COVER FEATURE

Digital Government Security Infrastructure Design Challenges

Designing security systems for a digital government's multidomain environment requires a careful balancing act between providing convenient access and carefully monitoring permissions.

James Joshi
Arif Ghafoor
Walid G. Aref
Eugene H. Spafford
Purdue University

Information Age technologies provide enormous opportunities for a government to transform its functions into the digital arena. Doing so taps the wellspring of information technology benefits that have driven down off-the-shelf component costs and fueled an unprecedented improvement rate in the cost-performance ratio. We can view a digital government (DG) as an amalgam of heterogeneous information systems in which government agencies and public and private sectors exchange a high volume of information.[1]

Several US government agencies have aggressively adopted information technologies and spearheaded the search for improved services and decision-making processes. These agencies seek to modernize the government's highly fragmented service-centric information infrastructure. Accumulating evidence indicates that electronically improving information flow and the decision-making process provides increased efficiency, streamlined functionalities, and more effective use of government resources.

Discarding the US government's traditional *command-and-control* public-management technique, in which agencies operate in a loosely coupled environment, newly evolving approaches depend on close collaboration, negotiation, and decision-making processes. These methods require more efficient, flexible, interoperable, and secure information systems. Creating such systems requires a holistic development approach to building a secure information infrastructure. This infrastructure supports both the intricate interdependence of government programs at different levels and between government and the private and public sectors that have become essential partners in supporting government's public services.

Although Information Age technologies provide intriguing opportunities for developing DG concepts, they also create significant infrastructure challenges. Key challenges include[1]

- ensuring secure interoperability among systems from several agencies,
- developing methods and measures of citizen participation in the democratic processes,
- fostering public-private partnerships and other networked organizational forms,
- archiving and managing electronic records,
- developing better methods for IT resource management, and
- ensuring availability and equity of access.

Information security pervades all such needs. In a DG environment, secure interoperation ensures confidentiality when individuals, private organizations, and government agencies access information. Electronic transactions and delivery systems must be secure to ensure protection against fraud and other vulnerabilities. The government's archived information should be protected from tampering yet remain accessible under proper authorizations. Among all government functions, maintaining collective security remains the most crucial element, requiring that security concerns be addressed at each level of the government's information infrastructure.

In general, the concepts and ideas we describe here—although applied to DG uses—are applicable

0018-9162/01/$10.00 © 2001 IEEE

to any distributed information systems that support workflow-based applications[2] across several domains.

INFORMATION SECURITY

As industry analysts have observed,[3] information system security goals include confidentiality or secrecy, integrity, availability, accountability, and information assurance. To ensure information's confidentiality, its disclosure must be restricted to authorized accesses only. Essentially, information integrity guarantees that information is protected from intentional or accidental modifications. Information availability implies access to information uninterrupted by malicious denials of service or unauthorized deletions. Accountability ensures that an entity's every action is uniquely traceable to that entity. Information assurance implies that a specific implementation provides some degree of confidence about pre-established security goals.

Depending on the environment, the relative emphasis assigned to each of these objectives may vary. For example, for defense applications, confidentiality may be the primary requirement, whereas in the commercial sector, information integrity is paramount. In many cases, a combination of these goals may be warranted. For example, in healthcare and airline applications, both confidentiality and data integrity can co-exist as main goals.

Key security mechanisms

An information security infrastructure's foundation consists of three key mechanisms: authentication, access control, and audit. Authentication establishes the identity of an entity and is a prerequisite for access control. Access control limits the actions or operations that a legitimate entity performs. The audit process collects data about the system's activity and detects possible security breaches.

Once it establishes user authentication, the system should enforce access control using an established technique such as a reference monitor that mediates each access by a user to an object. In large distributed and heterogeneous systems, like a DG, designing and implementing these mechanisms in an integrated manner poses a daunting challenge.

Access control

Researchers in the area of computer security have proposed several access-control models to address the security needs of information and database systems. Traditional access control models fall into two broad categories: discretionary (DAC) and mandatory (MAC). DAC policies let users grant their privileges to other users. MAC models use a classification approach for subjects and objects. User classification leads to several clearance levels for access control, whereas classification of objects can be established according to their sensitivity. To avoid the unauthorized flow of sensitive information, the MAC model—also referred to as the multilevel model[4]—can enforce *no read-up* and *no write-down* rules at a given level.

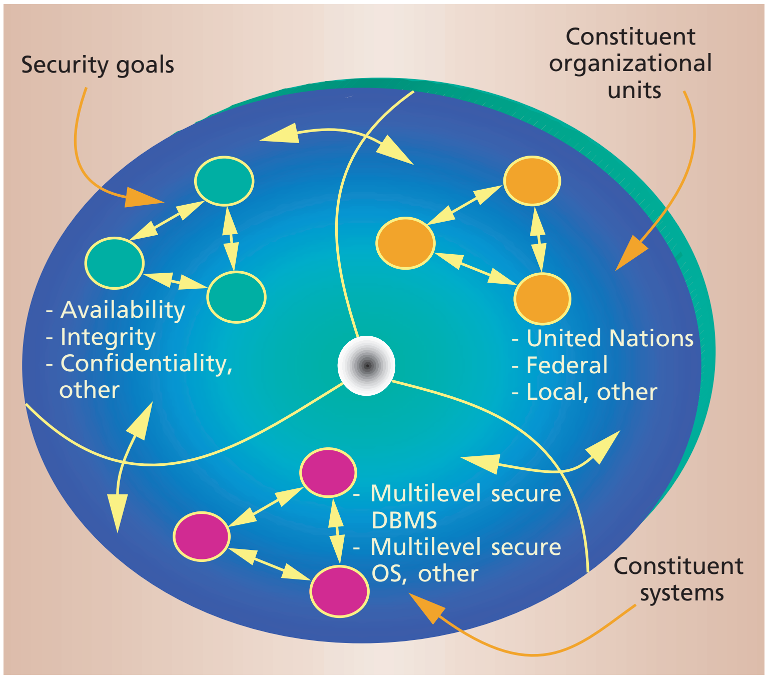


Figure 1. Main features of a digital government's multidomain environment. The components of a multidomain environment are composed of constituent organizational units such as the United Nations and federal and state agencies with different security goals (availability, integrity, and confidentiality). In addition, it consists of heterogeneous software system components (for example, multilevel secure OS, DBMS).

For supporting large-scale distributed applications, DAC and MAC models have several shortcomings. For example, MAC models lack flexibility to support the arbitrary security requirements an application may have. On the other hand, the high degree of flexibility in DAC models can let unauthorized users find ways to access protected objects.

Researchers have proposed several new approaches and models to address these issues. These models include role-based access control (RBAC) models, task-based access control (TBAC) models, and ticket-based approaches.[4-6]

MULTIDOMAIN SECURITY CHALLENGES

Inherently multidisciplinary and dynamic, a government's organizational and operational base requires a secure information infrastructure.

Multidomain environment

One key aspect of a multidisciplinary infrastructure is the existence of diverse information security policies employed by individual government agencies. These varied policies create a highly heterogeneous multidomain environment. Such environments should support interoperability of several security domains and allow strong interdomain interaction. The main features of a multidomain environment, as Figure 1 shows, consist of the following:

  - The environment can be composed of diverse interacting and collaborating constituent agencies that employ individual policies, such as private organizations or local, state, federal, and international government agencies.
  - The environment can have more than one security goal, with the multiple goals consisting of variations on the same goal or a series of drastically different ones. The collaborating agencies may design their constituent domains to achieve one or more security goals.
- The infrastructure supporting such an environment can have heterogeneous system components, services, and applications, which can include database federations, multilevel secure database management systems, and multilevel secure operated systems.[7]

Several security policies in a multidomain environment can coexist and evolve with the changing operational needs and services the government provides. The overall infrastructure must allow seamless and secure interoperation among diverse and heterogeneous security mechanisms. The infrastructure should be scalable, open, and extensible. Meeting all these requirements presents several technical challenges.

Semantic heterogeneity and metapolicy

The diversity of organizational and user-specific security policies in a DG environment requires powerful formalisms for efficiently mapping security attributes across interacting domains. Determining a specification's correctness and reasoning about the safety and liveliness properties of a multidomain environment's security mechanism require formal models and metapolicies. These models should be generic and flexible enough to express a wide range of security policies and must provide a semantic basis for policy composition and modifications.

Policy neutrality avoids restricting security implementations to DAC or MAC. Rather, a policy-neutral model supports arbitrary user-defined policies. The environment's formal models and framework should also provide a theoretical basis for assessing the level of security assurance.

Metapolicies must also allow autonomy and transparency for the policies adopted by an individual domain, which provides for the policies' continuous evolution.

Secure interoperability

Any policy change, addition, or deletion requires reevaluating the system's secure interoperability.

Any policy change, addition, or deletion requires reevaluating the system's secure interoperability. Secure interoperability poses a major challenge when dealing with an environment where subjects from a different domain access objects in a given domain. If more than one rule governs the interaction among multiple domains, such a situation can cause rule conflict and may require a mediation policy to identify appropriate rules.[8]

Many possible multidomain scenarios in a DG environment highlight the need for secure interoperability. For example, suppose an entrepreneur plans to establish a small business—a pharmaceutical factory—in a state in the US. Assume that the state government's Business Development Agency has developed an integrated information system that provides a one-point interface for helping establish such a business. The BDA lets the entrepreneur obtain all necessary information to set up a business, including finding a suitable site, facilitating the process of purchasing land, and acquiring the permits and clearances essential to establishing the business.

The system also supports an integrated electronic application process that automatically generates all necessary transactions to other relevant agencies required to determine whether the BDA can permit the applicant to establish a business. These agencies can include the Environmental Protection Agency, which certifies the use of chemicals and disposal plans for hazardous by-products; the local police departments; the FBI, which provides a background check to certify that the applicant has no criminal record; the real estate agencies that maintain the GIS information about the sites suitable for building a pharmaceutical factory; and the BDAs for states where the applicant has previously operated a business, who must certify that the applicant has maintained acceptable business conduct.

In such a scenario, the local state's BDA must interoperate with all these agencies. The information systems for each of these agencies can employ their own security policies, which can result in a possible access-rules conflict that requires a mediation policy. For example, in the case of poor metapolicy specifications, the domains of the local BDA and the real estate agency may overlap, with each agency explicitly defining access policies for common objects restricted only to their own domain users. In addition, such interoperation may create other risks.

In essence, secure interoperation should enforce the following two principles:[9]

- The autonomy principle, which states that if access is permitted within an individual system, it must also be permitted under secure interoperation.
- The security principle, which states that if an access is not permitted within an individual system, it must not be permitted under secure interoperation.

It is impossible to guarantee secure interoperation among multiple domains because finding a secure solution with some optimality presents an NP-complete

Table 1. Digital government security challenges and potential approaches to solving them.

Challenges	Solution approaches
Semantic heterogeneity/ metapolicy	Generic language such as Z specification language, algebraic, and security automata
	Policy-neutral models such as RBAC
	Typed extensions of access control matrix models such as TAM and DTAC
	Programmable security
	Export interfaces
Secure interoperation	*Conflict types*
	Domain conflict
	Rule conflict
	Conflict-resolution approaches
	Manual, need based
	Priority based, voting
	Composition operators such as Union, Intersection, Product
	Hierarchy of security properties
	Creating virtual roles in RBAC
Flexibility/extensibility	Separation of policy specification and enforcement components; enforce and decide
	Policy library
	Policy habitat and metapolicy
	Layered architecture
Risk control/assurance	Safety analysis such as static and dynamic checking in DTAC
	Use of least-privilege feature in RBAC system
	Inline code
	Retain reference monitor properties such as tamperproof, complete mediation, and verifiability
Administrative/management	Administrative models such as role based
	Auditing
	Risk, vulnerability analysis
	Security assessment and certification
	Layered architecture

problem.[9] Optimization can include maximizing the amount of shared data among all domains, maximizing the number of legal accesses, or—in an extreme case—minimizing the number of conflicting domains.

Assurance and risk propagation

In a multidomain environment, users must maintain a certain degree of assurance about the entire system's security. While some risks may be acceptable in a local system, such risks can, in a larger network, propagate and increase the level of vulnerability for all component systems.

For example, continuing our BDA scenario, if the local BDA has a security hole that lets the applicant obtain sufficient privileges in the BDA system, he can use it as a back door to access other systems and attempt to penetrate them. For example, assume that the local BDA and real estate agency use the following metapolicy rules:

- BDA employee B assumes the role of real estate agency employee E when B needs to access information in the real estate agency's system, and
- real estate agency employee F assumes the role of BDA employee A when F needs to access information in the BDA system.

In addition, assume that A is senior to B in that A has all B's security privileges, but B does not have all of A's privileges. Similarly, assume that E's role is higher than F's role. In this case, employee F can enter the BDA system and assume A's role and, since A is senior to B, can then assume B's role and enter the real estate agency's system using E's role. In this case, F can acquire all the privileges of his senior, E. Similarly, B can enter the real estate agency's system with E's role and access the BDA system assuming A's role, thus acquiring all A's privileges, even though A is senior to him.

A related issue, the cascading problem, also arises in multidomain environments. Consider two multilevel systems, X and Y. Suppose system X is designed for managing information classified as either secret or top secret and that all users of X are cleared for secret information at least. System Y can handle information classified as confidential or secret, and its users are cleared for confidential information at least.

Now, suppose their owners integrate the two systems, and the resulting three levels of clearance include confidential, secret, and top secret. In the merged system, the secret information can pass between the two systems. If a penetrator overcomes the protection mechanisms in both the individual systems, he can then downgrade the top-secret information of system X to the level of secret and pass such information to system Y. In system Y, the same penetrator can then downgrade that information to confidential. Thus, users having the lowest clearance in either system can access the top-secret information.

Management challenges

Thanks to the large number of administrative domains, subjects, and objects, security management in a DG infrastructure presents a challenging task. One characteristic of a DG is that it forms an essentially open system where the entities that represent users, objects, policies, security domains, and other components are transient. This inherent dynamism makes the task of overall management and, in particular, security configuration management, highly complex.

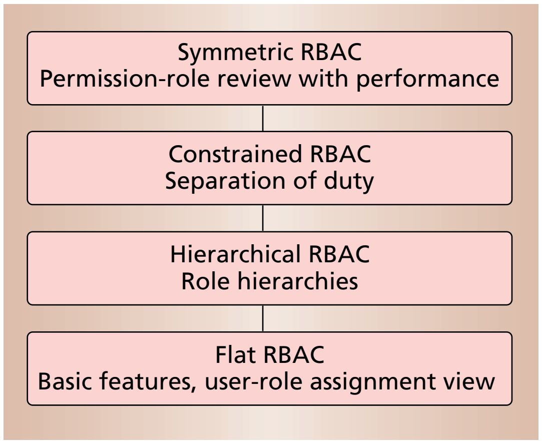
INFORMATION SECURITY APPROACHES

To meet the challenges we have described, security experts have developed several approaches to information security management. Here we profile the strengths and limitations of the most prominent methods, which Table 1 summarizes.

Policy-metapolicy specification

In a multidomain environment, establishing seman-

Figure 2. NIST's role-based access control (RBAC) model, a proposed standard reference model, uses a four-level hierarchy in which each level includes all the capabilities of the levels below it.



tically correct relationships among security policies is essential to ensuring secure cooperation. Metapolicies can specify such relationships as cooperation rules and guidelines for conflict resolution and interaction. Existing metapolicies are either ad hoc or based on formal approaches.

Ad hoc approaches. These policies put more emphasis on the system's development and implementation details. In particular, they emphasize conflict resolution among multiple domains. Hilary Hosmer[10] has proposed several conflict-resolution methods, including manual, standard form, and rule-based strategy approaches.

The manual approach, used most commonly, assigns a security officer the responsibility for manually integrating multiple policies and resolving conflicts. In the standard-form approach, the organization adopts some generic or policy-neutral guidelines to ensure secure interoperability. Each domain uses a conversion logic to translate its local rules to a global metapolicy schema.

In a rule-based strategy, the conflict resolution mechanism uses a predefined set of rules that can include either a voting technique or a set of informal guidelines. The conflict resolution mechanism can use various trade-offs while resolving conflicts.[9]

Formal approaches. Winfried Kuhnhauser's[6] formal approach to metapolicies classifies multidomain interactions into three unique classes:

- Class 1 represents the conflict-free interactions that occur when both subject and object belong to a single domain.
- Class 2 characterizes the situation in which no security policy can provide the rule for interaction across multiple domains. Such scenarios occur when the absence of a comprehensive security policy creates a policy hole.
- Class 3 describes those systems in which domains can overlap. In this case, a subject from a nonoverlapping domain can access objects that happen to be in the overlapped region. Such interactions can result in rule conflicts that require a mediation policy.

For mediation, Kuhnhauser's framework uses conflict and cooperation matrices. A conflict matrix provides a ranking mechanism to resolve conflicts between two policies. The cooperation matrix stores the information about a predetermined policy to be used when two domains interact.

Model-based methods

Traditional DAC and MAC models lack capabilities for expressing a domain's arbitrary security requirements. Increasingly, developers use flexible approaches that allow user-defined security policies. The RBAC model is a flexible approach that has generated great interest in the security community. Recently, Ravi Sandhu and colleagues have proposed the National Institute for Science and Technology RBAC[5] as a standard reference model. Depicted in Figure 2, RBAC uses a four-level system in which each higher level includes the functional capabilities from all levels below it. The levels correspond to four RBAC models: flat, hierarchical, constrained, and symmetric.

The flat RBAC model provides the minimal features essential for any RBAC mechanism. These include roles, user-role assignment, and role-privilege assignment.

Hierarchical RBAC includes as a requirement role hierarchies, which define relationships among roles in a domain. A role can be senior to other roles, in which case the senior role inherits all the privileges of the junior roles.

Constrained RBAC requires separation of duties (SOD), which aims mainly to avoid fraud and errors in an organization. For example, in some domains, the same user cannot be assigned two roles—such as accounting clerk and purchasing clerk. Such a model uses the notion of static SOD. Alternatively, in a dynamic SOD case, a user may be assigned two roles but is restricted from activating both roles in a single session.

The symmetric RBAC model adds a permission-role review requirement. As a result, the model allows identification of the permissions assigned to existing roles and vice versa. Most RBAC models use roles to imply a collection of access privileges.

The RBAC model is an attractive candidate for a DG infrastructure because it provides flexible support in a multipolicy environment. Security administrators can use role hierarchy mapping between two RBAC-based domains to define a metapolicy for interoperability. An RBAC's relatively simple security administration allows separation of user-role and role-privilege assignments. When users receive multiple roles, RBAC ensures that they can activate only the required roles for a particular access, thereby minimizing damage from inadvertent errors.

RBAC's policy neutrality, constraints, and role hierarchies make it a powerful model for specifying policies from other models such as DAC and MAC and for specifying rules from any arbitrary user-specific model. A mixture of such policies can coexist in a DG infrastructure. This heterogeneity makes RBAC useful in a multidomain environment. Further, models for administrative roles provide efficient mechanisms for distributing security management functions to a number of administrators.

Other new access control models that have shown potential for supporting a multipolicy environment include multiple-policy schematic protection, typed access matrices, and dynamically typed access control models, which use subject and object types. However, these models have reached only the initial phases of their development.

All the models we have described use the subject-object view for specifying security policies. These models have a limited scope and, in a DG environment, cannot be expanded to include an access policy based on the content of information or the nature of tasks and transactions. Applications and services in a DG environment can require automated transactional functions and workflow-based processing, which result in a highly transaction-intensive infrastructure. Roshan K. Thomas and colleagues[6] have proposed an initial TBAC family of models in which the authorization unit is a task.

Agent-based methods

With the growing maturity of software engineering, software agents have emerged as a popular system-building paradigm. Computer systems security designers can use agents—characterized by adaptation, cooperation, autonomy, and mobility—to provide security features for a DG infrastructure. Agent communication languages such as the Knowledge Query and Manipulation Language can negotiate policies during conflicts to ensure secure interoperation.[11] The servers and clients in a distributed environment can assign security enforcement tasks to agents.

Although the mobility and adaptability characteristics of agents provide essential features for the efficient use of system resources, they can pose several security threats. For example, an agent can engage in malicious behavior, thus disrupting the host's normal operation. Similarly, a host can hinder an agent's activity by denying required access to local information resources.

Information about agent technology that addresses the issue of security in a heterogeneous environment has begun appearing in the literature. For example, an agent-based architecture has been proposed[11] as a solution to the public-key infrastructure open standard, which facilitates interoperable and flexible authentication for various applications. System designers can use similar approaches to address the issue of access control policies.

Architectural methods

Several approaches that address the challenges of multipolicy environments also address architectural issues. Notable among these are the Object Management Group's Common Object Request Broker Architecture (Corba) and the Open Software Foundation's Distributed Computing Environment. Corba offers a security policy specification but lacks formal semantics, thus making security-handling mechanisms more or less ad hoc. DCE addresses the general issue of object interoperability by providing a middleware architecture that implements an ad hoc security mechanism.

Some other proposed architectures include the Distributed Trusted Operating System and the Meta Object Operating System Environment[12] (Moose). DTOS supports separation between the policy specification and policy enforcement components by using a mix of tabular representation and a language-based specification model to provide a high degree of flexibility in security policy selection. Moose's three-layer architecture uses a formal approach to integrate modeling, specification, verification, and implementation.

Models that use the subject-object view for specifying security policies cannot be expanded to include an access policy based on the content of information or the nature of tasks and transactions.

Database federation approach

The database federation approach, which integrates several database management systems, provides some solutions to the multidomain problem. Database researchers have proposed several approaches for developing systems that achieve the autonomy of component databases yet remain transparent at the federation level. These approaches also address a multidomain environment's security management issues. For example, Dirk Jonscher and Klaus R. Dittrich have reported a federated database system that uses several DAC and MAC policies.[13] This system uses a global access layer to map global authorizations into the local-access rights of individual databases.

The Distributed Object Kernel[7] is another example of a secure federated database system that uses a mapping technique to build a global-access policy from local DAC and MAC policies. In the DOK system, the enforcement mechanism for global security involves layered processing by agents designed to check attribute constraints and sanitize query results. Developers can expand approaches for federated-database schema integration to design the metapolicy for access control in a multidomain environment and to provide a viable security-management solution for a DG infrastructure.

Of the many technologies currently in development, RBAC models appear to be the most attractive solution for providing security features in a multidomain digital government infrastructure. RBAC features such as policy neutrality, principle of least privilege, and ease of management make them especially suitable candidates. In addition, RBAC models can address some of the challenges we have described.

For example, such models can express both DAC and MAC policies, as well as user-specific policies. In essence, RBAC models can provide a generic framework for expressing diverse security requirements.

Federated database management system approaches also show promise and will likely be expanded to effectively address multipolicy issues. Agent systems, on the other hand, require further exploration to evaluate their security enforcement features. Much work remains before we can use agents safely in a complex environment such as a digital government, which does not permit the viable use of a centralized reference monitor. From the security management perspective, the architectural separation of policy specification and enforcement mechanisms is growing in importance. Providing the techniques needed to evaluate these models for system assurance and risk analysis remains a major challenge. ✷

Acknowledgment

This study is supported, in part, by a grant from the Center for Education and Research in Information Assurance and Security at Purdue.

References

1. H. Schorr and S.J. Stolfo, "Towards the Digital Government of the 21st Century," *Workshop on Research and Development Opportunities in Federal Information Services* report, June 1997, http://www.isi.edu/nsf/final.html.
2. E. Bertino, E. Ferrari, and V. Atluri, "The Specification and Enforcement of Authorization Constraints in Workflow Management Systems," *ACM Trans. Information and System Security*, vol. 2, no. 1, 1999, pp. 65-104.
3. G. Stoneburner, "Information System Security Engineering Principles (ISSEP)—Initial Draft Outline," NIST, 2000, http://csrc.nist.gov/publications/drafts/epits-draft-0010231.pdf.
4. R.S. Sandhu, "Role Hierarchies and Constraints for Lattice-Based Access Control," *Computer Security ESORICS 96, LNCS 1146*, Springer Verlag, Berlin, 1996, pp. 65-79.
5. R.S. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST Model for Role-Based Access Control: Towards A Unified Standard," *5th ACM Workshop on Role-Based Access Control*, ACM Press, New York, 2000, pp. 47-60.
6. R.K. Thomas and R.S. Sandhu, "Conceptual Foundations for a Model of Task-Based Authorizations," *The Proc. 7th Computer Security Foundation Workshop* (CSFW94), IEEE CS Press, Los Alamitos, Calif., June 1994, pp. 66-79.
7. Z. Tari and G. Fernandez, "Security Enforcement in the DOK Federated Database System," *Database Security X: Status and Prospects*, Chapman and Hall, London, 1997, pp. 23-42.
8. W.E. Kuhnhauser and M.K. Ostrowski, "A Formal Framework to Support Multiple Security Policies," *Proc. 7th Canadian Computer Security Symp.*, Ottawa, Communication Security Establishment Press, 1995, pp. 1-19.
9. L. Gong and X. Qian, "Computational Issues in Secure Interoperation," *IEEE Trans. Software Eng.*, Jan. 1996, pp. 43-52.
10. H.H. Hosmer, "Metapolicies I," *ACM SIGSAC Review*, 1992, pp. 18-43.
11. Q. He, K. Sycara, and Z. Su, "A Solution to Open Standard of PKI," *Proc. 3rd Australian Conf. Information Security and Privacy*, ACISP 98, C. Boyd and E. Dawson, eds., Springer, Berlin, July 1998, pp. 99-110.
12. J. Hale, M. Papa, and S. Shenoi, "Programmable Security for Object-Oriented Systems," *Proc. Database Security XII: Status and Prospects*, Kluwer Academic, Boston, 1998, pp. 109-123.
13. D. Jonscher and K.R. Dittrich, "Argos—A Configurable Access Control System for Interoperable Environments," *Proc. 9th Ann. Working Conf. Database Security*, Chapman and Hall, London, 1995, pp. 43-60.

James Joshi *is a graduate student in the School of Electrical and Computer Engineering at Purdue University. His research interests are computer security and multimedia systems. He received an MS in computer science from Purdue University. He is a student member of the ACM and the IEEE. Contact him at joshij@ecn.pudue.edu.*

Arif Ghafoor *is a professor in the School of Electrical and Computer Engineering at Purdue University. His research interests are multimedia information systems, database security, and distributed computing. He received a PhD in electrical engineering from Columbia University. He is a Fellow of the IEEE. Contact him at ghafoor@ecn.purdue.edu.*

Walid G. Aref *is an associate professor of computer sciences at Purdue University. His research interests are database systems, spatial and multimedia data indexing, video servers, network-attached storage devices, data mining, algorithms, data structures, and geographic information systems. He received a PhD in computer science from the University of Maryland at College Park. He is a member of the ACM and the IEEE. Contact him at aref@cs.purdue.edu.*

Eugene H. Spafford *is a professor of computer sciences and the director of the Center for Education and Research in Information Assurance and Security at Purdue University. His research interests are computer and network security, ethical and societal implications of computing, software validation, and verification and debugging. He received a PhD in information and computer science from the Georgia Institute of Technology. He is a Fellow of the AAAS, the ACM, and the IEEE. Contact him at spaf@cerias.purdue.edu.*