representations within a host. As described in this chapter, vendors use their own ingenuity to implement labeling within a particular host architecture while achieving other objectives such as performance and transparency. These vendor design decisions make it more difficult to develop security labeling standards within a host. For example, the Sybase Secure SQL Server 4-byte SLID works well with the Sun Trusted Solaris 17-byte label, and the SecureWare `spax` variable length extended archive format works well with the SecureWare CMW+ 32-bit security attribute tag. Given the wide variety of label name space capacities in current label-based products, it may be appropriate to consider an appropriate minimum label name space capacity within a host. Also, it may be appropriate to standardize the human-readable representation of a label to facilitate efficient use of labels for users. Translations are required between each different label representation — labeling protocol standards provide interoperability and efficiency across a network, and the general lack of labeling standards within a host allows vendors the flexibility to achieve necessary host-dependent efficiencies.

4. FUTURE DIRECTIONS IN SECURITY LABELING

The previous chapters have examined relevant standardization issues and representative technologies related to security labeling. It is also important to understand where the DoD information system security emphasis will be in the future and how its focus will affect security labeling in general. DoD's recent adoption of the DoD Goal Security Architecture (DGSA) as part of the Technical Architecture Framework for Information Management (TAFIM) is a shift away from two traditional paradigms: the implicit Multi-Level Security (MLS) paradigm of information protection based on hierarchical classifications and clearances, and the DoD single organization "stovepipe" approach to managing and protecting information. Instead, the DGSA has embarked upon a new and innovative mission-based approach that focuses on non-hierarchical information protection across multiple organizations. A major thrust of the DGSA is to more closely parallel commercial protection requirements. This overlap in requirements is an essential component of achieving cost-effective security in commercial products that can be used in DoD information systems.

The DGSA responds to the DoD vision statement in a new draft information system security policy [DISSP93]. The policy calls for DoD information systems to (1) be sufficiently protected to allow connectivity via common carrier (public) communication systems, (2) be sufficiently protected to allow distributed information processing in accordance with open systems architectures, (3) support processing of information (e.g., sensitive unclassified and classified) under multiple security policies, and (4) support distributed information processing among users employing resources with varying degrees of security protections. Each of these areas can have an effect on the use of security labels.

This chapter defines and relates four fundamental components of *information management* articulated in the DGSA: security policies, controlled entities, information domains, and information systems.[7] These components play a significant part in defining

[7] To facilitate understanding of the DGSA and its potential effect on security labeling, the authors developed an abstract model to address the interaction of the four components of information management in an information system. Appendix A introduces the abstract model, and Appendix B describes the key principles of the DGSA in terms of the abstract model.

effective protection schemes for controlling and managing information and information system resources. They also have a direct effect on security labeling requirements.

4.1 SECURITY POLICIES

Security policies provide direction, define responsibilities, and establish accountability for information management by establishing constraints in the form of rules, conditions, or procedures. Such policies can be developed at different levels of abstraction, ranging from high-level national policy to specific enterprise policies supporting missions and organizations.[8] Security policies become less abstract as they are implemented at lower levels within an enterprise. Ultimately, security policies must be translated into information system requirements for specific security services[9] implemented by a combination of hardware, software, and firmware protection mechanisms. The protection mechanisms, in turn, exercise a degree of control over permissible operations (e.g., READ, WRITE, EXECUTE) on information and information system resources.[10]

4.2 CONTROLLED ENTITIES

Security policies are defined in terms of controlled entities and their associated security attributes. There are three distinct types of *entities* that play a significant role in information management: information objects, information system resources, and end users. These entities form the primitive elements from which all other structures are built.

> **Definition 1.** An *information object*, *o*, is an element of information organized by size or granularity (e.g., bits, bytes, words, pages, segments, fields, records, files) and by type (i.e., data, programs).[11]
>
> **Definition 2.** An *information system resource*, *r*, is a provider of information system services and/or facilities for processing, transmission, and storage (e.g., input/output devices, memory, registers, system functions).
>
> **Definition 3.** An *end user*, *u*, is a consumer or producer of information objects and information system resources.

---

[8] For example, Executive Order 12356 is a high-level security policy that prescribes a uniform system for classifying, declassifying, and safeguarding national security information [EO12356]; DoD Directive 5200.1 [DoD82] references Executive Order 12356 and establishes DoD policy for information security.

[9] The DGSA definition of security services is based on ISO 7498-2 [ISO89] and includes authentication, access control, data integrity, data confidentiality, non-repudiation, and availability.

[10] Security policies must also address protection requirements which provide the environmental boundary of the information system. These requirements involve physical, personnel, and procedural security components which can be selectively combined and controlled.

[11] Note that the DGSA concept of an "information object" is different from the Object Management Group (OMG) concept of an "object." OMG defines an object as a combination of state and a set of methods that explicitly embodies an abstraction characterized by the behavior of relevant requests [OMG93].

Entities are distinguished from one another by associating a set of *attributes* with each entity to describe its properties or characteristics. Attributes associated with an entity may be subsequently instantiated with a value, thus, completing the description. For example, an end user may have a set of attributes consisting of name, social security number, and date of birth, with corresponding values of Bob, 999-99-9999, and 01/01/01. Selected attributes are termed security attributes if they are relevant to the application of a particular security policy (e.g., nondisclosure, integrity, need-to-know).

Definition 4. A *security attribute, a*, is an element of information that is associated with an entity for the purpose of applying a security policy.

Each set of security attributes associated with an entity can be viewed as a "tag" or security label and is necessary for making access control decisions. Given that entities are defined as information objects, system resources, and end users, it is possible to extend these definitions to include security attributes.

Definition 5. A *controlled entity, e*, is an entity that has associated security attributes.

The security attributes associated with each type of controlled entity are based on the protection requirements established by the security policy. The protection mechanisms implementing the security policy then use the security attribute values to either grant or deny permission for requested actions. Having defined security policy and controlled entities, it is now possible to introduce the concept of an information domain.

4.3 INFORMATION DOMAINS

Information domains provide a mission-oriented view of information management in which domains are independent of the physical systems where the actual processing, transfer, and storage occur. An information domain represents a set of information objects and a set of end users constrained by a security policy, as illustrated in Figure 7.

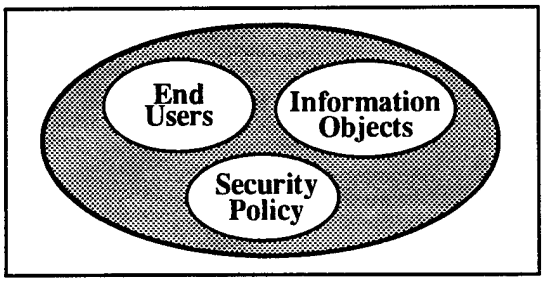


Figure 7. Conceptual View of an Information Domain

Three conditions must exist to successfully employ information domains: (1) a group must have a defined membership, (2) candidate information objects required by the

group must be uniquely identified, and (3) the security policy regarding the protection of the information objects must be agreed to by the members of the group [TAFIM93]. To explicitly define an information domain, it is first necessary to provide a formal definition of a domain security policy.

> **Definition 6.** An *information domain security policy, p*, is a security policy that provides a statement of the criteria for membership of end users in an information domain and the required protection, including conditions of use, for the information objects in the domain.

Using the definitions for information object, end user, and domain security policy, a formal definition of information domain can be constructed.

> **Definition 7.** An *information domain, d*, is a triple consisting of a set of uniquely identified information objects, O, a set of end users, U, and a domain security policy, p, and is denoted $d = (O, U, p)$.[12]

With these definitions, it is possible to state two axioms describing the relationships between controlled entities and information domains.

> **Axiom 1.** An information object cannot reside in two different information domains simultaneously.

Information domains are independent of one another and that membership in one domain with a set of privileges in that domain does not automatically provide privileges in other domains of the same or differing sensitivities. Access to information objects is granted because of explicit end user membership in the domain.

> **Axiom 2.** End users can be members of multiple information domains simultaneously.

In general, there are no restrictions on users being members of multiple information domains other than the criteria established by the appropriate domain security policy.

4.3.1 Intra-Domain Constraints

There are also two intra-domain constraints placed on controlled entities in an information domain.

> **Constraint 1.** Information objects must have the same set of policy-specified security attributes and the same security attribute values.

No security-relevant distinction is made between information objects in an information domain. For example, information objects in domain, d_1, may have a security

---

[12] Note that the concept of an "information domain" is entirely different from the concept of a "domain of interpretation" as described on page 19 in Chapter 3. Both terms tend to use the shorthand of "domain" in conversations and literature, which has been a cause for confusion in the trusted product community.

attribute, a_1, representing *sensitivity*. The instantiated attribute value for each information object must be the same (e.g., *COMPANY CONFIDENTIAL*).

> **Constraint 2.** End users must have the same set of policy-specified security attributes but may have different security attribute values.

Some end users in an information domain may have different privileges than other end users in the domain in accordance with the domain security policy. For example, end users in an information domain, d_2, may have a security attribute, a_2, representing selected access privileges. End user, u_1, may be granted READ access while end user, u_2, may be granted READ and WRITE access. Note that the privilege values for each end user extend across all information objects in the information domain.

4.3.2 Inter-Domain Constraints

The establishment of new missions, mission relationships, and organizations are the types of events that may trigger requirements to share information objects or to transfer[13] information objects between domains [TAFIM93]. Since Axiom 1 precludes an information object from residing in more than one information domain at any given instance, the DGSA imposes certain restrictions on information sharing and transfer to maintain the integrity of information domains resident on end systems. Information objects in different domains can be *shared* in either of two ways. In the first (and arguably the most simple) method, new end users can be accepted into an existing information domain and be granted appropriate privileges. The end user privileges extend to all information objects in the domain. The second method is somewhat more restrictive and can be employed if more protection is required. If there is a need to share some but not all information objects in an information domain, then a new domain can be created with a particular set of information objects (to be shared), a designated set of end users, and a domain security policy.

Information objects can be *transferred* between information domains only in accordance with the domain security policies of each participating domain.

> **Definition 8.** The part of an information domain security policy that establishes the rules, conditions, and procedures for the transfer of information objects among a set of information domains, $\{d_1,...,d_n\}$, $n > 1$, is defined as *information domain import-export rules*.[14]

---

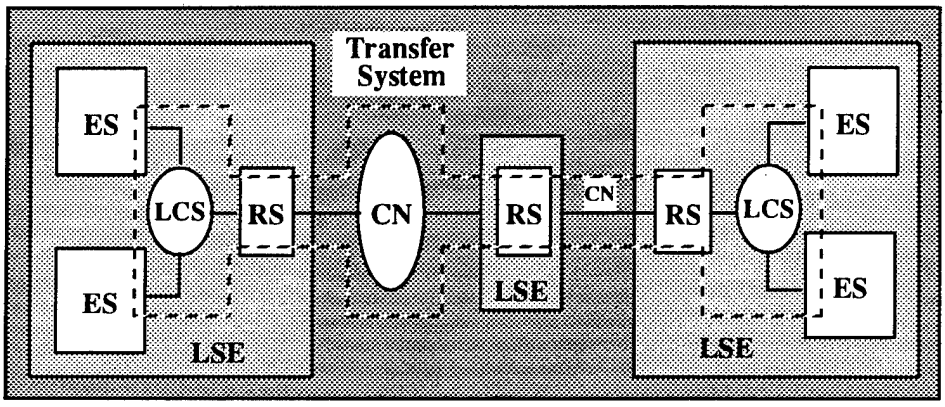
[13] The DGSA defines a transfer operation as either a move operation or a copy operation. According to the DGSA, an information object is relabeled when it is moved or copied from one information domain to another.

[14] The term *information domain import-export rule* does not appear explicitly in the goal security architecture. The concept was, however, articulated by the principal authors of the DGSA in their initial review of material contained in this report.

Import-export rules reflect the agreement between participating domains regarding the inter-domain transfer of information objects. Agreements can be established between two information domains (*bilateral*), between a set of information domains (*community*), and/or they can be made mandatory for all information domains (*unconditional*). Unconditional import-export rules override any bilateral or community rules.

4.4 INFORMATION SYSTEMS

To complete the set of information management components, it is necessary to discuss the physical components where information processing, transmission, and storage occur. These components, in general, make up the constituent parts of an information system. Building on the previous definitions, the components of an information system are considered controlled information system resources. Figure 8 illustrates the DGSA conceptual view of an information system with its fundamental components [TAFIM93].[15]



CN Communications Network
ES End System
LCS Local Communications System
LSE Local Subscriber Environment
RS Relay System

Figure 8. DGSA Conceptual View of an Information System

[15] The DGSA view of an information system maps to the information system model in this report as follows. An end system (ES) includes an operating system, applications, human-readable devices, and storage media. The transfer system (bounded by dotted lines) represents the network. The DGSA also distinguishes between end systems and relay systems. An end system provides traditional information processing capability whereas a relay system provides limited functionality related to information transfer and is employed primarily in support of a transfer system. In reality, end system functions and relay system functions can exist on the same (hardware) platform. For purposes of this report, references to end systems are taken to be inclusive of relay systems as well.

4.4.1 Local Subscriber Environment

The local subscriber environment (LSE) consists of all information processing, transfer, and storage devices under user or organization control [TAFIM93]. These devices can be categorized as end systems, local communications systems, and relay systems.

> **Definition 9.** An *end system, es*, is an information processing system which includes processor and input/output devices (e.g., workstation, personal computer, server, minicomputer, mainframe, disk drive, printer, telephone) that are directly accessible by end users.
>
> **Definition 10.** A *local communications system, lcs*, is a set of communication devices (e.g., ring, bus, twisted pair, coaxial cable, fiber-optic cable) that provides connectivity between end systems under the direct (physical) control of a local subscriber environment.
>
> **Definition 11.** A *relay system, rs*, is a specialized information processing system (e.g., multiplexor, router, switch, cellular node, message transfer agent), not directly accessible by end users, that provides connectivity between LSEs and communications networks.

Having defined the components of an LSE, we now describe the communications infrastructure that connects the end systems together over a distributed network. The successful linkage of multiple end systems can be attributed to communication networks and the transfer system.

4.4.2 Communications Networks and the Transfer System

Communication networks are the devices outside the direct local control of LSEs that are used to connect LSEs. Communications networks, in general, can be implemented using private links or public common carrier links. The transfer system is a logical grouping of communications protocols integrated into end systems, relay systems, local communications systems, and communications networks.

4.4.3 Relationship to Information Domains

Logically, an information domain is a collection of information objects, end users, and a domain security policy; in reality, the controlled entities constituting a domain are distributed across a set of end systems in their local subscriber environments. From a systems implementation standpoint, it is important to be able to relate the logical concept of an information domain to the physical environment where information processing, transfer, and storage occur. The term *projection* is used to describe the actual implementation of an information domain on a specific end system or set of end systems.

Figure 9 provides the four types of possible projections of information domains onto end systems.

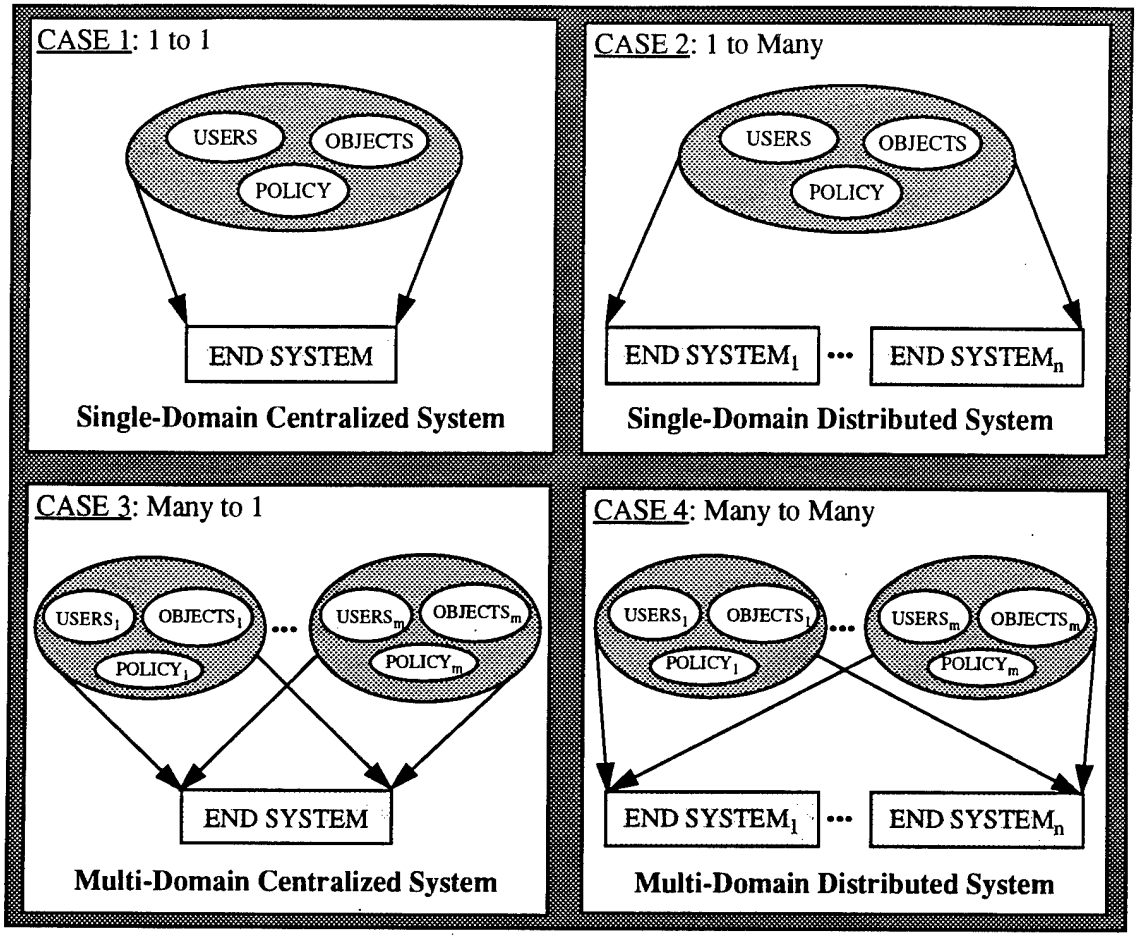


Figure 9. Types of Information Domain-End System Relationships

Case 1, the *single-domain centralized system* projection, is the simplest domain-system relationship representing a single end system processing information objects from a single information domain with no connectivity to other end systems (e.g., using a stand-alone personal computer (PC) or minicomputer to perform payroll functions). Case 2, the *single-domain distributed system* projection, is a one-to-many domain-system relationship representing a network of end systems processing information objects from a single information domain (e.g., using a distributed communications system to perform payroll functions). Case 3, the *multi-domain centralized system* projection, adds a degree of complexity by increasing the number of distinct information domains operating on an end system. This many-to-one domain-system relationship represents a single end system processing information objects from multiple information domains with no connectivity to other end systems (e.g., using a stand-alone PC or minicomputer to perform personnel,

payroll, and mission functions). Case 4, the *multi-domain distributed system* projection, is the most general domain-system relationship. This many-to-many relationship represents a network of end systems simultaneously processing information objects from multiple information domains (e.g., using a distributed communications system to perform personnel, payroll, and mission functions).

Multiple end systems dictate the use of protection mechanisms (e.g., security label protocols) for information objects in transit between end systems. Multiple information domains require appropriate protection mechanism support (e.g., security labels) on the end system to ensure domain separation and enforcement of the domain security policies. The specific details of protecting information in transit and providing information domain separation and mechanism support on end systems are discussed in Appendix B.

5. FINDINGS AND CONCLUSIONS

This report consists of three assessment areas: security labeling standards, security labeling technical investigations, and future directions in security labeling. This chapter summarizes the report through a number of findings from each of the assessment areas. Each finding presented is followed by a short discussion of its potential effect on security labeling. The findings section is followed by a conclusions section consisting of the key points in the report.

5.1 FINDINGS

The first two findings note the shift in paradigm toward the DGSA and its concept of an information domain. The third finding discusses the different representations of a security label in an information system. The fourth finding presents DoD's move toward commercial standards, and the final finding cites the need for a registry to support security labeling.

Finding 1: The DGSA is a paradigm shift from the Multi-Level Security (MLS) approach.

In contrast to the explicit domain membership requirements of the DGSA, the MLS paradigm employs the concept of *implicit* membership (e.g., end-user security clearance) and *dominance* within the context of a mathematical lattice. Resulting products have used security labels to support implementations of lattice-based machines and to conduct MLS operations within end systems. The emerging goal security architecture establishes a significant new paradigm. Information domains, as described in the DGSA, contain objects of the same sensitivity and, in addition, there are no nested relationships between domains. The basic principle of reference mediation is still required by the DGSA; however, the security attributes, rules, and relationships used in the reference monitor would not be the same as those in an MLS system. This finding is particularly important in light of the Paige memorandum [OASD95], which states that new DoD systems development and modernization programs must conform to the TAFIM.

Finding 2: The information domain concept is fundamental to the DGSA.

Information domains provide the central focus for all information management activities within the DGSA and offer a fundamental departure from the traditional methods used to control and protect information. The DGSA changes the multi-level approach to information protection and employs a mission-based approach using information domains. Information domains are intended to be general enough to support any desired information security policy. Object identification, domain membership, security attribute relationships, and import-export rule development all play key roles in the implementation of information domains. The potentially large number of information domains needed to satisfy enterprise requirements may have performance implications that could be a limiting factor of the new architecture.

Finding 2.a: To support the concept of information domains, information objects must be uniquely identifiable.

For potential security label implementations supporting the DGSA, a domain identifier attribute and an object name attribute (and their associated attribute values) can serve to uniformly and uniquely identify any information object. For example, *X.sample* uniquely identifies an object named *sample* within information domain *X*.

Finding 2.b: To support the concept of information domains, explicit domain membership is required of all end users.

Based on the criteria established by the domain security policy, each information domain must have explicitly defined end user membership. End users must be associated with a set of information domains (to which they have authorized membership) and must be assigned a set of privileges per domain. The end system must retain this information on each end user for use in enforcing the domain security policy. Membership in one domain does not necessarily provide any credentials for membership in another domain.

Finding 2.c: To support the concept of information domains, every object in a domain has the exact same security attribute values, and there is no relationship between the attribute values in different domains.

Each information domain has policy-specific security attributes and attribute values that express the actual sensitivity of every information object in the domain. Unlike the MLS paradigm, there is no implicit relationship between objects with similar, yet different, security attribute values. For example, belonging to domain *Y* with security attributes (*TOP SECRET, NATO*) does not necessarily mean a user can access an object in domain *Z* with

attributes (*SECRET, NATO*). Access to the information objects is reduced to a simple test for domain membership.

> **Finding 2.d:** To support the use of information domains, explicit import-export rules must be developed, as part of a domain security policy, to control the flow of information between domains.

The DGSA concept of an information domain with its domain security policy makes the rules, conditions, and procedures for the transfer of information objects between domains explicit. Since there are no implicit relationships between domains, the only way to transfer information between two domains is to establish import-export rules in both domains that agree on how transfers can take place. Thus, establishing an explicit rule in domain *W* that allows information objects to be exported to domain *X* will have no effect unless an explicit rule is established in domain *X* that allows information objects to be imported from domain *W*.

> **Finding 3:** A security label has different representations (e.g., native, network, and human readable) as it travels through the five components of an information system.

Networks function best with some sort of interoperable network representation of a label. Operating systems, applications, and storage media implement their own native representations that provide optimal throughput of label processing on product-specific architectures. Human-readable devices provide human-readable representations that help users understand what the label means when it is printed or displayed. These three representations should be retained in the DGSA to maintain flexibility for vendor product development. In particular, human-readable representations of security attributes must always be available to enforce the security policy on output from human-readable devices. Human-readable representations may also be needed to implement import-export rules and determine eligibility for end-user domain membership. Information domains may simplify network label protocols and may require a minimum label name space capacity.

> **Finding 3.a:** Existing protocol specifications and standards (e.g., CIPSO, RIPSO, DNSIX, CSL, SSL) provide syntax for security attributes to be communicated between end systems over a network, with one of the fields being an identifier (e.g., a "domain of interpretation" or a "named tag set") to establish security attribute semantics.

One of the most valuable results of the different security protocol efforts is the clear trend that syntax should be provided for needed attributes and a field should be reserved to communicate the semantics of the attribute values. Since a domain of interpretation (DOI) is a unique number that conveys certain semantics, it seems reasonable that an information domain identifier, which can also be a unique number that conveys (different) semantics, could be transmitted in the "DOI field" of existing protocols. In the ideal implementation of the DGSA, explicit security attribute values would not need to be passed across a network because the values would be implicit in the information domain identifier; protocol syntax for security attribute values would collapse to a single field (i.e., domain identifier), and access to objects would be reduced to a simple test for domain membership. In the meantime, however, security attribute values still need to be communicated explicitly.

Finding 3.b: The name space capacity for supporting labels varies widely over existing trusted product implementations.

As shown in Table 1 on page 13, the capacities for representing labels range from small name spaces with a fixed set of security attributes to large name spaces with an arbitrary number of security attributes. This wide range of capacities indicates that integration of label implementations is already difficult. As the DGSA concept of information domain becomes more prevalent, name space capacities must address potentially large numbers of information domains and may result in near-term difficulties of transitioning existing architectures to the DGSA.

Finding 4: The DoD community will focus more on commercially developed standards in the future.

As indicated by the Perry memorandum [OSD94], DoD will be relying more and more on commercial standards. As commercial solutions to labeling standardization issues are pursued, however, it is important that those solutions are tempered by DoD policy. This goal might be accomplished by dividing security labeling specifications between a generic standard describing label structure and format (i.e., syntax) and specific standardized implementation guidance for particular communities of interest (e.g., DoD). Implementation guidance may address security labeling semantics and be coordinated with any registration activities.

Finding 4.a: There are three very similar protocols (i.e., CIPSO, CSL, and SSL) for transferring security labels over a network.

The recent DoD-sponsored development of the CSL closely parallels a similar NIST-sponsored effort to develop the SSL. The CIPSO commercial security labeling specification (rapidly becoming a de facto standard), championed by the TSIG, is virtually the same as the CSL. Due to the close cooperation among key agencies (DISA, NSA, and NIST) during the past year as part of a joint working group, there was a concerted effort to make the CSL and SSL virtually identical in structure, and to move toward a single standard for network-based security labels endorsed by both NIST and NSA. This effort has been largely successful and is consistent with the recent memorandum by the Secretary of Defense regarding specifications and standards [OSD94]. The CSL has been published as an military standard (MIL-STD 2045-48501) [DoD95] to serve the current DoD network security labeling requirements. The SSL has been published as a Federal Information Processing Standard (FIPS 188) [NIST94]. However, the adoption of a single national standard has not yet been fully accomplished.

Finding 4.b: The supply of and demand for label-based commercial products are relatively low.

After 10 years of experience promoting the development and use of commercial trusted products based on the MLS paradigm established in the Trusted Computer System Evaluation Criteria [TCSEC85], very few label-based commercial products have been developed and relatively few customers are using such products. The underlying paradigm for labels has now changed to the DGSA, so the promotion of trusted products is going to have to start all over again. The DGSA was developed, in part, to more closely parallel commercial protection requirements. However, commercial industry will not recognize the DGSA until DoD identifies and promotes the significant overlap of DoD and commercial protection requirements. The similarity between CIPSO, CSL, and SSL is a good example of the overlap of security label requirements. Once the common needs have been defined and articulated in terms of the DGSA, users must demand a more robust set of label-based products, and vendors must be encouraged to build such products.

Finding 5: There is no registry and associated organization for security label registration within the DoD.

Without a registry for security labeling, critical information regarding "domains of interpretation" cannot be catalogued for general use by the DoD community. Although there has been considerable effort by DoD to develop a common security label for information transfer, there has not been a parallel effort to develop an appropriate

registration process and to assign a responsible organization to carry out the necessary registration activities.

5.2 CONCLUSIONS

Security labels and security labeling requirements do not exist in a vacuum, but rather support a much larger context involving information system security, in general. The successful use of security labels depends on two key factors: customer *demand* and government and industry *supply*. Customer demand from communities of interest (e.g., financial, medical, personnel) is essential and is created through an educational approach that involves direct participation in the decision-making process regarding the protection of their information and information system resources. Protection requirements can derive from national, governmental, or enterprise policies or be motivated strictly by organization or individual desires. Government and industry supply provides a means to satisfy the customer demand through the following: security labeling *standards*, administrative *organizations* to support security labeling activities (e.g., information domain development, domain registration, security management), and the availability of affordable *technologies* to implement security labeling within enterprise information systems. The cost effectiveness of security management, in general, is the most important factor in the final acceptance of security labels within DoD information systems. The effect of labeling on system performance is also a key consideration in achieving an increase in demand. Therefore, DoD must coordinate with commercial standardization efforts in developing security labeling standards that meet its needs.

The technical investigations of label implementations throughout an information system support two conclusions about standards. First, network label standardization has strong support from commercial vendors and should continue to be pursued in light of the DGSA. Trusted systems developers must be able to interoperate, so existing efforts (e.g., CIPSO, CSL, SSL) should be adapted to support interoperability based on information domains. Second, commercial vendors are much less willing to support security label standardization within an end system (specifically in the operating system). Trusted systems developers should be allowed the maximum flexibility to design cost-effective, label-based technologies that provide optimal local performance. The two exceptions might be standardizing the human-readable representations of security attribute values and considering a minimum label name space capacity that would be required to support information domains.

The recent development of the DGSA has provided a dramatic new paradigm for protecting enterprise information assets. As a central component of the TAFIM, the DGSA offers a basis for all future architectural decisions relating to the design, development, and implementation of secure information systems. The DGSA will change the way security labels are used in DoD information systems and thus affect future label standardization activities. Future security labeling activities or standardization efforts must directly support or complement the overall security objectives of the DGSA. The current MLS paradigm and label-based technologies must be re-examined in detail with respect to the new goal security architecture to ensure that DGSA-based implementations provide equivalent (or better) strength of protection and utility.

It is not likely that commercial products will produce a complete solution to the DoD labeling problem in the near to mid-term. The government must bear the costs and risks associated with developing and proving DGSA-supportive technology, and then must provide the opportunities and paths to transfer this technology to the commercial world. Security labeling standards cannot influence the technology to the degree necessary to motivate widespread development of label-based technologies and products. Instead, DoD should identify and promote the significant overlap of DoD's and commercial industry's need for security labels, and should articulate these common needs in the context of the DGSA. DoD should also emphasize generating demand for security labeling technology from end users by focusing on the new protection paradigm established by the DGSA. Given the commitment to the DGSA as part of the Technical Architecture Framework for Information Management (TAFIM), continuing investment in new security label standards outside of the scope of the new goal security architecture does not appear to be in the best interests of DoD or the Federal Government, and could prove to be counter-productive in the long term.

6. RECOMMENDATIONS AND ACTION PLAN

The set of recommendations presented in this chapter are based on the findings described in Chapter 5. These recommendations address the key issues associated with security labeling and migration to the DGSA. The recommendations form the basis for a follow-on action plan presented at the end of this chapter. The action plan provides specific action items for implementing the recommendations.

6.1 RECOMMENDATIONS

The first three recommendations focus on moving standards and technology toward the DGSA and its central concept of an information domain. The fourth recommendation reinforces the benefits of using non-military standards with supporting DoD-specific implementation guidance being provided separately. The final recommendation emphasizes the need to register security labels.

> **Recommendation 1:** DISA should consider the appropriateness of current and emerging security labeling standards in light of migrating existing technology toward the DGSA.

Recommendation 1 establishes the baseline for all future security label standardization efforts. There is a need to obtain a greater fundamental understanding of the DoD security vision for the future and the role security labels will play in achieving that vision. The DGSA, as part of the TAFIM, will affect every aspect of an information system including communications networks, operating systems, and applications as well as the current MLS paradigm for protecting sensitive information. Thus, there is a need to focus on migrating current security labeling standards to standards which support the DGSA and to devote additional effort to those new security labeling standards which may be required to be fully consistent with the DGSA principles and concepts.

> **Recommendation 2:** As part of its migration strategy to the DGSA, DISA should pursue the use of information domains as the focal point for establishing future security labeling standards.

The information domain is the central construct within the DGSA. As such, it will affect the structure (syntax) of any future security labeling standard. The information domain will also affect how security label structures are interpreted (semantics) and must be fully integrated into a recognized registration process and supporting security label registry. It is essential that an orderly process for information domain development be established. The process should assist DoD functional groups in addressing the following critical topics: (1) definition of an information domain security policy, including import-export rules for inter-domain transfer operations, if required, (2) selection of appropriate policy-specific security attributes (e.g., labels) and attribute values necessary to support the policy, (3) determination of optimal domain size (end users and information objects) for efficiency of operation, (4) development of an information domain registration process, and (5) development of generic information domain types, consisting of a set of pre-defined policy-specific security attributes and a range of allowable attribute values.

Since a new information domain must be created when any new or different security attribute value is required, the number of information domains could potentially be large. Existing trusted products support a wide range of label name space capacities. Vendors should consider a minimum name space capacity for all implementations that is sufficiently large to support DGSA information domains. Current security labeling structures used in the SSL, CSL, and CIPSO may be modifiable to support information domains. Any modification must include the potential for future growth to handle a large and diverse community of users in many different organizations.

Although it appears the DGSA approach allows an information domain identifier to fully represent a set of security attributes within an information system, those attributes still need to be maintained on each appropriate end system to facilitate end user understanding of the labels when information is printed or displayed. Human-readable representations of security attributes associated with an information domain must be attached to output media to allow end users to enforce the security policy outside the information system.

> **Recommendation 3:** DISA should begin to aggressively market the DGSA in order to build necessary support from its customer base and the commercial vendors.

A successful transition to the DGSA will require an extensive marketing effort in several key areas. DoD customers and commercial vendors must be educated in the principles and concepts of the DGSA. Customers must see a transition path to the DGSA and must be able to develop appropriate migration strategies for their current and evolving

information systems. Commercial vendors must see a "market-driven" strategy for producing products supporting the DGSA in general, and security labeling in particular. The transition to the DGSA also depends on the phased development of various supporting technologies in the DGSA-critical areas of strict isolation, absolute protection, security management, and security association (described in Appendix B). Security labeling can play a key role in supporting strict isolation and absolute protection through the employment of separation kernels and security policy decision and enforcement functions on end systems. Labels can also be used in security management operations and in establishing secure virtual channels between end systems through security association.

> **Recommendation 4:** DISA should accelerate the migration of security labeling standards to international, national, or commercial standards, as appropriate, and continue to emphasize the development of DoD-specific implementation guidance in support of those standards.

Adopting a single security labeling standard endorsed by both NSA and NIST can assist in (1) maintaining consistency of the standard over time, and (2) sending a clear, unambiguous message to manufacturers and users of label-based technologies about which requirements to support and incorporate into their product lines. NSA or DISA could then develop DoD-specific implementation guidance in support of the single standard. Guidance that conforms to international, national, or commercial standards, and is targeted to communities of interest would probably change more frequently than the basic security label standard as systems evolve through normal life-cycle activities. These changes would not affect the basic security label standard if the guidance were placed in a separate, supporting publication. Such guidance could be revised as the DoD transitions to the DGSA and supporting security labeling technologies.

The SSL and CSL efforts are prime candidates for this kind of standard-guidance relationship. The SSL and the CSL are essentially identical except for the additional comprehensive implementation guidance that the CSL included for the DoD community. The SSL should be the single national standard and the CSL implementation guidance should be published separately in support of the SSL.

> **Recommendation 5:** Working through the Office of the Secretary of Defense and employing appropriate international and national standards, DISA should accelerate its effort to develop and promulgate a formal security labeling registration process and controlled registry for DoD use that is coordinated with other Federal agencies.

The successful wide-scale implementation and use of security labels depend on a carefully constructed registration process and registry to promote interoperability among information systems. While a formal structure for registration of objects has been defined nationally and internationally [NIST93], specific implementation guidance is necessary for the DoD community to establish what exactly must be registered and how the process is to be managed. DISA should identify and task an organization to be responsible for implementing a DoD registry for security labels.

6.2 ACTION PLAN FOR IMPLEMENTING THE RECOMMENDATIONS

To implement the preceding recommendations, a top-level action plan is proposed. The plan includes key action items, in priority order, which can be initiated in the near term by DISA. General tasks, deliverables, and resource requirements have been identified where possible. It is assumed that a more detailed plan will be developed by the specific organization responsible for carrying out each action item that is undertaken.

> **Action Item 1:** To ensure the successful implementation of international, national, or commercial security labeling standards, begin an initiative to institutionalize the security labeling registration process that includes designating responsible organizations and developing specific implementation guidance for DoD customers.

In coordination with the existing NIST object registry, a responsible organization must be identified and tasked to develop and manage a security label registry for DoD use. A DoD registration process must be defined, and resources must be acquired to review registration applications and manage the operation of the registry database. The structure and format of the CSL and SSL should be some of the first items to be registered. To facilitate interoperability of security labels, the specific numbers assigned to information domains or domains of interpretation should also begin to be registered to provide links to the actual meanings (semantics) of the encoded label. Establishing the registry and the registration process will form a baseline for future discussion of broadly applicable security label usage.

The following tasks should be carried out in support of Action Item 1. Estimated resource requirements (in staff months) for each of these tasks are provided in Table 4.

Task 1: Research and document current registration procedures employed in DoD, national, or international organizations.

Task 2: Select a specific organization to be responsible for DoD security label registration.

Task 3: Develop draft implementation plan and user's guide for DoD security label registration.

Task 4: Coordinate draft implementation plan and user's guide with non-DoD agencies.

Task 5: Develop final implementation plan and user's guide.

Table 4. Resource Requirements (in Staff Months) for Registration Tasks

Activity	1st QTR FY96	2nd QTR FY96	3rd QTR FY96	4th QTR FY96	Total
Task 1	0.5				0.5
Task 2	0.5				0.5
Task 3	2.0	1.0			3.0
Task 4		2.0			2.0
Task 5			2.0		2.0
Total	3.0	3.0	2.0		8.0

Two deliverables should be provided in support of Action Item 1: a "DoD Implementation Plan for Security Label Registration," and a "DoD User's Guide for Security Label Registration."

Action Item 2: To develop broad support and consensus among customers, begin a cooperative effort to form a syndicate of customers from the public and private sectors to assess the current DGSA and initiate the necessary steps to migrate toward a national goal security architecture in support of the National Information Infrastructure (NII).

To achieve the ultimate goal of obtaining a wide variety of cost-effective commercial security labeling products for DoD customers, the DGSA must be recast as a national goal security architecture addressing the broad security requirements of the Federal government and private sector. Exposing the DGSA principles and concepts to many communities of interest will generate a effective dialog on the technical merits of the architecture as well as the ability (or willingness) of the commercial vendors to support the design and development of security labeling and other security-related products. Tapping into the common security requirements of the public and private sectors is an excellent way to demonstrate broad markets to the vendors and provide the necessary motivation for them to invest in the capital-intensive technology development.

The following tasks should be carried out in support of Action Item 2. Estimated resource requirements (in staff months) for each of these tasks are provided in Table 5.

Task 1: Form partnership with NIST and NSA to explore the potential for developing a national goal security architecture from the DGSA.

Task 2: Assess current DGSA and NII security requirements.

Task 3: Develop suggested modifications to DGSA and revise as needed.

Task 4: Coordinate revised DGSA with non-DoD agencies, commercial information technology vendors, and systems integrators.

Task 5: Develop a final draft of DGSA based on community-wide comments and suggestions.

Task 6: Coordinate final draft with DoD agencies and integrate into the TAFIM.

Table 5. Resource Requirements (in Staff Months) for DGSA Tasks

Activity	1st QTR FY96	2nd QTR FY96	3rd QTR FY96	4th QTR FY96	Total
Task 1	1.0				1.0
Task 2	2.0				2.0
Task 3	2.0				2.0
Task 4		3.0			3.0
Task 5			3.0		3.0
Task 6				3.0	3.0
Total	5.0	3.0	3.0	3.0	14.0

One deliverable should be provided in support of Action Item 2: a revised "DoD Goal Security Architecture."

Action Item 3: To show the potential use of standard security labels within the framework of the goal security architecture, begin an initiative to assist DoD functional organizations in researching and developing information domains that express enterprise protection policy requirements, and assess the technical feasibility of implementing those domains on commercially available systems.

The use of security labels within an organization is driven by its security policy. Creating an effective security policy based on an organization's philosophy of protection and defining an appropriate set of security attributes and allowable attribute values are two essential steps in the domain development process. These steps dictate the types of information that will be retained by the information system and associated with the domain. Cataloging the potential set of security attributes and attribute values and developing prototype information domains for implementation in demonstration projects on commercially available operating systems and hardware platforms would facilitate greater

understanding of the use of security labeling in the DGSA and provide valuable insights for systems migration.

The following tasks should be carried out in support of Action Item 3. Estimated resource requirements (in staff months) for each of these tasks are provided in Table 6.

Task 1: Review the fundamental technical concepts of the DGSA in the areas of separation technology, security management technology, and security protocol technology.

Task 2: Establish a technical feasibility criteria for operating systems and hardware architectures to support information domains within the DGSA.

Task 3: Identify candidate commercially available operating systems and hardware architectures that appear to have the potential to support information domains as described by the DGSA.

Task 4: Develop set of prototype information domains using appropriately defined security attributes based on identified security policies.

Task 5: Conduct a detailed investigation of the technical feasibility of implementing information domains on commercially available operating systems and supporting hardware platforms.

Table 6. Resource Requirements (in Staff Months) for Information Domain Tasks

Activity	1st QTR FY96	2nd QTR FY96	3rd QTR FY96	4th QTR FY96	Total
Task 1	1.0				1.0
Task 2	1.0				1.0
Task 3	1.0	2.0			3.0
Task 4		2.0	2.0		4.0
Task 5			6.0	6.0	12.0
Total	3.0	4.0	8.0	6.0	21.0

One deliverable should be provided in support of Action Item 3: a white paper providing the results of the detailed technical investigation.

Action Item 4: To build commercial vendor support for security labeling within the framework of the goal security architecture, begin an initiative to encourage commercial vendors to develop the appropriate demonstration projects for the critical elements of the DGSA.

DoD must bear most of the costs and assume most of the risks of conducting demonstration projects to show that DGSA principles and concepts have commercial

viability. A clear demonstration of potential markets may serve to motivate vendors to build the kinds of security labeling products needed to support the DGSA. Close cooperation with developers of demonstration projects is required to develop and promote the appropriate security labeling standards at the most opportune time. Involvement with ongoing research and development efforts, such as the NSA Synergy Project, can provide significant benefits to standards activities. Leveraging off of current international, national, and commercial standards (e.g., CIPSO and SSL) and standards efforts (e.g., TSIG) is also an effective way to achieve positive results in obtaining the necessary security labeling standards to support the DGSA.

No tasks, deliverables, or resource requirements have been identified for Action Item 4. Such items could be developed at a later date at the direction of DISA.

APPENDIX A.
ABSTRACT MODEL FOR INFORMATION MANAGEMENT

Given the information management components of security policy, controlled entities, information domain, and information system discussed in Chapter 4, and their potential effect on security labeling, an abstract model was developed to unify these components and facilitate understanding of their interaction. This appendix provides a detailed description and derivation of the model and establishes the foundation for discussing labels and the fundamental security concepts articulated in the DGSA. Appendix B provides additional information on specific DGSA concepts and principles.

A.1 DEFINING THE THREE-DIMENSIONAL MODEL VIEW

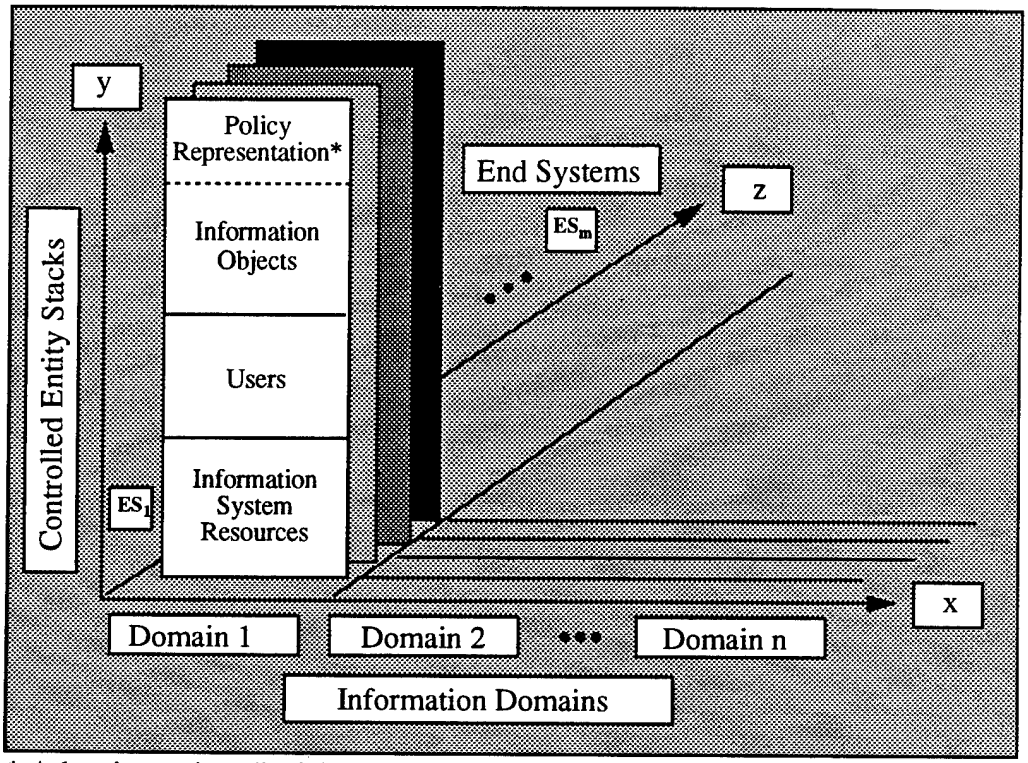
The Entity-Domain-System (EDS) Model is an abstract representation of the DGSA, and provides a framework to describe the management, administration, and implementation issues associated with the transition to the goal security architecture. In general, the EDS Model represents the operation of information domains on a set of end systems in their local environments. The EDS Model provides a three-dimensional view of information management, where the *x*, *y*, and *z* axes represent information domains, controlled entities, and end systems, respectively.[20] Thus, each entry on the *x*-axis represents a specific information domain, each entry on the *z*-axis represents a specific end system, and each entry on the *y*-axis represents a controlled entity (i.e., end user, information object, or information system resource) associated with the domain-system pair.

It should be noted that an information domain security policy is implemented locally on an end system by one or more information objects. As stated previously, a domain security policy must eventually be translated into an information system security policy to enforce the domain security policy on an end system. As such, the domain security

---

[20] While the EDS Model uses the term "three-dimensional coordinate system" to model the components of information management, each entry on the *x*-axis, *y*-axis, and *z*-axis is a specific entity represented as a volume and should not be interpreted in the mathematical context of actual ordered coordinates along axes.

policy becomes an information object, or set of objects, within a specified information domain when projected onto an end system.[21] Figure A-1 illustrates the three-dimensional view of the EDS Model. Specific derivations of each component of the model are provided in the following sections.



* A domain security policy is implemented on an end system by one or more information objects.

Figure A-1. The Entity-Domain-System Model

A.2 DERIVING THE COMPONENTS OF THE MODEL

In deriving the components of the EDS Model, it is first necessary to address the set of controlled entities defined by the *y*-axis, or the set of controlled entities associated with a specific end system and information domain. To accomplish this task, several additional definitions are needed to distinguish the respective types of controlled entities within the model.

---

[21] Information domains containing security policy-related information objects are discussed in greater detail in Section B.2 on page B-5.

A.2.1 Specific Information Domains

Having defined both the logical construct of an information domain and its projection onto a set of one or more end systems, it is necessary to describe, in further detail, a domain in terms of its physical distribution across end systems. The logical components of an information domain (i.e., end users, information objects, and domain security policy) must be distinguished from the subsets of controlled entities that represent the components on specific end systems. That is, each end system servicing an information domain has controlled entities operating in its local environment that are a subset of the totality of controlled entities implementing the logical domain components. The projection of an information domain onto a set of one or more end systems produces a set of one or more specific information domains.

> **Definition A.1.** A *specific information domain*, d_{es}, comprises a set of uniquely-identified information objects, O_{es}, a set of end users, U_{es}, and a domain security policy, p_{es}, operating on a specific end system, *es*, and is denoted $d_{es} = \{O_{es}, U_{es}, p_{es}\}$.

Let d represent an information domain projected over a set of end systems, $ES = \{es_1,...,es_q\}$, $q > 0$. Let es_j represent the jth end system from ES. Then information domain, d, can be expressed in terms of a set of specific information domains, $\{d_{es(1)},...,d_{es(q)}\}$ as

$$d = \{O_{es(1)} \cup \ldots \cup O_{es(q)}, U_{es(1)} \cup \ldots \cup U_{es(q)}, p_{es(1)}, \ldots, p_{es(q)}\}$$

where each $O_{es(j)}$, $U_{es(j)}$, and $p_{es(j)}$ from $d_{es(j)}$ represents the projection of domain, d, onto end system, es_j.[22] The specific information domain security policies, $\{p_{es(1)},...,p_{es(q)}\}$, are implemented on each end system, $\{es_1,...,es_q\}$, servicing information domain, d, as a unique set of information objects.

The distribution of controlled entities from an information domain to a specific information domain must be accomplished in accordance with the following axioms which appropriately extend and amplify Axioms 1 and 2 on page 28.

> **Axiom A.1.** An information object may only exist on one end system at a time.

Axiom A.1 implies that no two information objects are exactly the same. The contents of an object container (e.g., a file) could be identical to the contents of another object container, but the container would be labeled uniquely.

[22] It is possible that the projection of an information domain, d, onto a set of end systems, ES, reflecting the dynamic nature of information domains, may result in some specific information domains containing no information objects from d at a particular instant of time. The same situation can occur with end users.

Axiom A.2. End users may operate on more than one end system simultaneously.

Axiom A.2 implies that, in general, there are no restrictions on end users operating on multiple end systems other than those restrictions and controls dictated by the information domain security policy as implemented on particular end systems.

A.2.2 Controlled Entity Stacks

Given the definition of a specific information domain, D_{ES}, it is possible to fully define the set of controlled entities associated with an end system.

Definition A.2. A *controlled entity stack*, *ce*, is a set of controlled entities obtained by holding information domain, *d*, on the *x*-axis and end system, *es*, on the *z*-axis constant, and taking the vertical projection along the *y*-axis.

A controlled entity stack[23] can be viewed as the instantiation of an information domain on an end system. The derivation of a controlled entity stack is a function of two variables, an information domain, *d*, and an end system, *es*, denoted as

$$ce_{(d_i,es_j)} = \{o_1, \ldots, o_n, u_1, \ldots, u_m, p_{es}, r_1, \ldots, r_t\}$$

where n, m, and $t \geq 1$. In essence, the controlled entity stack groups a subset of the controlled entities from an information domain assigned to a specific operating environment (i.e., a specific information domain) with an appropriate set of information system resources, $R_{ES} = \{r_1,\ldots,r_t\}$, supporting that environment (e.g., CPUs, disk drives, printers).[24] The result is the controlled entity stack for the *i*th information domain, d_i, operating on the *j*th end system, es_j.

In the EDS Model, the controlled entity stack is the conceptual structure that unifies the logical information domain with the physical information system. It is possible to establish an axiom describing the relationship between information system resources and controlled entity stacks.

---

[23] A *controlled entity stack*, as defined in this paper, has no relation to the traditional concept of a stack or a queue as data structures in a computing system. While a controlled entity stack can be considered a data structure for modeling purposes, there is no implied order or entry/exit criteria for elements in the stack. A controlled entity stack is dynamic in nature, and represents a group of controlled entities associated with one another because of their membership in a particular information domain and their residing in a specific operating environment where information processing, transfer, and storage occur. Thus, the controlled entity stack changes over time with entities being added and removed as required.

[24] The distinction between R_{ES} and *es* is noteworthy. In the EDS Model, an end system, *es*, contains a complete set of information system resources, R_{ES}, that supports its local operating environment.

Axiom A.3. An information system resource can support (be a member of) more than one controlled entity stack, but only in a time-multiplexed manner.[25]

Thus, at any point in time, a subset of information system resources, $R_{ES} = \{r_1,...,r_t\}$, $t \geq 1$, is dedicated to supporting a particular end user, u, in a specific information domain, d, on a particular end system, es.

A.2.3 Information Domain and End System Views

It is possible to define two distinct views of information management. Let d_i represent the ith information domain from a set of information domains, $D = \{d_1,...,d_v\}$, $v > 0$, projected over a set of end systems, $ES = \{es_1,...,es_q\}$, $q > 0$. Let es_j represent the jth end system from ES.

Definition A.3. An *information domain view* of information management is defined by a set of controlled entity stacks, CE, obtained by holding information domain, d_i, on the x-axis constant (i.e., selecting a particular information domain), and taking the vertical projection along the y-z plane, and can be expressed as

$$CE_{d_i} = \bigcup_{1 \leq j \leq q} ce_{(d_i,es_j)}$$

or the union of controlled entity stacks across information domain, d_i.

Conversely, a complementary view can be defined by varying projections along axes.

Definition A.4. An *end system view* of information management is defined by a set of controlled entity stacks, CE, obtained by holding end system, es_j, on the z-axis constant (i.e., selecting a particular end system), and taking the vertical projection along the x-y plane, and can be expressed as

$$CE_{es_j} = \bigcup_{1 \leq i \leq v} ce_{(d_i,es_j)}$$

or the union of controlled entity stacks across end system, es_j.

A.3 INTERPRETING THE EDS MODEL

The EDS Model is a generic model that can be used at various levels of abstraction. For example, the three-dimensional model can represent an enterprise as large as the DoD or an organization as small as an individual division in a private-sector company. For the former, the model represents all information domains and end systems within the DoD. For

---

[25] The phrase *time-multiplexed manner* implies that at a particular instant in time, an information system resource (e.g., CPU, memory, disk drive, or printer) is dedicated to a particular end user as a member of a controlled entity stack.

the latter, the model represents only those information domains and end systems owned by the respective division. It is also possible to view the EDS Model at the most abstract level where the dimension along the x-axis represents the universal set of all information domains, the z-axis represents the universal set of all end systems, and the y-axis represents the universal set of all controlled entities.

APPENDIX B.
RELATING DGSA CONCEPTS TO THE MODEL

Chapter 4 and Appendix A provided descriptions of the components of information management and an abstract model to facilitate understanding of the complex interactions between controlled entities, information domains, and information systems. It is now possible to describe the fundamental security concepts articulated in the DGSA and interpret those concepts with respect to the abstract model.

B.1 SECURITY MANAGEMENT

Within the information systems environment, there must be an appropriate supporting infrastructure to effectively manage the information domains that are resident on end systems. Security management provides the security services necessary for the protection of controlled entities on an end system in accordance with applicable information domain security policies. The security management information necessary to implement the supporting infrastructure must be separated from the controlled entities within the supported information domains. In general, this separation is accomplished by placing all critical security management information in distinct information domains. Security management information is maintained as sets of controlled entities in security management information domains.

A security management information domain consists of the same types of components as a general purpose information domain, including a set of information objects, a set of end users, and a security policy. The information objects consist of security management information (data and programs) necessary to provide appropriate protection for the domain. Each set of security management objects supporting a particular information domain is contained in a logical repository called a *security management information base (SMIB).*[25] End users within the security management information domain are typically privileged users such as systems administrators or system security officers. The domain security policy provides a statement of the criteria for membership of (privileged) end users in the security management information domain and the required

protection for the information objects in the domain. Figure B-1 illustrates a conceptual view of a security management information domain (SMID) with its constituent components.

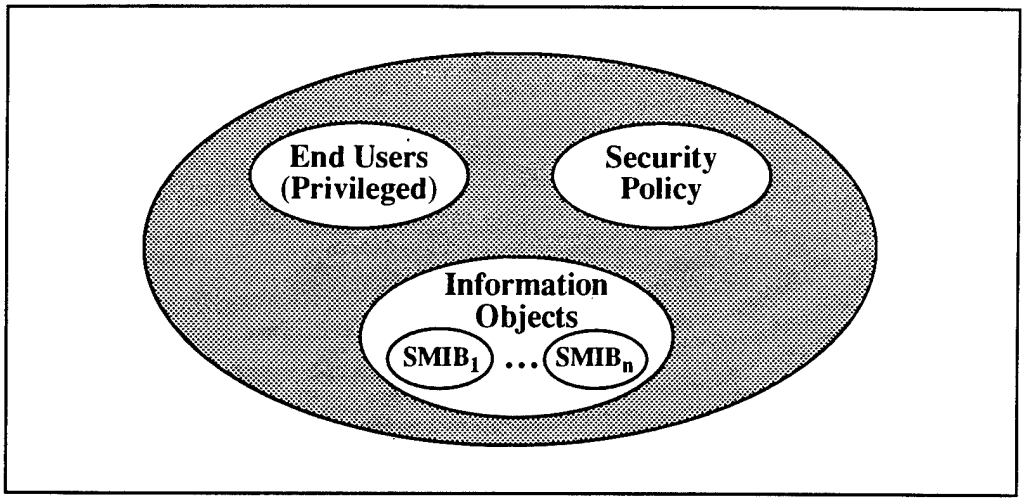


Figure B-1. Conceptual View of a Security Management Information Domain

Given the general description of a security management information domain, it is possible to construct a more formal definition.

> **Definition B.1.** A *security management information domain*, d_m, is a triple consisting of a set of uniquely identified information objects, O_m, grouped into security management information bases (SMIBs), a set of privileged end users, U_m, and a management information domain security policy, p, and is denoted $d_m = (O_m, U_m, p)$.

Figure B-2 provides the three general ways that a security management information domain can be employed to support general purpose information domains [TAFIM93]. Case 1, *single security management domain to single information domain*, describes the support relationship in which one security management information domain is dedicated to supporting one information domain. The security management information domain contains only one SMIB. Case 2, *single security management domain to multiple information domains*, describes the support relationship in which one management information domain supports a set of n information domains, $n > 1$. The security management information domain in this case contains n SMIBs (i.e., one SMIB per

[25] SMIBs supporting information domains contain information domain policy rules, end user registration information, end user authentication criteria (e.g., strength of mechanism required), end user security attributes, and security service and security mechanism requirements for inter-domain information transfers. Programs in execution which operate on the SMIB are called *security management application processes (SMAP)* and are considered information system resources.

information domain supported). Case 3, *embedded security management information domain*, is a special case in which all of the security management information is embedded in the information domain being supported.[26] As in Case 1, the information domain contains a single SMIB.

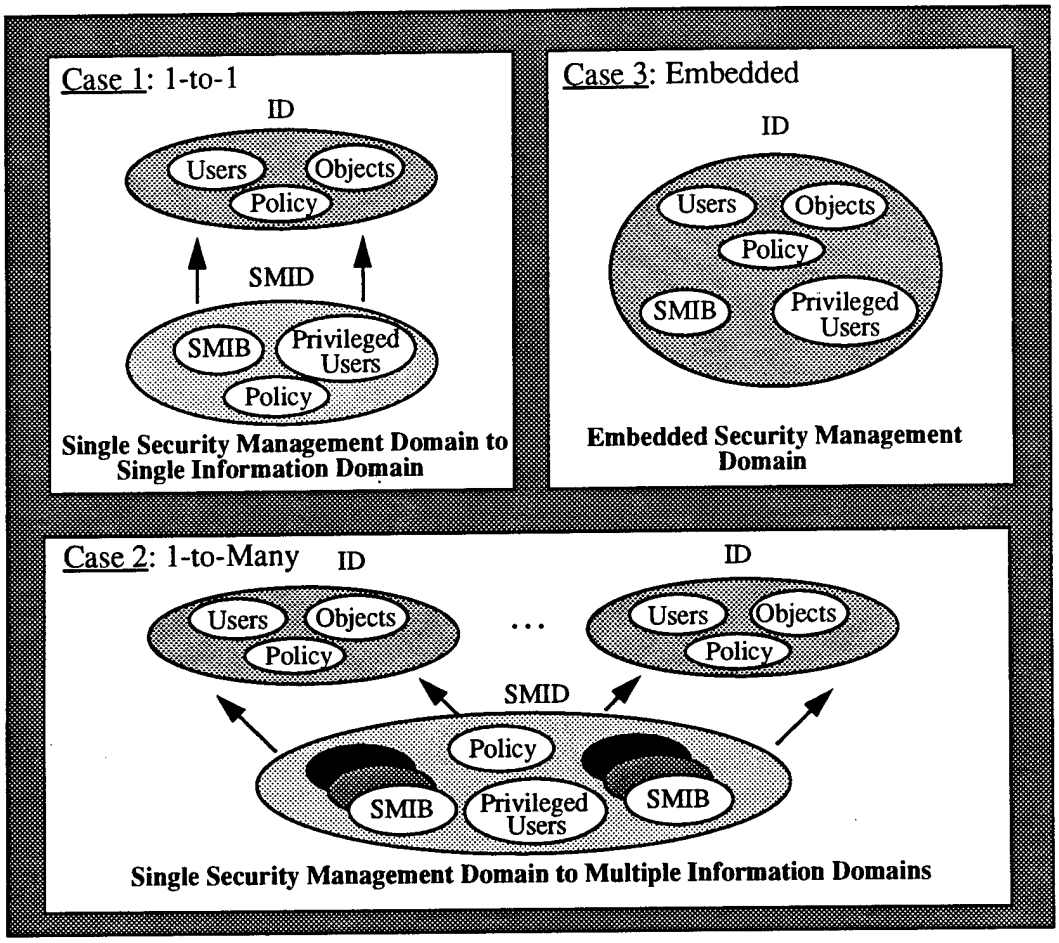


Figure B-2. Security Management Domain Support Relationships

Depending on operational requirements, the components of a security management information domain can be distributed across multiple end systems. Thus, as with information domains in general, security management information domains can be projected onto a set of end systems producing a set of specific information domains for security management.

---

[26] A security management information domain, for example, may contain its own security management information.

Definition B.2. A *specific security management information domain*, d_{sm}, is a triple consisting of a subset of uniquely identified information objects, O_{sm}, from O_m (i.e., a portion of the security management information bases necessary to support a set of information domains), a subset of privileged end users, U_{sm}, from U_m, and a specific end system implementation of a management information domain security policy, p_{es}, operating on a specific end system, and is denoted $d_{sm} = (O_{sm}, U_{sm}, p_{es})$.

The controlled entities from the specific security management information domain become part of the controlled entity stack for the particular end system designated to receive the security management information as a result of the domain projection.

In addition to providing security management information domains, end systems must be capable of providing separate security management for shared system resources (e.g., security functions, services, mechanisms, devices, memory, registers). Thus, in addition to the domain security policies, there must be a separate security policy established for the end system that addresses the management of shared system resources.

Definition B.3. An *end system security policy*, es_p, is a security policy that specifies how sharing of information system resources (e.g., security functions, services, mechanisms, devices, memory, registers) is accomplished on an end system, *es*, in support of a set of information domains, D, resident on *es*.

The end system security policy is separate and distinct from the security management information domain security policy and focuses solely on the sharing of information system resources. End system security policies must be controlled in the same manner as information domain security policies, and therefore, exist within an end system information domain.

Definition B.4. An *end system information domain*, es_d, is used to control and manage system resources (e.g., login procedures, security management information domains, and multi-domain objects[27]) resident on an end system, *es*.

In addition to the SMIBs supporting each information domain, each end system has a unique SMIB to control the shared system resources.[28] The end system SMIB, along with the end system security policy, privileged end users, and information system resources,

---

[27] Multi-domain objects are special composite virtual information objects created from constituent objects from different information domains and are strictly limited in their use. Multi-domain objects are discussed in greater detail in Section B.5.

[28] SMIBs supporting end systems typically contain end system security policy rules, management information for security services and mechanisms, and management information for supporting services and mechanisms (e.g., auditing, alarm reporting, key distribution, security contexts) [TAFIM93].

become part of the controlled entity stack for a particular information domain, d, and end system, *es*, augmenting the controlled entities previously accrued from the projection of the domain onto the end system. Thus, the controlled entity stack is the unifying structure that brings together the controlled entities from the (logical) information domains and the repository of (physical) information system resources.

B.2 STRICT ISOLATION

To support multiple information domains on a single end system,[29] the DGSA employs a protection strategy called strict isolation [TAFIM93].

> **Definition B.5.** *Strict isolation* is the logical and/or physical separation of a set of information domains, $D = \{d_1,...d_n\}$, $n > 1$, and their associated controlled entities from other domains on a particular end system, *es*.

End system, *es*, through its underlying hardware features and operating system functions, must provide appropriate security mechanisms to enforce the separation between domains in such a manner as to satisfy the requirements of each information domain security policy. The DGSA mandates that the strict isolation between information domains be enforced as a default condition unless an explicit relationship between domains is defined by an information domain security policy through a set of import-export rules.

In implementing strict isolation, it is necessary to confine information objects used by end users to their information domains. This confinement is accomplished via security contexts.[30] Recalling that an information domain is defined as $d = (O, U, p)$, a formal definition for security context can be constructed.

> **Definition B.6.** A *security context*, *sc*, is a subset of controlled entities from a controlled entity stack, *ce*, supporting a particular end user, u, on an end system and consists of selected information objects from O_{es}, a specific end

---

[29] The diversity of missions and information protection requirements may result in the proliferation of information domains, each with its own security policy and protection requirements. While this statement implies that there will be many unique information domain security policies, in reality, a number of domain security policies may be very similar. Thus, when new information domains are created, there could be significant potential for re-use of existing security policies — only changing the policy where necessary to meet the specific protection requirements of the new domain.

[30] The DGSA also includes *security doctrine* as part of a security context. Security doctrine addresses the specific conditions of use for a particular component, facility, or system. The specification of conditions of use within a specific environment is intended to complement the protection provided by the hardware, firmware, and software mechanisms as part of the original product design [TAFIM93]. The topic of security doctrine is beyond the scope of this paper.

system implementation of the information domain security policy, p_{es}, and selected information system resources from R_{es}.

The security context, *sc*, can be expressed as

$$sc_{(u, d, es)} = \{o_1, \ldots, o_n, u, p_{es}, r_1, \ldots, r_t\}$$

where n and $t \geq 1$. A security context is closely related to the concept of a user or system process space implemented by an operating system and selected hardware features. It is the collection of all data, programs, and system resources (e.g., hardware, system software, end user application software, and information) necessary to support a particular end user or system function operating in a particular information domain in accordance with a specific domain security policy. Thus, an end user must have a specific security context established for each information domain where processing is required. An end system may maintain multiple security contexts, depending on the number of information domains and end users supported. The operating system must maintain all essential information required to enforce security context separation.[31] Figure B-3 illustrates a simplified conceptual view of a security context.[32]

B.3 ABSOLUTE PROTECTION

To support intra-domain communication over multiple end systems, it is necessary to extend the protection strategy of strict isolation to effectively satisfy the domain security policy on each of the end systems where the information domain operates. The strategy employed by the DGSA to support information domain operation on multiple end systems is absolute protection [TAFIM93].

> **Definition B.7.** For a given information domain, *d*, projected onto a set of end systems, $ES = \{es_1,\ldots,es_q\}$, $q > 1$, the condition of achieving the minimum required strength of protection (for the domain) on each end system, es_i, from *ES*, is *absolute protection.*

Using this definition, it is possible to formalize the concept of absolute protection for an information domain, *d*, and a set of end systems, *ES*. Let special function, $protected_{(d,es)}$, describe the condition of achieving the minimum required strength of

---

[31] The approach described in the DGSA calls for the separation of security contexts through the employment of a *separation kernel* similar to that defined by Rushby [Rushby84].

[32] Figure B-3 is intended to convey the fact that multiple security contexts can exist on an end system obtaining users, objects, and policy from the same information domain. Nevertheless, each security context is distinct and maintains strict isolation from other security contexts.

protection for information domain, *d*, on a specific end system, *es*. Then, absolute protection can be defined by the following expression:

$$protected_{(d, ES)} = \{protected_{(d, es_1)} \bullet \ldots \bullet protected_{(d, es_q)}\}$$

where $\bullet$ represents a logical *and* operator in boolean algebra.

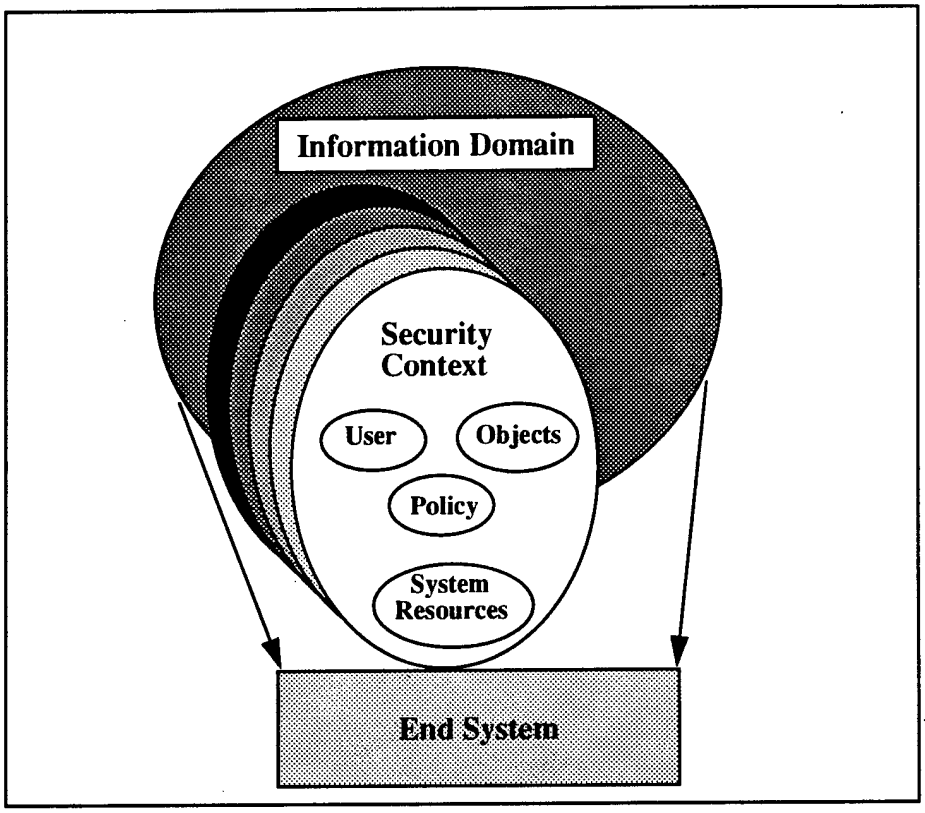


Figure B-3. Simplified Conceptual View of a Security Context

Absolute protection requires that each end system supporting an information domain possess the minimum requisite strength of protection necessary to ensure the domain security policy is adequately enforced. Consistency of protection across end systems is a function of the effectiveness and correctness of security mechanism implementation. While the DGSA mandates that the strength of mechanisms be consistent across end systems, the goal architecture allows maximum flexibility in the actual design and implementation of security mechanisms. There is also no inherent restriction on providing additional protection for an information domain, given that the minimum required strength of protection (on the end system) has been achieved in accordance with the domain security policy.

It is the objective of the DGSA transfer system to create an environment in which separate security contexts operating from different controlled entity stacks on physically distributed end systems can communicate securely as if the security contexts resided on the same end system. The need for secure, distributed communications between end systems supporting the same information domain produced the requirement for distributed security contexts. Let d represent an information domain and ES represent a set of end systems where $ES = \{es_1,...,es_q\}$, $q > 1$. Let es_i and es_j represent the ith and jth end systems from ES and let $d_{es(i)} = (O_{es(i)}, U_{es(i)}, p_{es(i)})$ and $d_{es(j)} = (O_{es(j)}, U_{es(j)}, p_{es(j)})$ represent the specific information domains derived from the projection of d onto es_i and es_j, respectively.

> **Definition B.8.** A *distributed security context*, *dsc*, is a set of controlled entities from a set of controlled entity stacks, $CE = \{ce_{(d,es(i))}, ce_{(d,es(j))}\}$, supporting a particular end user, u, in a specific operating environment and consists of selected information objects from $O_{es(i)}$ and $O_{es(j)}$, specific end system implementations of information domain security policies, $p_{es(i)}$ and $p_{es(j)}$, and information system resources from $R_{es(i)}$ and $R_{es(j)}$.

A distributed security context can be interpreted as joining two end system security contexts that have been established for an end user in support of the same information domain, thus giving the end user use of information domain objects that reside on remote end systems using the information system resources of the local and remote systems. It is possible to formally define a distributed security context as

$$dsc_{(u, d, ES)} = sc_{(u, d, es_i)} |X| sc_{(u, d, es_j)}$$

where |X| is a special infix operator representing the *joining* of two security contexts. Generalizing the above equation, an n-way distributed security context can be expressed as

$$dsc_{(u, d, ES)} = sc_{(u, d, es_i)} |X| ... |X| sc_{(u, d, es_n)}$$

where n represents the number of security contexts participating in the distributed security context.[33]

---

[33] Conceptually, the formation of a distributed security context can be n-way (i.e., involving more than two security contexts). However, the current state of information system security technology and practical considerations limit the formation of distributed security contexts to pair-wise associations.

B.4 SECURITY ASSOCIATION

Distributed security contexts require a secure virtual channel to exist between end systems in order to enforce absolute protection. This secure virtual channel is established using a set of security mechanisms called a security association.

> **Definition B.9.** A *security association* is the set of all information system resources (i.e., security and communications protocols, security functions, security services, and mechanisms) employed to securely link two distinct security contexts, sc_i and sc_j, on different end systems, es_i and es_j, supporting the same information domain, d.

An effective security association extends the protection provided by the participating end systems through the transfer system across the communications network. Critical information needed to establish a security association is exchanged between end systems using a Security Association Management Protocol [TAFIM93].[34] The protocol information is contained in the SMIBs of the participating specific security management information domains supporting the logical domain.[35]

B.5 INFORMATION SHARING AND TRANSFER

The DGSA restricts the sharing and transfer of information between domains in order to maintain the integrity of information domains resident on end systems. Information objects in different domains can be *shared* by accepting new end users into an existing information domain or by creating a new domain with a particular set of information objects (to be shared), a designated set of end users, and a domain security policy. Information objects can be *transferred* between information domains only in accordance with import-export rules that are part of the domain security policies for each participating domain. There are three fundamental types of import-export rules that can be expressed by an information domain security policy: bilateral, community, and unconditional.

- *Bilateral* import-export rules are established between two information domains and result in a set of rules that reflect the constraints on the transfer of information objects between the two domains. The import-export rules must be part of the domain security policies of each participating information domain.

---

[34] Note that the use of the term Security Association Management Protocol is different from the Security Attribute Modulation Protocol discussed in Chapter 3.

[35] A SMIB data structure, the Agreed Set of Security Rules (ASSR), provides domain label information as well as cryptographic keying information for the security association [TAFIM93].

- *Community* import-export rules are established among a set of information domains where the end users collectively decide on the set of rules that control the inter-domain transfer of information objects. As in the bilateral case, the established import-export rules for groups or communities must be a part of the domain security policies of all participating information domains.
- *Unconditional* import-export rules establish the specific rules for inter-domain transfer of information objects that must remain in effect at all times under any circumstances. Note: Unconditional import-export rules override any bilateral or community rules.

There are also a few constraints employed by the DGSA with respect to inter-domain transfer of information objects.

> **Constraint B.1:** An end user transferring information objects between domains must be a member of both the source and destination information domains and must possess appropriate privileges (including release authority).

> **Constraint B.2.** Inter-domain transfers of information objects can occur only if the source and destination information domains are resident on the same end system.

Constraint B.2 implies that inter-domain transfers cannot occur among distributed systems (i.e., end systems and relay systems). The rationale for including this restriction involves the definition of a security association as discussed in Section B.4. While an individual end system uses appropriate protection mechanisms to ensure domain separation, a distributed system must employ other mechanisms (e.g., cryptographic mechanisms) to protect information in transit between end systems. An essential requirement in employing cryptographic mechanisms is the sharing of key material and other supporting information. Sharing key information from different information domains is much more difficult and results in additional complexity in communications and security protocols. Thus, for implementation reasons, the DGSA restricts inter-domain information transfer across multiple end systems. Given the inter-domain restriction described by Constraint B.2, it is possible to provide a complementary view of information transfer restrictions between end systems.

> **Constraint B.3.** Distributed (end) systems can support information transfer operations only within a single information domain.

This constraint implies that any transfer of information objects between two end systems, es_i and es_j, must occur within the context of a single information domain, d. The rationale for including this restriction is, in essence, the same as for Constraint B.2.

It is possible to use the EDS Model to graphically describe some of the constraints outlined above. Referring to Figure A-1, consider the x-z plane or space of information domains and end systems. Constraint B.2 limits information flow to the x-axis only, that is, given a particular end system, es_i, information objects can be transferred between any information domain in the set of domains, $D = \{d_1,...,d_n\}$, $n > 1$, resident (or hosted) on es_i in accordance with domain security policies. Constraint B.3 limits information flow to the z-axis only, that is, given information domain, d_i, information objects from d_i can be transferred between any end system in the set of end systems, $ES = \{es_1,...,es_m\}$, $m > 1$, in accordance with the domain security policy.

B.6 MULTI-DOMAIN INFORMATION OBJECTS AND POLICIES

To support mission-related activities, it may be necessary to combine information objects from different information domains into composite information objects. The DGSA recognizes this requirement but places severe restrictions on how this composition is accomplished. Through the security management facilities on the end system, end users can create the perception that a set of information objects, O_1, from information domain, d_1, and a set of information objects, O_2, from information domain, d_2, form a single composite information object. These information objects are, in reality, virtual objects, and are referred to as *multi-domain objects*. Multi-domain objects can only be used to print, display, or transfer information between end systems from multiple information domains.

Multi-domain information objects are virtual objects in the sense that a real composite object is never actually created on the end system. A set of pointers (one possible implementation) located within the end system information domain could be used to provide address locations of those constituent objects required to form the composite information object. It is important to maintain the security concept of strict isolation during the process of using multi-domain objects, and it is always the case that the constituent objects of the multi-domain object must be protected in accordance with the security policies of the information domains where the objects reside.

> **Constraint B.4:** A multi-domain object can only exist on end systems that support all of the information domains contributing to the formation of the composite object.

Constraint B.4 is employed to facilitate the transfer of a multi-domain object between end systems. The requirement to support all information domains contributing to the formation of the composite information object ensures that a separate set of distributed security contexts can be established for each constituent component of the multi-domain object. That is, the security association between end systems must establish a separate set of security contexts for each component of the composite object deriving from a different information domain, thus maintaining strict isolation between domains while the information objects are in transit across the network. This is analogous to having multiple secure channels between the end systems. Figure B-4 illustrates a conceptual view of a multi-domain object transfer between end systems.

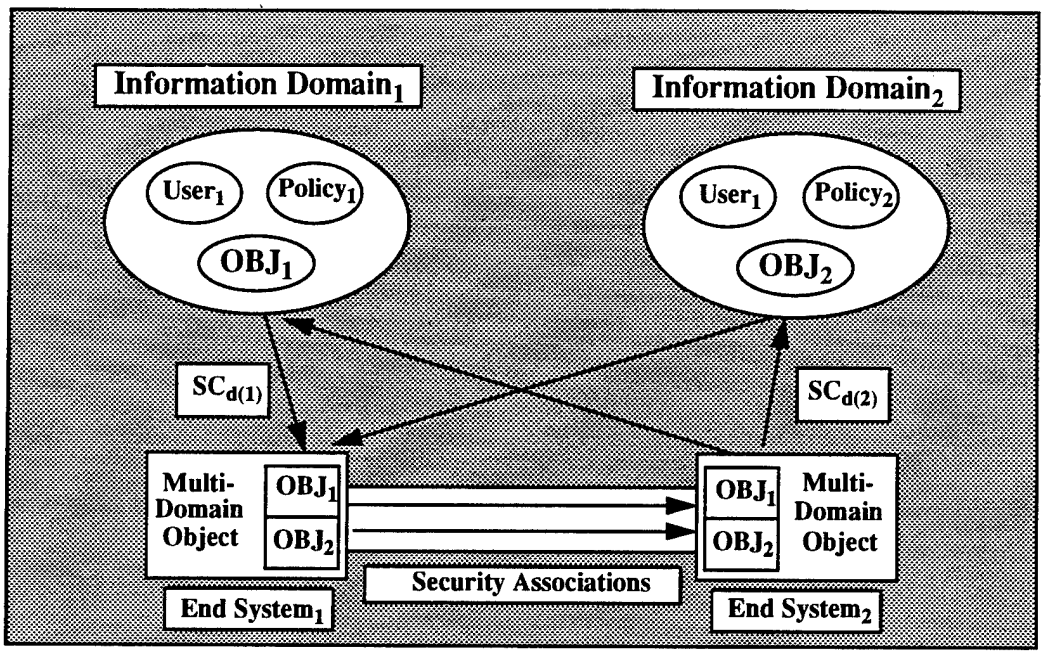


Figure B-4. Multi-Domain Information Object Transfers

B.7 UNIFORM ACCREDITATION

The fundamental security concepts described in the goal security architecture (i.e., information domains, strict isolation, absolute protection) provide the basis for achieving a uniform process for accrediting information domains. In essence, each information domain must be accredited to process information on each supporting end system (and relay system) as part of a local subscriber environment (LSE).

Definition B.10. A *uniform accreditation* process must ensure that each information domain security policy is enforced on each end system where processing will occur.

The objective is to achieve consistency of protection on all end systems supporting a given information domain by providing at least the minimum strength of protection on each end system necessary to enforce the domain security policy.

From an implementation perspective, each information domain must have an accreditation authority responsible for obtaining the necessary evaluations of all LSEs supporting the accreditor's domain. The evaluation of each LSE assesses the capability of each specific end system (or other LSE component) to support strict isolation of the information domain. The complete set of LSE evaluations assesses the capability to achieve absolute protection for the information domain. The results of the overall evaluation provide the information domain accreditor with a documented assessment of the level of residual risk assumed by placing the domain into operation.

LIST OF REFERENCES

[Cargill89] Cargill, C.F., *Information Technology Standardization: Theory, Process, and Organizations*, Digital Press, 1989.

[DGSA93] Department of Defense, *DoD Goal Security Architecture (DGSA)*, Center for Information Systems Security (CISS), Defense Information System Security Program (DISSP), Version 1.0, August 1993.

[DIA91a] Defense Intelligence Agency, *Compartmented Mode Workstation Labeling: Encodings Format*, DDS-2600-6216-91, November 1991.

[DIA91b] Defense Intelligence Agency, *Department of Defense Intelligence Information System (DoDIIS) Network Security for Information eXchange (DNSIX)*, Version 2.1, DDS-2600-5984-91 (Interface Specifications) and DDS-2600-5985-91 (Detailed Design Specification), October 1991.

[DISA93] Defense Information Systems Agency, *Assessment of the Standardization of DoD Information Processing Security Labeling (IPSL) for Automated Information Systems (AIS)*, Final Draft, October 1993.

[DISSP93] Department of Defense, *DoD Information Systems Security Policy, DISSP-SP.1*, February 1993.

[DoD82] Department of Defense, *DoD Information Security Program*, DoD Directive 5200.1, revised June 1982.

[DoD95] Department of Defense, *Common Security Label (CSL)*, MIL-STD-2045-48501, January 1995.

[EO12356] President of the United States of America, *Executive Order 12356 — National Security Information*, Federal Register, Vol. 47, No. 66, April 1982.

[Gasser88] Gasser, M., *Building a Secure Computer System*, Van Nostrand Reinhold Co., N.Y., 1988.

[IETF93] Internet Engineering Task Force CIPSO Working Group, *Common Internet Protocol Security Option*, Version 2.3, March 1993.

[ISO89] International Organization for Standardization (ISO), Information Processing Systems - *Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture*, ISO 7498-2, 1989.

[JSC94] Joint Security Commission, *Redefining Security*, Report to the Secretary of Defense and the Director of Central Intelligence, February 1994.

[Kent91] Kent, S., *Security Options for the Internet Protocol*, RFC1108, 1991.

[Mayfield91] Mayfield, W.T., Boone, J.M., McDonald, C.W., and Welke, S.R., *Proceedings of the Invitational Workshop on Vendor Perspectives on Federal Information Security (INFOSEC) Product Evaluation Criteria*, IDA Document D-1086, November 1991.

[NIST93] National Institute of Standards and Technology, *General Procedures for Registering Computer Security Objects*, NISTIR 5308, December 1993.

[NIST94] National Institute of Standards and Technology, *Standard Security Label (SSL) for Information Transfer*, FIPS 188, September 1994.

[NSA94] National Security Agency, *Information Systems Security Products and Services Catalog*, GPO 908-027-00000-1, April 1994.

[OMG93] Object Management Group, *The Common Object Request Broker: Architecture and Specification*, OMG Document Number 93-12-43, Revision 1.2, December 1993.

[OSD94] Perry, W.J., "Specifications & Standards — A New Way of Doing Business," Memorandum, Office of the Secretary of Defense, Washington, D.C., June 1994.

[OASD95] Paige, E., "Technical Architecture Framework for Information Management (TAFIM), Version 2.0," Memorandum, Office of the Assistant Secretary of Defense for Command, Control, Communication, and Intelligence (ASDC3I), Washington, D.C., March 1995.

[Pfleeger89] Pfleeger, C.P., *Security in Computing*, Prentice Hall, N.J., 1989.

[Rushby84] Rushby, J., "A Trusted Computing Base for Embedded Systems," *Proceedings of the 7th DOD/NBS Computer Security Symposium*, pp. 294-311, September 1984.

[SecureWare93] SecureWare, Inc., *MaxSix for Open Desktop Administrator's Guide*, 010-00069-00, Revision F, July 1993.

[TAFIM93] Department of Defense, *DoD Technical Architecture Framework for Information Management (TAFIM)*, Defense Information Systems Agency (DISA), Version 2.0, November 1993.

[TCSEC85] National Computer Security Center, *Trusted Computer Systems Evaluation Criteria*, DoD 5200.28-STD, December 1985.

[Webster88] Merriam-Webster, Inc., *Webster's Ninth New Collegiate Dictionary*, Merriam-Webster, Inc., M.A., 1988.

BIBLIOGRAPHY

The documents listed in this bibliography provide background information in the general field of computer security and/or the specific topic of security labeling. This list is intended to supplement the information available to the reader on these topics, but is not intended to be comprehensive.

Abrams, M., King, O., Lazear, M., and Olson, I. 1991. Prototype Implementation of ORGCON Policy, MP-91W00051. McLean, VA: The MITRE Corporation.

Congress of the United States of America. 1988. *Computer Security Act of 1987*, Public Law 100-235 [H.R. 145], pp. 101 Stat. 1725 - 1730.

Denning, D.E. 1982. *Cryptology and Data Security*. Reading, MA: Addison-Wesley Publishing Company.

Department of Defense. 1979. *Security Requirements for Automatic Data Processing (ADP) Systems*, DoD Directive 5200.28.

Department of Defense. 1992. *Information Technology Standards Management Plan*, DISA JIEO Plan 3200.

Department of Defense. 1994. *DoD Goal Security Architecture (DGSA) Transition Plan*, Center for Information Systems Security (CISS), Defense Information System Security Program (DISSP), Preliminary Draft.

Epstein, J. 1993. A High Assurance Window System Prototype. In *Journal of Computer Security* 2. 159-190. IOS Press.

Housley, R. 1990. Security Labels in Open Systems Interconnection. In *Proceedings of the 13th National Computer Security Conference, 1-4 October 1990, Washington, D.C.* Gaithersburg, MD: National Institute of Standards and Technology.

International Organization for Standardization. 1992. Information Technology - *Open Systems Interconnection - Security Frameworks in Open Systems - Part 3: Access Control*, ISO CD 10181-3.

Longley, D. and Shain, M. 1987. *Data & Computer Security: Dictionary of Standards, Concepts and Terms*. New York: Stockton Press.

Maggelet, T.F., Morton, R.P., Walker, R.P., Nash, S.H., and Greer, P.B. 1994. *Technical Standards for Command and Control Information Systems (CCISs) and Information Technology,* IDA Paper P-2959, ATCCIS Working Paper 25, Edition 4.

National Computer Security Center. 1987. *Trusted Network Interpretation,* NCSC-TG-005 Version-1. Washington, D.C.: U.S. Government Printing Office.

National Computer Security Center. 1988. *Glossary of Computer Security Terms.* NCSC-TG-004 Version-1. Washington, D.C.: U.S. Government Printing Office.

Russell, D. and Gangemi, G.T., Sr. 1991. *Computer Security Basics.* Sebastopol, CA: O'Reilly & Associates, Inc.

Saltzer, J.H. and Schroeder, M.D. 1975. The Protection of Information in Computer Systems. In *Proceedings of the IEEE* 63, (September): 1278-1308.

Williams, J.C. and Day, M.L. 1988. Sensitivity Labels and Security Profiles. In *Proceedings of the 11th National Computer Security Conference, 17-20 October 1988, Baltimore, Maryland.* Gaithersburg, MD: National Bureau of Standards [now the National Institute of Standards and Technology].

Woodward, J.P.L. 1987. Exploiting the Dual Nature of Sensitivity Labels. In *Proceedings of the 1987 Symposium on Security and Privacy, 27-29 April 1987, Oakland, California.*Washington, D.C.: Computer Society Press of the IEEE.

GLOSSARY

absolute protection: requires each end system supporting an information domain to possess the minimum requisite strength of protection necessary to ensure the domain security policy is adequately enforced.

access control list (ACL): an ordered list of subject/permission pairs that is associated with each object and is used to enforce the DAC portion of a security policy.

attribute: a property of an entity that defines a characteristic of the entity.

client: a program that is invoked by a subject to perform some task.

communications network: device outside the direct control of any local subscriber environment (LSE) that is used to connect LSEs.

controlled entity: an entity (i.e., end user, information object, or domain security policy) that has associated security attributes.

controlled entity stack: the instantiation of an information domain on an end system that brings together the controlled entities from (logical) information domains and the repository of (physical) information system resources. A controlled entity stack is dynamic in nature, and represents a group of controlled entities associated with one another because of their membership in a particular information domain and their residing on a specific end system.

discretionary access control (DAC): a means of restricting access to an object based on the identity of subjects and/or groups to which the subjects belong [TCSEC85]; DAC permissions are assigned at the discretion of the owner of the object.

distributed security context: the joining of two end system security contexts that have been established for an end user in support of the same information domain.

domain: see *domain of interpretation* or *information domain.*

domain mapping: the correlation of a local representation of a domain of interpretation to a non-local representation of a domain of interpretation.

domain of interpretation (DOI): (also known as "named tag set" or "domain of translation") a unique value included in the security header of a network packet that identifies the semantics of all other security attributes in the packet.

domain security policy: a security policy for an information domain.

end system (ES): an information processing system to include processor and input/output devices (e.g., workstation, personal computer, server, minicomputer, mainframe, disk drive, printer, telephone) directly accessible by end users [DGSA93].

end system information domain: an information domain containing end system entities, where the end users (typically systems administrators) and the information objects (information system resources) are governed by an end system security policy.

end system security policy: a security policy that specifies how sharing of information system resources (e.g., security functions, services, mechanisms, devices, memory, registers) is accomplished on an end system in support of a set of information domains.

end user: a consumer or producer of information objects and information system resources [DGSA93].

enterprise: the top level of integration in the Defense Information Infrastructure that includes policy, doctrine, standards, models, architectures, methods and tools, and shared computing and telecommunications services.

enterprise security policy: a security policy for an enterprise.

entity: a primitive element (i.e., information object, end user, or information system resource) from which all other structures are built.

information: a signal or character representing data [Webster88].

information domain: a logical information management concept consisting of a set of uniquely identified information objects, a set of end users, and a domain security policy [DGSA93].

information domain import-export rules: the part of an information domain security policy that establishes the rules, conditions, and procedures for the transfer of information objects among a set of information domains.

information domain security policy: a security policy that provides a statement of the criteria for membership of end users in an information domain and the required protection, including conditions of use, for the information objects in the domain.

information label: a piece of information, representing a hierarchical sensitivity level and a set of non-hierarchical sensitivity categories, that is associated with each subject and object and is used to indicate the actual sensitivity of the information contained in the subject of object.

information object: a structural element of information organized by size or granularity (e.g., bits, bytes, words, pages, segments, fields, records, files) and by type (i.e., date, programs); note that the DGSA concept of an "information object" is different from the Object Management Group (OMG) concept of an "object."

information system: any component or group of components that generates, collects, processes, stores, transfers, disseminates or disposes of information.

information system resource: a provider of information system services and/or facilities for processing, transmission, and storage (e.g., input/output devices, memory, registers, system functions).

infrastructure: standardization infrastructure consists of organizations that control the development of standards and the processes that determine the effectiveness of those standards (e.g., registration and the standardization process itself).

integrity label: a piece of information, representing a hierarchical integrity level and a set of non-hierarchical integrity categories, that is associated with each subject and object and is used to enforce the MAC portion of a security policy addressing integrity.

interprocess communication (IPC): communication between processes.

label: see *security label, sensitivity label, integrity label,* and *information label.*

local communications system (LCS): a set of communication devices (e.g., ring, bus, twisted pair, coaxial cable, fiber-optic cable) under the direct (physical) control of a local subscriber environment.

local subscriber environment (LSE): a set of end systems, a set of relay systems, and a set of local communications systems.

mandatory access control (MAC): a means of restricting access to an object based on a comparison of the label associated with the object to the label associated with the subject requesting access [TCSEC85]; MAC permissions are assigned by the system and must be satisfied as a prerequisite to checking DAC permissions.

mission: a specific task with which a person or a group is charged [Webster88].

multi-domain object: a special composite virtual information object created from constituent objects from different information domains.

native representation: machine-readable representation of a security label on an end system.

network representation: machine-readable representation of a security label in the network.

object: see *information object* for the DGSA definition. The OMG defines an object as a combination of state and a set of methods that explicitly embodies and abstraction characterized by the behavior of relevant requests. An object is an instance of an implementation and an interface. An object models a real-world entity, and it is implemented as a computational entity that encapsulates state and operations (internally implemented as data and methods) and responds to requestor services [OMG93].

organization: an administrative and functional structure [Webster88].

pipe: an IPC mechanism that allows the transfer of data between processes in a first-in, first-out manner.

reference mediation: an access control concept in which an abstract machine (i.e., reference monitor) mediates all accesses to objects by subjects [TCSEC85].

relay system (RS): an information processing system (e.g., multiplexor, router, switch, cellular node, message transfer agent) not directly accessible by end users.

security association: the set of all information system resources (i.e., security and communications protocols, security functions, security services, and mechanisms) employed to securely link two distinct security contexts on different end systems supporting the same information domain.

security attribute: an element of information that is associated with an entity for the purpose of applying a security policy (e.g., a label or an ACL).

security context: a collection of all data, programs, and system resources (e.g., hardware, system software, end user application software, and information) necessary to support a particular end user or system function operation in a particular information domain in accordance with a specific domain security policy [DGSA93].

security doctrine: addresses the specific conditions of use for a particular component [DGSA93].

security label: a tag or marking associated with a container of information (e.g., paper document or electronic file) that is used in a manual or automated fashion to mediate sharing and separation required by a given security policy.

security management: concerned with the management of security policies, security services, security mechanisms, mechanism support, and transfer system security.

security management application process (SMAP): a program in execution that operates on a SMIB.

security management information base (SMIB): contains information domain policy rules, end user registration information, end user authentication criteria (e.g., strength of mechanism required), end user security attributes, and security service and security mechanism requirements for inter-domain information transfers.

security management information domain: an information domain containing security management entities, where the end users are typically systems administrators, the information objects are SMIBs, and the domain security policy states membership and protection requirements for the security management information domain.

security policy: a statement of intent and a course of action with regard to protection of information or information system resources that provides direction, defines responsibilities, and establishes accountability. Such policies can be developed at different levels of abstraction ranging from high-level national policy to specific enterprise policies supporting missions and organizations.

security policy decision function: responsible for making all security policy decisions within an information system.

security policy enforcement function: enforces result returned by a security policy decision function.

security service: a service (e.g., authentication, access control, data integrity, data confidentiality, non-repudiation, or availability), provided by a layer of communicating open systems, which ensures adequate security of the systems or of data transfers.

sensitivity label: a piece of information, representing a hierarchical sensitivity level and a set of non-hierarchical sensitivity categories, that is associated with each subject and object and is used to enforce the MAC portion of a security policy addressing confidentiality.

socket: an IPC mechanism that is an endpoint for communication; typically, sockets are connected to establish a two-way communications link between processes.

standard: guideline documentation that reflects agreements on products, practices, or operations by a particular group. Formal standards are developed by nationally or internationally recognized industrial, professional, trade association, or governmental bodies. De facto standards are adopted based on their prevalence in the marketplace.

strict isolation: the logical and physical separation of a set of information domains and their associated controlled entities from other domains on a specific end system.

subject: an active entity (e.g., a process) that can access and/or manipulate information.

system security policy: security policy translated into entities and attributes that exist on a system.

thread: a single instruction stream executing within a subject; a multi-threaded process can have more than one instruction stream executing in parallel (i.e., parallel processing).

transfer system: a logical grouping of communications protocols integrated into end systems, relay systems, local communications systems, and communications networks.

trusted process: a system utility that may bypass the security mechanisms of the system but is trusted not to violate the system security policy.

uniform accreditation: ensures that each information domain security policy is enforced on each end system where processing will occur in order to achieve consistency of protection on all end systems supporting a given information domain.

LIST OF ACRONYMS

a	attribute
A	set of attributes
ACL	access control list
ANSI	American National Standards Institute
ASSR	Agreed Set of Security Rules
AT&T	American Telephone and Telegraph
BLS	B Level Security
ce	controlled entity
CE	set of controlled entities
CIPSO	Common Internet Protocol Security Option
CMW	compartmented mode workstation
CN	communications network
CPU	central processing unit
CSL	Common Security Label
d	information domain
D	set of information domains
DAC	discretionary access control
DBMS	database management system
DCPS	Data Communication Protocol Standards
DoD	Department of Defense
DGSA	DoD Goal Security Architecture
DIA	Defense Intelligence Agency
DII	Defense Information Infrastructure
DISA	Defense Information Systems Agency
DISSP	Defense Information System Security Program
DNSIX	DoDIIS Network Security for Information eXchange
DoDIIS	DoD Intelligence Information System
DOI	domain of interpretation
dsc	distributed security context

DTMP	DCPS Technical Management Panel
e	entity
E	set of entities
EDS	Entity-Domain-System
EPL	Evaluated Products List
es	end system
ES	set of end systems
ES	end system
ESA	Enterprise Systems Architecture
FIPS	Federal Information Processing Standard
GOSIP	Government Open Systems Interconnection Profile
GTNP	Gemini Trusted Network Processor
IBM	International Business Machines
ID	identifier
ID	information domain
IEEE	Institute for Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPC	interprocess communication
ISO	International Organization for Standardization
ISWG	Information Systems Security Standards Working Group
LAN	local area network
LCS	local communications system
LSE	local subscriber environment
MAC	mandatory access control
MIL-STD	military standard
MLS	Multi-Level Security
MVS	Multiple Virtual Storage
NII	National Information Infrastructure
NIST	National Institute of Standards and Technology
NSA	National Security Agency
o	information object
O	set of information objects
OASD	Office of the Assistant Secretary of Defense
OBJ	information object

OMG	Object Management Group
OS	operating system
OSD	Office of the Secretary of Defense
OSF	Open Software Foundation
p	security policy
PC	personal computer
POSIX	Portable Operating System Interface for Computing Environments
r	information system resource
R	set of information system resources
RIPSO	Revised IP Security Option
RS	relay system
SAMP	Security Attribute Management Protocol
SAMP	Security Attribute Modulation Protocol
sc	security context
SC	security context
SCOMP	Secure Communications Processor
SGI	Silicon Graphics Computer Systems, Inc.
S-HTTP	Secure HyperText Transfer Protocol
SLID	sensitivity label ID
SMAP	security management application process
SMIB	security management information base
SMID	security management information domain
SNSS	Secure Network Server System
SSL	Secure Socket Layer
SSL	Standard Security Label
TAFIM	Technical Architecture Framework for Information Management
TCB	trusted computing base
TCSEC	Trusted Computer System Evaluation Criteria
TIS	Trusted Information Systems, Inc.
TSIG	Trusted Systems Interoperability Group
TSIX	Trusted System Interoperability for UNIX
u	end user
U	set of end users
VAX	Virtual Address Extension
VSLANE	Verdix Secure Local Area Network Exportable

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	July 1995	Final

4. TITLE AND SUBTITLE
Recommendations and a Plan of Action for Standardized Security Labeling

5. FUNDING NUMBERS
DASW01-94-C-0054

Task Order T-S5-1210

6. AUTHOR(S)
Ron S. Ross, Stephen R. Welke, John M. Boone, Edward A. Feustel, W.T. Mayfield

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Defense Analyses (IDA)
1801 N. Beauregard St.
Alexandria, VA 22311-1772

8. PERFORMING ORGANIZATION REPORT NUMBER
IDA Paper P-3044

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
DISA/JIEO/CFS
Attn.: JEBCE
10701 Parkridge Blvd.
Reston, VA 22091-4398

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12b. DISTRIBUTION CODE
2A

13. ABSTRACT (Maximum 200 words)
The Department of Defense (DoD) recognizes that today's ever-increasing use of information technology to conduct routine business makes protecting automated information essential. Security labels are one type of computer security mechanism used to facilitate controlled access to information in a shared resource environment. This report recommends how computer security label standards should be pursued in light of existing labeling technology and the new security architecture being developed for DoD. The authors examined existing label implementations, leveraged and synthesized related work, and studied existing and emerging label standardization efforts to gain a better understanding of the successes and failures of labeling technologies and standards. The report presents a brief description of the DoD Goal Security Architecture (DGSA) fundamental security concepts and principles, and makes recommendations for pursuing computer security label standards in light of existing label technology and the DGSA.

14. SUBJECT TERMS
Security Labels, Computer Security, DISSP, DoD, Goal Security Architecture (DGSA), TAFIM, Information Systems, Networks, Models.

15. NUMBER OF PAGES
108

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR