Security Policies in Distributed CSCW and Workflow Systems

Tanvir Ahmed and Anand R. Tripathi
{tahmed,tripathi}@cs.umn.edu
Department of Computer Science
University of Minnesota, Minneapolis, MN 55455

***Abstract*— In this paper, we have surveyed Computer-Supported Cooperative Work (CSCW) and workflow systems based on a time-line and have categorized the systems based on their emphasis on user or process interactions. Unique security requirements for these systems are also discussed. Security models to specify and verify security requirements related to security attributes – namely, availability, integrity, confidentiality, and access leakage – are presented. Research challenges of role based access control models for security policies in distributed CSCW and workflow systems are presented. Lastly, current concerns in security policy enforcement mechanisms in these decentralized systems are discussed.**

***Index Terms*— CSCW, Groupware, Workflow, Web Services, Security models, Security policy specification and verification, Role based access control.**

I. Introduction

Web has become an integrated part of our life, enabling applications for interacting with users and services – from traditional electronic mail to recent web-based tax filing, distance learning, and a wide range of web services. On the other hand, the enabling open network infrastructure has raised security concerns for every day applications as oppose to security concerns in traditional government or commercial systems. It was widely accepted, as expressed by Clark and Wilson [1], that government or military systems are mainly concerned with confidentiality or information flow related security properties; On the other hand, commercial systems tend to emphasize on integrity and availability of data and services. With the unsecure network, the security attributes – namely integrity, confidentiality, and availability – desired for a system are not limited to a specific attribute. Many of these systems interact with users that are strangers or reside in different organizations or countries. This itself requires reexamination of the traditional security models that are used to express and enforce security policies.

Social aspects for usage of these Internet-wide collaboration systems have raised demand for security services that were not prevalent earlier. For example, due to recent privacy concerns and laws in healthcare information systems, confidentiality has become a primary concern for many commercial systems. Utilizing the Internet, these systems share data crossing their organizational and security boundaries. Moreover, with the availability of mobile and ubiquitous computing devices and the development of recent technologies, such as peer-to-peer systems and web services, different types of collaboration systems are emerging. Computations in these systems are increasingly performed across organizational boundaries, often close to data. In addition, the open network has introduced new security threats, such as *denial of service*. Security models are developed based on assumptions on system environments including security threats. The new threats require consideration for security models that support security policies in open environments.

In this paper, unique security requirements that are present in current distributed collaboration systems are presented. We investigate the existing security models and policy specification methodologies to express these requirements.

II. CSCW and Workflow Systems

The examples of computer supported collaboration include online conferencing, product design and development, authoring of documents, workflow in an office environment, healthcare activities in a hospital, and collaboration among different organizations. In Figure 1[1], historical perspective of various systems for computer supported collaboration are presented. Early examples of collaboration systems are data processing systems, decision support systems, and management information systems that are responsible for organization level management aspects [2]. Later, systems for software development, collaborative design, office automation, document management, and other systems that manage projects involving multiple users were introduced. Many of these systems are targeted towards small group activities and are termed *groupware*. Groupware systems are differentiated from other multi-user systems, such as database systems, based on their application level requirements of data sharing [4], [2]. Compared to concurrent access of data through transaction management facilities in database systems, groupware systems coordinate interacting users, where the interactions can occur in real-time. These interactions in groupware resemble long term transactions. During the interactions, the users are usually aware of each-other's

This work was supported by National Science Foundation grants ITR 0082215.

Contact author, tahmed@cs.umn.edu, Room 4-192, 200 Union Street SE, University of Minnesota, Minneapolis, MN 55455, U.S.A.

[1] The time line is based on [2], [3]

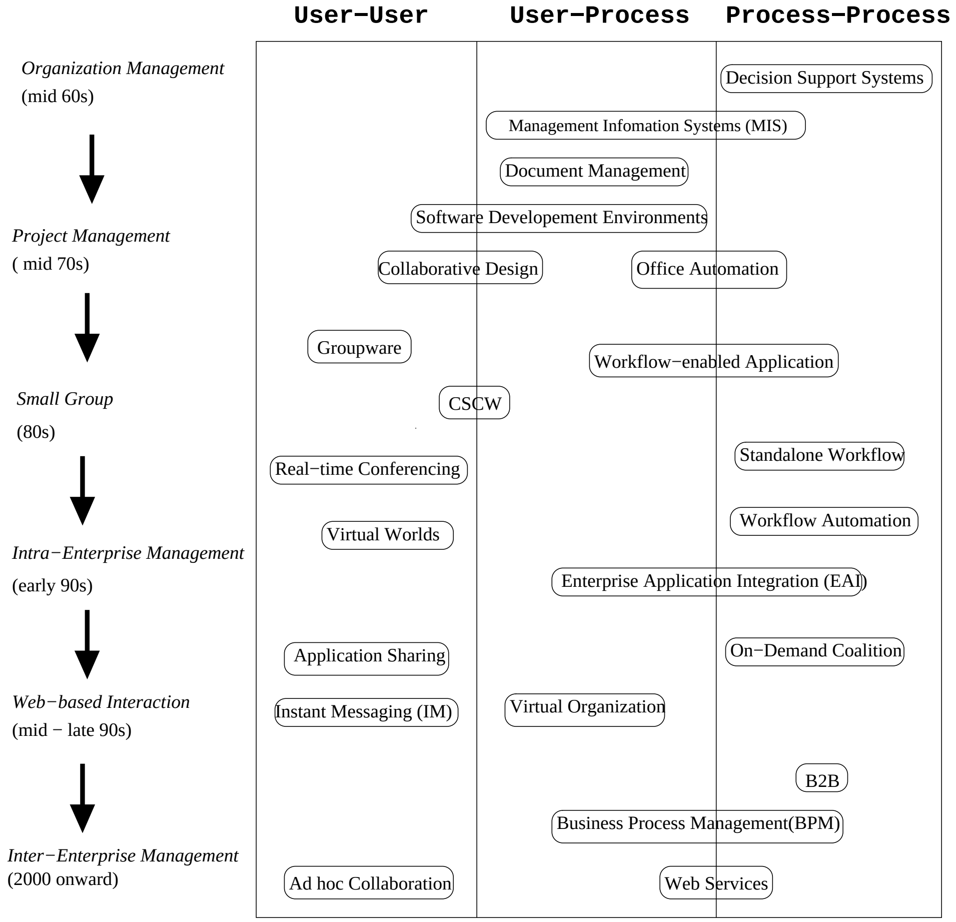

Fig. 1. Time line of CSCW and workflow systems: management aspects and interaction models

presence or other user-level contextual information, hence these systems are often termed as *group-aware*. During 80's, the term Computer-Supported Cooperative Work (CSCW) [4] was introduced for group oriented systems; however, CSCW is a multidisciplinary field that addresses all aspects of group activities including ethnographic, social, technological, and theoretical issues for enabling group activities.

Formally, in groupware systems, multiple users cooperate using shared data and artifacts towards some common objectives [5]. It is envisioned in CSCW systems that the collaborators may not initialy know their common objectives, but rather "discover" similar goals after their interactions have progressed to a certain point [6]. Due to enabling facilities to discover common grounds with other cooperating users, flexibility of sharing information including meta and contextual information is viewed as a primary attribute of groupware systems. CSCW researchers have differentiated cooperative work from coordinated work based on that in the cooperative work users share the work objective, which enables the cooperating users to adapt their actions with each other to achieve the objective [7].

On the other hand, workflow systems were introduced following the tradition of office automation systems that define and automate routine tasks. Workflow automation systems are derived from existing and conventional practices that consist of a set of well-defined activities to execute some enterprise-wide process. These systems are criticized for their emphasis on workflow processes rather than users [3]. For commercial use, many groupware applications become workflow enabled, i.e., groupware applications became aware of workflow stages.

Enterprise Application Integration (EAI) is an enactment of workflow systems for intra-enterprise management. Current workflow systems are synonymous with Business Process Management (BPM), which also incorporates inter-organizational workflow or B2B (Business-to-Business). *On-demand coalition* is another type of cross organizational workflow system. Example coalition systems are disaster reliefs and war-time sharing of data. In the commercial arena, E-Services are the outgrowth of E-Commerce or the web based interaction of buyers and sellers [8]. E-Services model has evolved into Web Services that provide web-delivered services [9]. The motivation behind Web Services is lightweight integration of business processes supporting interoperability across platforms or technologies. A primary attribute of Web Services is XML technologies that define, describe, discover, and invoke these web-delivered services.

In Figure 1, different CSCW and workflow systems are presented across within three cells: groupware that

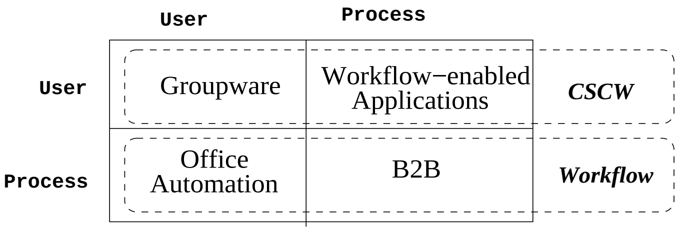

Fig. 2. User process matrix for CSCW and workflow systems

Communication	Information Manipulation	Coordination
Electronic Mail	Shared Whiteboard	Calendars
Real−time Conferencing	Application Sharing	Scheduling
Multicast Video/Audio	Virtual Worlds	Office Automation

Fig. 3. Example CSCW systems based on types of supporting technologies

coordinate users, workflow enabled applications that coordinate users with workflow processes, and workflow systems that coordinate processes of different applications or organizations. Distributed workflow systems, such as BPM and Web Services, support interactions of inter-organizational processes and interactions of users in these processes. On the other hand, systems such as online interactive games represent distributed groupware that support interactions of users situated in different locations. Due to popularity of mobile devices, ad hoc collaboration systems to share network resources are emerging.

In Figure 2, a user-process matrix classifies CSCW and workflow systems based on their emphasis on supporting user or process interactions. The top two cells representing groupware and workflow enabled applications are classified as CSCW systems as their emphasis is on supporting group works. The bottom two cells represent workflow systems as their emphasis is on process automation of organizational activities.

To categorize the wide range of CSCW systems, several CSCW researchers, such as [2], [10], have looked into the types of technologies that collaboration systems support: (1) communication, (2) information manipulation and storage, and (3) coordination (See Figure 3). Most of the collaboration systems do not exclusively fall into a single category, but rather incline toward one of these supporting technologies. Systems such as electronic mail, real-time conferencing, multicast video and audio are usually associated with communication technology. Examples of shared information space technology include shared whiteboard, application sharing, meeting facilitation, virtual worlds, threaded discussions, document management, and information management, e.g., Lotus Notes [11]. Lastly, examples of coordination technology are calendars, scheduling, and office automation systems.

All CSCW systems support some form of coordination technology. CSCW systems are also classified based on the nature of coordination among the participants. The nature of coordination can range from tightly coupled synchronous interactions, as in online conferencing and shared whiteboard environments, to loosely coupled asynchronous interactions, which are largely characterized by workflow environments. In real-time synchronous collaborations, users are connected to the system simultaneously and their interactions occur through interactive sharing and manipulation of graphical and multimedia objects. Generally, the interactions tend to be unstructured, often spontaneous and the coordination management is concerned with concurrency control of shared resources. In contrast, in loosely coupled collaboration, all the users may not be present at the same time, and their coordination actions tend to be coarse-grain. These are usually workflow-enabled systems.

III. Security Requirements in CSCW and Workflow Systems

In CSCW and workflow systems, traditional security requirements of integrity, availability, and confidentiality are naturally present. However, these security requirements in CSCW and workflow systems have application level as well as organizational level attributes. During 80s, Greif and Sarin [4] noted that the protection mechanisms commonly provided by operating systems and database systems tend to be largely inadequate for collaborative applications. Such applications require protection mechanisms to support convenient realization of application-oriented security policies. For example, in a collaborative software engineering application, users assigned to review code may only modify the comments section of the code, whereas a developer can modify any sections of the code. Moreover, the code review can be performed at a code review phase, which can only start after the development phase. These policies widely vary from an organization to another, and also change with new software development methodologies. Hence, policy specification and protection mechanisms are usually build within the application. Based on the management aspects – such as small group, intra-enterprise, inter-enterprise – and the deployment environments, i.e., intranet or Internet, the security requirements of the distributed CSCW and workflow systems vary.

Security requirements in multi-user systems can be traced back to Clark and Wilson in 1987 [1]. They emphasized on two fundamental concepts to ensure integrity of data. These are (1)“well-formed transaction” ensuring that data can only be modified based on predefined constraints, and (2) “separation of duties”, which ensures that conflicting users cannot modify data. Later, various forms of “separation of duties” and “Chinese Wall Security Policy” [12] have been addressed by commercial workflow systems.

Organization level security policies are usually specified based on the role of a specific user group. Roles [13] represent well defined organizational entities, and different users may be assigned to the same role in the lifetime of an organization. The use of role based security policies in CSCW and workflow systems has been found to be quite natural as participants perform a set of well-defined tasks pertaining to their expertise and responsibilities in the organization. In a role based security model, a role represents a set of privileges. A user assigned to a role acquires those privileges.

In the following subsections, various aspects of the security requirements in distributed workflow and CSCW systems are discussed.

A. Separation of Duties

"Separation of duties" policies are utilized in organizations for integrity of business processes to properly address and circumvent conflict-of-interest situations. "Separation of duties" policies can also mandate that more than one user are involved in different stages of critical business processes. For example, in an invoice processing workflow, a "separation of duties" policy can be that the invoice preparer and invoice approver are two distinct users ensuring a user cannot approve his/her own invoice. There are various forms of "separation of duties" policies [14], [15], [16]. The widely used variants are presented below:

Role based static separation-of-duties: This requires that two given roles should never be assigned to the same person. For example, a user cannot be both the accountant and the manager for an organization.

Role based dynamic separation-of-duties: The static "separation of duties" may in some organizations turn out to be overly restrictive. In the above example, once a user is assigned to be either of the accountant or the manager roles, he/she cannot join the other role even though there can be multiple monetary transactions with different business entities. Hence, the dynamic "separation of duty" requires that two given roles cannot be assigned or activated concurrently by the same person. This enables a participant to be a manager for one transaction and an accountant for another transaction.

Identity based separation-of-duties: Enforcement of "separation of duties" policies may require information, such as users' identities, that are acquired from sources outside the policy enforcement mechanisms. An identity based separation-of-duties can require that two particular users should not be assigned to the same role. For example, both the spouses may not be assigned to the same decision making group. This types of policies may require that a specified user should never be assigned to a given role. For example, a rogue user may not be assigned to a system administrator role.

Object based separation-of-duties: This specifies that a user cannot perform multiple operations on the same object by participating in two different roles. A purchase order may not be prepared and approved by the same manager. An "object based separation of duties" policy on the purchase order ensures that two distinct managers are involved in writing and approving a purchase order.

Operational separation-of-duties: The "operational separation of duties" requires that no single participant of a role can perform all the operations related to a business transaction. Instead of specifying constraints based on objects, these types of policies concentrate on tasks within a transaction. For example, an accountant can prepare tax-filing and approve the tax-filing, where the same accountant cannot perform both the tasks of the tax-filing transaction. Compared to "object based separation-of-duties" that specify constraints on the lifecycle of an object, and "operational separation of duties" specify constrains on the lifecycle of a business process.

History based separation-of-duties: This imposes predefined order on the actions performed by roles. "Operational separation of duties" is also a form of "history based separation of duties". Other examples include where a user can only perform a task if he/she has performed another set of tasks. For example, a user can only agree to an online-privacy agreement after he/she has read (reached the end of the agreement statement using a web browser) the privacy statement.

B. Chinese Wall Security Policy

Chinese Wall security policy is another widely utilized policy to resolve conflict-of-interest situation that arise when data from different organizations are accessed by a user. This type of policy is expressed to ensure that once a user accesses some resources, that user cannot access any resource that would otherwise create a conflict-of-interest situation. Financial institutions, such as bank, enforce this type of policy so that an agent cannot access financial data of conflicting clients.

Chinese Wall security policy puts constraints on a subject's access of data based on the subject's current access rights on other data. The datasets are grouped into "conflict of interest classes" and a subject is allowed access on only one dataset of each of the conflict of interest classes [12]. For example, datasets for a financial institution can be grouped into two conflict of interest classes: *Oil Company* and *Insurance Company*. The class *Oil Company* has datasets for multiple companies, e.g., *A*, *B*, and *C* and the *Insurance Company*. has datasets for companies *X*, *Y*, and *Z*. Once an agent of the financial institution accesses the dataset *A*, the agent is prohibited from accessing any other datasets of the class *Oil Company*; however he/she can access a dataset, either *X*, *Y*, or *Z*, of the class *Insurance Company*.

C. Confidentiality and Privacy

Confidentiality and privacy policies are related to information flow constraints. Traditionally, information flow policies in military information systems protect confidential data. In these systems, mandatory access control (MAC) policies ensure that data can only flow on a predefined path of subjects that are classified from low to high clearance levels, such as *Public*, *Military*, and *Secret Service*. Data are also categorized to specify policies on categories of data that can be accessed by specific classes of subjects. Systems for dynamic coalition also require similar confidentiality policies on critical data shared among collaborators in distributed domains.

Confidentiality policies are often distinguished from privacy policies in that confidentiality policies express the interest of organizations where privacy policy protects the interest of individual user[17]. Privacy is also defined as having control over information about oneself [18]. In many systems, including healthcare information systems, the terms are utilized interchangeably.

In recent CSCW systems, security related to presence-awareness and privacy has become a concern. In contrast to operating systems and database management systems, which tend to hide the presence of one user from another, a collaboration system is required to support user-presence awareness. Privacy can also become an issue when one may need to hide the identity of one participant from another. In such cases, the presence of a participant may be only visible through his/her role or a pseudonym but not by name. Pseudonyms may be required when a particular member has performed a role's task and in future he/she may need to be referred to as part of the policy specification. For example, the identity of a reviewer of a conference paper needs to be hidden from the paper authors though the authors are able to direct questions to a particular reviewer.

Similarly, the working or the protocol of a collaborative activity may need to be hidden. For example, if tasks are assigned to its participants in a round-robin manner, this information may need to be hidden so that a task requester is not able to use this information for his/her own gain. A similar collaboration requirement can specify that only the owner of a role or group knows the identities of the role members. For example, in a conference submission workflow, the *Program Chair* role owns the *Reviewer* roles, who review submitted papers. However, none other than the users in the *Program Chair* role are permitted to view the identities of the members of the *Reviewer* roles.

To protect user specific data collected by online service providers, the Platform for Privacy Preferences (P3P) [19] provides recommendation for the provider to publish agreements on the way the data is collected, used, and stored. Hence P3P privacy policies not necessarily limit information flow initiated by subjects but agree on certain aspects on collection, usage, storage, and sharing or distribution of data related to user interactions with a specific service provider.

The "Standards for Privacy of Individually Identifiable Information" promulgated by the Department of Health and Human Services [20] regulates how personal information can be shared among various parties in healthcare services. The regulation sets boundaries on the use and release of individual data and holds violator accountable. A primary aspect of security policy in privacy domain is the control of the user on the data related to him/her-self, similar to *Originator Controlled Access Control* [21] policy. These privacy policies are expressed in terms of *consent*, *obligation*, and *data category* and *context*. For example a patient's consent is required when data is released to a different hospital. An obligation of the healthcare provider can be that all the access to the patient data will be logged. An example policy based on data category and context is that only the patient's billing information can be disclosed to insurance providers.

A related challenge in preserving privacy is to sanitize the data, when used for research or statistical reports, so that information cannot be linked with individuals. Research in database security [22] proposed solutions based on hiding related data in relational databases. Due to relational nature of the data and meta information of the data, traces of the hidden data remain. For such cases, replacement with false data is proposed. However, these types of solutions are proposed for centralized database systems. In current environment of distributed service providers, data is shared with organizations in different administrative domains. This introduces additional policy requirements to specify usage restrictions on the shared data. For example, an organization may like to specify a policy that shared data can only be used for research purposes and cannot be shared with other parties.

D. Context Sensitive Security Policies

Policies related to security attributes – privacy or confidentiality, integrity, and resource access – in CSCW and workflow systems need to handle various context-sensitive aspects of the collaboration environment. Such context can be physical location, coordination state of the cooperating users, proximity to devices, or any other application defined contextual information. For example, in a healthcare activity, physician can only view certain test results only during the surgical procedure.

Access rights, privileges, and ownership of objects may change in collaborative environments as activities progress. Sometimes permissions may change due to the user's own actions, such as making a final agreement by signing a contract. For example, only after final submission of a tax filing request by a user, an accountant can view the submitted information.

Several types of "separation of duties" constraints and history-based access control conditions also fall into the category of dynamic access control policies. Additional context-based access control may be related to physical environment's events. For example, in an organization, context based access control may specify that the managers can access employee files only when they are physically present in a specific room and that too only during a predefined period.

E. Security Policy in Workflow Systems

A workflow involves well-defined work activities or tasks, and security policy is expressed based on who can perform specific tasks. Security policies that are expressed based on tasks, such as "operational separation of duties" and Chinese Wall policy, are widely utilized in workflow systems. Task-based security policies are also constrained based on time period. For example, a workflow security policy can be that invoices can only be approved on Fridays.

The security policy in workflow systems has many constraints representing "well formed transaction", as expressed by Clark and Wilson [1]. An example of "well formed transaction" is the constraint that requires that entries in two accounting books are updated for each banking transaction. This provides integrity of data when one of the entry is corrupted. Other policies in workflow systems represent similar constraints. For example, in a banking system, a vault can only be opened with simultaneous presence of two bank managers. In many cases, these policies are derived from organizational level risk management strategy. For example, bank tellers require concurrent approval of another employee to transfer money over the limit of a certain amount.

Recent BPM systems, including E-commerce, B2B, and Web Services, have introduced various security requirements. Confidentiality requirements in an online auction with sealed-bids may require that the bidder identities to be kept secret. Online job-search and other match-making services provide confidentiality ensuring that match-making is performed based on only non-identifiable attributes of the clients [23]. E-commerce that brings together multiple parities – such as buyer, seller, bank, and insurance – need to ensure that only required information can be accessed by the parties involved. For example, the bank cannot access description of the product and the seller cannot access the bank account. Another form of access control, termed *usage control* [24], is also discussed in current workflow systems, so that the subject's access to a resource is revoked after usage limit. For example, an online service provider may only allow usage up to 30 hours.

In the area of E-Services, anonymous service, both the anonymity of service requesters and the anonymity of service providers are important security requirements. For example, a patient may like to get health related medical information without revealing his/her identity. On the other hand, service providers may like to sell items or services without their identity revealed.

Workflow	CSCW
Task based security policies	Unobtrusive coordination
Well formed transactions	Short−term credentials
Seperation−of−duties	Fine−grain access control
Chinese Wall security policies	Contex−sensitive access control
Privacy agreement	Privacy in presence awareness
Anonymous service	
Usage control	

Fig. 4. Attributes of security policies in workflow and CSCW systems

F. Security Policy in CSCW Systems

In CSCW systems, security is usually ignored to emphasize the motivations of cooperation and shared objectives. However, several literature and field studies, namely [25],[26], and [11], pointed the need of control and existence of conflicting goals among participants in groupware systems. Certain collaboration activity, such as a collaborative preparation of settlement documents by lawyers, is inherently adversarial [27]. In many existing groupware literature, a coordination constraint is viewed as an access control constraint and a clear separation of coordination and security policies is not present. In shared-view GUI oriented groupware systems in regard to unobtrusive coordination, such as preventing "scroll-war" [28] is addressed as a protection issue. In Suite [28], a collaborative editor, the scroll bar is a shared resource, and the collaborating users can manipulate it on their discretion. When the collaborative users block each other on their usage of the scroll bar, without following social courtesy, it is termed a "scroll-war". All the participants of a collaboration may not be identifiable, and they may not follow social courtesy of not engaging in a "scroll-war".

Security policy can be argued to be distinct from coordination policy. Access control policy denies access to information and restricts the flow of information. Security policy in a collaboration treats its participants as adversarial. On the other hand, coordination policy assumes the participants of a collaboration are cooperative for their best interest to achieve a goal. The coordination policy exists to facilitate the participants to share resources in an agreed upon manner to reach that goal. Hence, security policy needs to be specified for resources in a shared environment irrespective of the presence of a collaboration.

As in workflow systems, in groupware systems, users also perform their tasks based on well defined roles. For example, in a whiteboard sharing activity, users can be in *Moderator*, *Writer* and *Viewer* roles and acquire corresponding privileges for role specific tasks. However, unlike workflow, a user's roles may represent short-term privileges. For example, *Writer* role may be assigned to different users, one at a time, to coordinate sharing of the whiteboard. Another distinction is that security policies in groupware are often based on the attributes or the structure of the shared resources. This in turn requires fine-grain access control policies [28]. In collaborative design, different users may access or modify different parts of a shared object. Group-awareness technologies, such as presence-awareness in Instant Messaging, has introduced security policies regarding who can acquire the presence data, such as "buddy-list". This type of policies are tradeoff between unwanted interruption and enabling discovery of group activities. In video conferencing tools, hiding visible objects surrounding the communicating users is also addressed as a security issue to ensure that unwanted information does not leak [29].

G. Meta-Level Administrative Security Policies

In a decentralized execution environment, participants from domains with mutual distrust need to manage the shared resources, activities, and participants' roles in a collaboration. Policies related to identity management, object access, task creation and coordination, and other administrative aspects of managing collaborative activities require to be enforced in security domains managed by *administrators* who are *trusted*.

There are mainly two distinct forms of trust that are discussed in the context of distributed systems. First, trust relations are defined for distributed authentication of identity related certificates for encryption and access control [30]. Second, trust is defined as an entity's "trustworthiness" and discussed in the context of deriving recommendation or reputation [31], [32]. This type of trust is termed as *behavior trust* [32]. In addition to these trust concepts, in a decentralized collaboration environment, trust needs to be assigned to entities for administrative task of enforcing policies and performing management functionalities that may arise at runtime.

In most of the existing systems, administrators are trusted not to violate polices under their control. When administrators are in domains with mutual distrust or lack of trust, meta-level policies are required specifying facts regarding trust among these cooperating domains. Meta policies are needed to be specified for administrative roles that are trusted to enforce policies. Examples of such meta policies include specifying who can be present in these administrative roles.

In collaboration environments, such as ad hoc collaboration of mobile users and peer-to-peer systems, there may not be any dedicated administrative entity. Rather each user represents the administrator of its own unique domain. For example, in a conference review process workflow, participants may form a collaboration using their own mobile computing devices and collaboratively maintain review decisions and corresponding reports. Similarly, in peer-to-peer systems, a user may trust other online users to store his/her resources. Trusting strangers to form such collaborations raises security policy questions that are traditionally addressed as part of the security policy enforcement mechanisms. For example, identification of users based on their off-line credentials and accepting resources from users whose identities may not be verified.

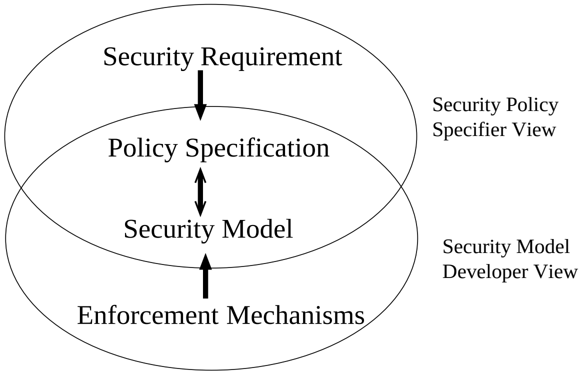

Fig. 5. Security requirement, policy, model, and enforcement mechanisms

In distributed environments, due to the lack of cost effective solutions of cryptographic services, the users are prompted to choose from a wide variety of cryptographic protocols, algorithms, and related attributes, such as, the length of the shared secret key and the strength of the cryptographic protocols. Often these decisions are pushed to the user level as security policy requirements. For example, group session security requirements are addressed by GSAKMP [33].

IV. Security Models

Security policy is the specification of security requirements, usually specified based on some security model (see Figure 5). A security model usually represents a particular set of policies [34]. Traditional security models are classified in two groups: Discretionary Access Control (DAC) and Mandatory Access Control (MAC) models. In DAC models, access right can be changed on discretion of a user, for example access control model in a UNIX file system. On the other hand, MAC models represent mandatory policies where access control are enforced by trusted organization level enforcement mechanisms, and an individual user does not have control over changing access rights. For example, lattice based military access control models.

As security models are developed based on security requirements of particular system, often these models are specific to security properties, e.g., Clark and Wilson model for integrity, Bell and LaPadula model [35] and its derivative lattice based models for confidentiality, and take-grant model [36] for access leakage. Though these models have various formal results regarding the types of security properties they support, there are several concerns:

1) The security model needs to be expressive to specify a wide range of security policies, such as separation of duties and other context dependent security policies.

2) Administrative capabilities to manage security policies is a major usage concern. Management of security policies is a challenge where user population is large and dynamic, i.e., users are joining and leaving an organization. On the other hand, MAC based policies are hard to manage due to the administrative requirements of proper categorization of data and users.

3) Ideally a security model should be able to ensure that the specified constraints do not violate any desired security properties, such as:

   a) Confidential information cannot flow to unauthorized users;

   b) Authorized information can be accessed;

   c) Any temporal or conditional constraints on accessing objects can be satisfied;

   d) No rights can be leaked to unauthorized users.

   Most of the existing security models, specifically for distributed CSCW and workflow systems, cannot support all of the above security properties.

4) As security requirements may depend on complex conditions, conflicting or inconsistent security requirements may be specified. An inconsistent policy may specify that a user can access an object if he/she has accessed it earlier. Policy conflict also arise due to granting negative and positive access-rights to the same user. For distributed CSCW and workflow systems, a security policy specification methodology is required that either provides conflict resolution rules [37] or supports verification of security policies to ensure that the policy specification satisfies security requirements [38].

A. Security Model for Access Leakage

Ideally, a security model should be able to ensure that no rights can be leaked to unauthorized users. This property is known as *safety* property. The safety property of the HRU access matrix model [39] that represents generic protection systems is not decidable. On the other hand, safety in take-grant model [36], a graph based security model, is not only decidable but also has linear complexity that is proportional with the initial size of the access graph. This is due to the fact that take-grant model is restricted to express capability based systems, i.e., it only expresses properties related to delegation of rights. Based on this fact, later access control models have placed constraints on access control structures to facilitate analysis of safety properties, such as, Typed Access Matrix [40]. In other cases, users with administrative rights, e.g., *create-subject*, are trusted for not violating security properties.

B. Security Models for Confidentiality

Initially, access control model was proposed as security model for confidentiality. Early example of access control model for confidentiality is the Bell and LaPadula model that imposes mandatory access control on the data that a subject can read or write based on classification of data and subjects. Based on imposed access control restrictions, information can only flow from users with low classification to users with high classification. Access control models are easy to implement with tamper-proof execution monitor, also known as “reference monitor”, that interposes on subject’s request to access any object [41]. On the other hand, this type of model can only enforce confidentiality policies on information that can be leaked through *storage channel*, e.g., disk file or computer memory. Another threat of information leakage is *covert channel* [42] that is not intended for information transfer but represents effect of the runtime program. For example, the size of the UNIX *tmp* directory can be used to pass information between two confined programs. McLean [41] argued that covert channels are real threats as the capacity of such channels can be large due to the rise of computers’ processing speed. Even when the users are trusted not to violate confidentiality policies, *Trojan Horses* can utilize these channels. Trojan Horses are programs that disguise themselves as legal programs but have covert behaviors.

To analyze covert channels as part of security models, *interface model* for confidentiality has been introduced. Interface models specify restrictions on systems input and output interfaces to ensure enforcement of confidentiality properties. Based on the behaviors of systems and covert channels, various types of confidentiality properties for interface models have been introduced, such as noninterference, noninference, and non-deducible [43]. For runtime enforcement of confidentiality properties, it is argued that an execution monitor to ensure secrecy cannot be complete [44], due to covert channels. Denial of an execution step by itself represents a covert channel. Therefore, static analysis of a system for confidentiality properties is preferred to ensure secrecy.

Model oriented formal methods, such as CSP (Communicating Sequential Processes) [45], SPA (Security Process Algebra) [46], TLA (Temporal Logic of Actions) [47] and state based techniques [48] are used to statically verify various types of confidentiality properties. In the property-oriented formal methods, theorem provers [49], [50] and type systems [51], [52], [53], [54] are used for analyzing information flow. However, these approaches assume that the verified programs will run under trusted subjects. In decentralized administration of systems, programs are type-checked based on assigned “trust” [55] or labeling data [56]; however, the execution environment is assumed to be entirely

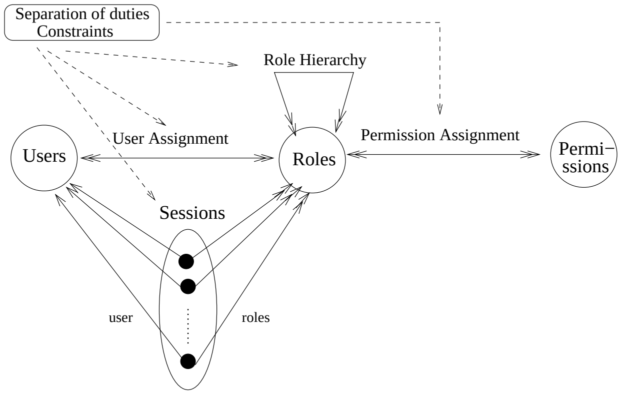

Fig. 6. NIST RBAC models [13]

trusted. We have utilized finite state based model checking to verify security properties in collaboration systems [57]. In our verification approach, some of the distributed security domains can be under the control of administrators who may not be trusted, and the behaviors of untrusted administrators are modeled to verify security properties [38].

V. Role Based Access Control

Role based access control (RBAC) is policy neutral [58], i.e., it is not tied to any specific policy model, such as MAC or DAC, and is able to express security policies related to MAC and DAC models. As the definition of a role within an organization changes less frequently compared with the turn-around rates of people, role based systems are said to provide ease of user management. The primary motivation behind role based access control models is their ability to express various forms of security constraints, such as "separation of duties" and role cardinality constraints. A role cardinality constraint ensures the presence of a maximum or a minimum number of users before a role privilege can be executed.

The concept of roles and related theories have been widely studied in the past in the context of behavioral science [59]. Due to different interpretation of roles, based on application domains, such as distributed systems [60], network management [61], database management systems [62], and interactive groupware [28], a wide range of role based models have been proposed. NIST has proposed a unified standard for role based access control (RBAC) reference models [13]. NIST RBAC models have three primary constructs: *User*, *Role*, and *Permission*, as shown in Figure 6. Roles are assigned a set of permissions and users are assigned to roles. A user can have multiple roles, and the same permission can be assigned to multiple roles. The one-to-many relation is shown with a double arrow, where as the one-to-one relation is presented with a single arrow, in the Figure 6.

NIST RBAC models support the concept of role hierarchy to define roles where a *senior* role in a hierarchy can acquire all the privileges of *junior* roles. In the Figure 7, a *System Administrator* role has the *execute* permission, a *Developer* role has the *write* permission, and a junior role *Reader* has the *read* permission. Both the *System Administrator* and *Developer* roles acquire the *read* permission. In this example role hierarchy, the *Reader* role is declared to contain the common permission *read* for the two senior roles. On the other hand, the *Manager* role aggregates all permissions from the subordinate *Test Manager* and *System Administrator* roles.

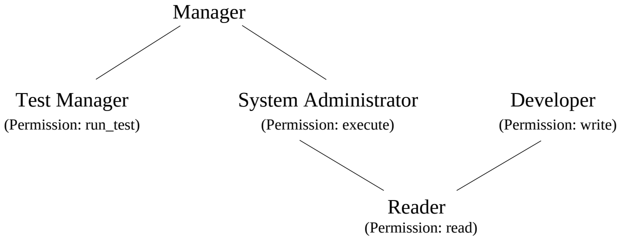

Fig. 7. Example role hierarchy

To perform some task, a user may require a set of permissions. These permissions may be acquired based on different sets of roles that are assigned to the user. Assume in our example, a user is assigned to both the *System Administrator* and the *Developer* roles. When the user wants to read a file, there are multiple choices of roles that can perform the task. These choices are *System Administrator*, *Developer*, *Reader*, or any combinations of these roles. In RBAC [13], a user's acquisition of a set of roles to perform some task is defined as a *session*. To decide on the set of roles for a session, different rules can be used. Based on the least privilege principle, a set of roles can be selected that result in least number of permissions. In this example, the *Reader* role has the least number of permissions for the session to read a file.

In NIST RBAC models, "separation of duties" constraints can be specified on the relations of *user assignments to roles*, *permission assignments to roles*, role hierarchies, and sessions. A "static separation of duties" can be realized by a constraint on user assignments to roles.

NIST acknowledges that RBAC is an open-ended technology and does not address all the issues in role based security [13]. In the following subsections, several such issues that are essential in distributed CSCW and workflow systems are discussed.

A. Role Membership Management

NIST RBAC models do not incorporate administrative issues in user assignments and permission assignments to roles due to lack of consensus. Administrative RBAC that is proposed with *administrative roles* can manage user assignment in a RBAC model [63]. ARBAC declares separate role hierarchies with administrative roles that have administrative privileges of managing users on the roles declared within an organization.

On the other hand, in distributed CSCW and workflow systems, role based model needs to address constraints on users' acquiring a role. Role admission constraints specify the conditions that need to be satisfied when a user requests to join a role. The admission constraints can be based on several different criteria: a list of users that should be allowed to join a role; a list of those who should never be admitted; role membership cardinality specifying the maximum number of users that can join the role; events that must happen before a user can be admitted in a role, such as a predefined time period, tasks performed by others, or previous qualifications requiring that the requesting user has been or is currently admitted in some other given roles. The constraints based on previous qualifications may require that the membership in a prerequisite role is also *valid*. If a role membership in a prerequisite role depends on the user's membership in other roles, than those memberships must also be current. Traditionally RBAC models are centralized and only address user assignment to roles. In distributed systems, these role admission related constraints support the functionality of acquiring and revoking role memberships [60].

B. Dynamic and Context Sensitive RBAC Models

NIST RBAC, and most of the existing MAC and DAC style security policies are static or passive, i.e., they do not depend on time or other events. Moreover, they do not differentiate permission assignment and permission activations [64]. In collaboration environments, different permissions may need to be assigned to roles based on various contexts and events. Moreover, NIST RBAC models do not address any history dependent constraints.

Active database research has addressed trigger based authorization changes and used the notion of condition based access control [65]. In recent years, different RBAC models are proposed to address issues related to context sensitive access control constraints. Team Based Access Control (TMAC) [64] discusses *team* as a context for roles. A *Physician* role can be member of multiple teams, e.g., *Surgery* and *Emergency Room*. Based on the context of the team, *Surgery* or *Emergency Room*, the *Physician* role acquire different sets of privileges. Task based authorization models [66], [67] express constraints on tasks that can be performed by a role based on its task execution history.

C. Intra- Role Constraints

When multiple users are allowed to be present simultaneously in a role, their actions may need to be constrainted based on actions of other participants within the role, which is termed as *intra-role constraints* [68]. Multiple users present in a role can participate either *independently* or *cooperatively*. In *independent participation*, all role specific task-responsibilities are assumed individually by a role member, irrespective of the presence of the other members, e.g., every member of a conference *Reviewer* role has to independently write a review. On the other hand, when the members in a role are assuming task responsibilities *cooperatively*, their actions need to be coordinated. For example, in a hospital patient ward, several nurses may be present in the role of *nurse-on-duty*. However, some medical procedure on a patient may need to be performed only once by any of the nurses. Another type of cooperation may require a task to be performed by all the members of a role, like jointly opening a bank vault. Moreover, in some collaboration environments there may be no coordination among participants' actions, e.g. in an unrestricted whiteboard sharing.

D. Policy Specification Methodology: Role Engineering and Constraint Specification

NIST RBAC models do not address role modeling or consistent specification of role constraints. Modeling or engineering of roles is a challenging concern in role based security models. Role has different interpretation based on application domains. In distributed systems [60], role is viewed a certified capabilities for authorizing access to services. In network management [61], role based management in proposed with support for *obligation* policy. Obligation policy specifies the actions that a role has to perform when certain conditions become true. In database management systems [62], role is termed "Named Protection Domain" and resembles capabilities. Only a few recent research addresses software engineering methodology to model roles [69].

In workflow management systems, *task* is primary construct of modeling workflow processes, and many security constraints are specified based on subjects' authorization to perform specific tasks. In workflow systems [70], constraints are specified with a mapping between roles and tasks. In this model, roles need to be related according to a global order so that roles can be prioritized during task assignment. SecureFlow [66] imposes workflow authorization constraints on tasks using *Authorization Template*. An *Authorization Template* is a tuple specifying privileges to be granted to a subject of a given role on a object of a given type during a given time interval. There, the permissions are activated based on tasks.

In RBAC, safety of various role based constraints, such as "separation of duties", have been analyzed with logical expression using rule-based systems [67], [71], [66] and graphical models [16], [15], [72], [73]. However, the verification in most of these existing research [67], [71], [66] is either performed in the context of centralized management of systems or the participants in administrative roles are trusted to enforce policies without taking into account the mutual distrust in collaborating domains.

We have developed a methodology, based on an extended RBAC model, to specify and verify security requirements in distributed CSCW systems [68], [57], [38]. For ease of modeling collaboration environments, we defined the concept of *activity*. In our collaboration model, an activity defines how a group of users cooperate toward some common objectives by conducting their individual tasks on a set of shared objects.

In an activity, users are represented by their roles, and roles are assigned privileges to perform certain tasks, termed *operation*. Each operation has *preconditions*, and each role has *admission constraints*. These conditions ensures various role constraints, such as “separation of duties”, intra-role constraints, and context sensitive access control policies.

In our model, an activity is an abstraction of a collaboration session, which provides a scope for objects, roles, and privileges in the collaboration. An activity can be structured hierarchically, consisting of multiple nested concurrent activities. Objects can be passed into nested activities, and users in roles from the parent activity can be statically or dynamically assigned to roles in nested activities. An *activity template* specifies a generic collaboration pattern among a set of roles using some shared objects. Any number of instances of a template can be dynamically and concurrently created.

VI. Policy Enforcement Challenges in Distributed CSCW and Workflow Systems

Policy enforcement mechanism plays an important role in developing security policy specification models. Without proper authentication, access control models are useless. If a user can acquire multiple identities within a domain, various cross-organization level security policies, such as Chinese Wall policy, cannot be enforced. For the same reasons, if the enforcement mechanisms are under control of an administrator who cannot be trusted, security policies cannot be enforced.

In this section, two important aspects of security policy enforcement mechanism in distributed CSCW and workflow systems are discussed, namely trust on the enforcement mechanisms and interoperability of security policies.

A. Trust on Security Mechanisms

In the paradigm of trusted computer system initiated by the *Trusted Computing System Evaluation Criteria* (TCSEC), trust is viewed as a property of a system [74], [75]. According to the Criteria, the Trusted Computing Base (TCB) is the only part of the system that need to be evaluated to prove that trust is present. Hence, the trust on the TCB can be viewed as the knowledge acquired based on derived trust from: (1) the trust in the formal verification methods and (2) the trust in the trust derivation mechanism [76]. Later TCB concept is extended in distributed TCB, where untrusted communication medium is introduced under TCB boundaries, as oppose to traditional untrusted applications run over a TCB boundary, and messages through untrusted media is proposed to be protected through cryptographic techniques [77].

Trust is defined as the result of an assessment made by an observer about a person, organization, or any other entity [74]. In distributed systems, trust is classified where being trusted in a class means that an entity is trusted to perform specific task, such as providing identification of another entity, providing good quality keys, maintaining secret, and providing options about the trustworthiness characteristics of other entities [78]. As opposed to trust derived in TCB from the proof of the the specification of functionality, in recent Internet wide systems, the trust concern in shifted to the interacting entities including human users and services. Trust is discussed in the context of certification of an entity’s attributes, such as roles. These attribute certificates are also called credentials. Often these credentials are cascaded, i.e, validity of a credential depends on the validity of other credentials. Cost effective solutions to find the right set of credentials for an authorization and to revoke credentials are challenges of trust management systems.

In collaboration environments, in addition to be concerned about the trust on the enforcement mechanisms and the credential management, research challenges include ensuring trust on entities with management responsibilities. In decentralized setting, this implies assigning trust on an entity for administrative task of enforcing policies and performing management and obligatory functionalities that arise at runtime.

B. Interoperability of Security Policies

For E-Commerce, Secure Electronic Transaction (SET) [79] protocol provided security services, such as authentication, authorization, integrity, non-repudiation, and confidentiality, for multi-party financial transactions. However, such protocols are limited to well-understood security requirements in specific service domain. For multi-domain business orchestration, such as B2B and Web services, interoperability of security policies of different organizations and systems is a major concern. Security aspects of service provisioning need to address agreements among these domains to adopt a standard vocabulary to express security requirements and a common transport mechanism to interact.

To solve these challenges, Web Services has adopted Extensible Markup Language (XML) to resolve all interoperability concerns, from policy specification to message transport. XML provides the facilities to define tags to express any structural content using text. Based on domain knowledge, XML schemas are developed to represent such structures, and these schemas are shared among interacting systems. Several security related schema standards for Web Services security have been developed. Among them, Security Assertions Markup Language (SAML) [80] provides a schema for Web Services to exchange information related to authentication and authorization using trusted statements, termed *assertions*. In SAML, there are three types of security assertions: (1) authentication assertions issued by authentication servers, (2) attribute assertions, by attribute servers, related to attributes required to access a service, and (3) authorization assertions, which are generated based on the previous two types of assertions, to access a service. A service request in Web Services may require interactions of multiple service providers, and SAML provides assertion, similar to “single sign on”, to interact with these services.

Another schema, XML Access Control Markup Language (XACML) [81], is developed to express access control policies for XML content. For example, healthcare providers can express what parts of an patient record can be accessed. However, XACML does not define the schema terms that are required to express privacy polices by healthcare providers, such as obligation, consent, and purpose, as discussed in Section III. Based on a model to express privacy related policy constructs in enterprises a schema is proposed to W3C [82], [83].

VII. Conclusions

In this paper, we have discussed security policy requirements in distributed CSCW and workflow systems. Based on the historical perspective of management aspects and systems' emphases on users or processes, example systems that are discussed in CSCW and workflow literature are presented. Special aspects of the security requirements for these systems are also discussed. Traditional security models that are developed to ensure security properties related to confidentiality, integrity, and access leakage have been surveyed. As role based access control models are widely utilized in CSCW and workflow systems, we have discussed RBAC models and research challenges in RBAC. Lastly, we present two concerns in security policy enforcement mechanisms in distributed CSCW and workflow systems, namely trust on enforcement mechanisms and interoperability of cross-organizational policies.

References

[1] D. D. Clark and D. R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," in *Proceedings of the IEEE Symposium on Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1987, pp. 184–194.

[2] J. Grudin and S. E. Poltrock, "Computer-Supported Cooperative Work and Groupware," *Advances in Computers*, vol. 45, pp. 269–320, 1997.

[3] D. Hollingsworth, "The Workflow Reference Model: 10 Years On," Workflow Management Coalition, Tech. Rep., 2004, available at http://www.wfmc.org/information/info.htm.

[4] I. Greif and S. Sarin, "Data sharing in group work," *ACM Transactions on Information Systems*, vol. 5, no. 2, pp. 187–211, 1987.

[5] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: some issues and experiences," *Communication of ACM*, vol. 34, no. 1, pp. 39 – 58, January 1991.

[6] J. A. Oravec, *Virtual individuals, virtual groups : human dimensions of groupware and computer networking*. Cambridge; New York: Cambridge University Press, 1996.

[7] J. Bardram, "Designing for the Dynamics of Cooperative Work Activities," in *Proceedings of CSCW'98*, 1998, pp. 89–98.

[8] T. F. Stafford, "E-Services," *Communications of the ACM*, vol. 46, no. 6, pp. 27–28, June 2003.

[9] W3C, "Web Services Activity," Available at URL http://www.w3.org/2002/ws/, 2002.

[10] P. Dourish, "Software infrastructures," in *Computer Supported Cooperative Work*, M. Beaudouin-Lafon, Ed. John Wiley & Sons Ltd., 1999, vol. 7 of Trends in Software, pp. 195–219.

[11] W. J. Orlikowski, "Learning from Notes: Organizational Issues in Groupware Implementation," in *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work*, December 1992, pp. 362 – 369.

[12] D. Brewer and M. Nash, "The Chinese Wall Security Policy," in *1989 IEEE Symposium on Security and Privacy*, 1989, pp. 206 –214.

[13] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: towards a unified standard," in *Proceedings of the Fifth ACM Workshop on Role-based Access Control*. New York: ACM, July 2000, pp. 47–63.

[14] R. Simon and M. Zurko, "Separation of duty in role-based environments," in *10th Computer Security Foundations Workshop*. Los Alamitos, CA: IEEE Computer Society Press, 1997, pp. 183 –194.

[15] M. Nyanchama and S. Osborn, "The Role Graph Model and Conflict of Interest," *ACM Transaction on Information System Security*, vol. 2, no. 1, pp. 3–33, February 1999.

[16] T. Jaeger and J. E. Tidswell, "Practical Safety in Flexible Access Control Models," *ACM Transactions on Information and System Security*, vol. 4, no. 2, pp. 158 – 190, May 2001.

[17] R. J. Anderson, "A security policy model for clinical information systems," in *IEEE Symposium on Security and Privacy*, May 1996, pp. 30 – 43.

[18] H. T. Tavani and J. H. Moor, "Privacy protection, control of information, and privacy-enhancing technologies ," *ACM SIGCAS Computers and Society*, vol. 31, no. 1, pp. 6 –11, March 2001.

[19] W3C, "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification, W3C Recommendation," Available at URL http://www.w3c.org/TR/P3P/, April 2002.

[20] Department of Health and Human Services, "Standards for Privacy of Individually Identifiable Health Information," Available at URL http://aspe.hhs.gov/admnsimp/final/pvcguide1.htm.

[21] R. Graubart, "On the need for a Third Form of Access Control," in *Proceedings of the 12th National Computer Security Conference*, Oct. 1989, pp. 296–304.

[22] T. Lunt, D. Denning, R. Schell, M. Heckman, and W. Shockley, "The SeaView security model," *IEEE Transactions on Software Engineering*, vol. 16, no. 6, pp. 593 –607, June 1990.

[23] S. Jajodia, M. Kudo, and V. S. Subrahmanian, "Provisional authorization," in *In Proc. Ecommerce Security and Privacy*. Kluwer Academic Publishers, 2001, pp. 133–159.

[24] J. Park and R. Sandhu, "The UCONABC usage control model," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, February 2004.

[25] R. Kling, "Cooperation, Coordination and Control in Computer-Supported Work," *Communications of the ACM*, vol. 34, no. 12, pp. 83 – 88, December 1991.

[26] M. Kyng, "Designing for Cooperation: Cooperating in Design," *Communications of the ACM*, vol. 34, no. 12, pp. 65–73, 1991.

[27] A. L. Cohen, D. Cash, and M. J. Muller, "Designing to support adversarial collaboration," in *Proceeding on the ACM Conference on Computer supported cooperative work*, December 2000, pp. 31 – 39.

[28] P. Dewan and H. Shen, "Controlling access in multiuser interfaces," *ACM Transaction Computer-Human Interaction*, vol. 5, no. 1, pp. 34 – 62, March 1998.

[29] S. E. Hudson and I. Smith, "Techniques for addressing fundamental privacy and disruption tradeoffs in awareness support systems," in *Proceedings of ACM conference on computer supported cooperative work* , November 1996, pp. 248 – 257.

[30] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized Trust Management," in *Symposium on Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1996, pp. 164–173.

[31] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of the Workshop on New Security Paradigms Workshop*. New York: ACM, 1998, pp. 48 – 60.

[32] F. Azzedin and M. Maheswaran, "Trust modeling for peer-to-peer based computing systems," in *Parallel and Distributed Processing Symposium*, April 2003, pp. 99 –108.

[33] H. Harney, A. Colegrove, and P. McDaniel, "Principles of Policy in Secure Groups," in *Proceedings of Network and Distributed Systems Security*, February 2001.

[34] M. Bishop, *Computer Security: Art and Science*. Addison Wesley Professional, 2000.

[35] D. E. Bell and L. La Padula, "Secure Computer System: Unified Exposition and Multics Interpretation," ESD-TR-75-306, ESD/AFSC, Hanscom AFB, Bedford, MA, Tech. Rep., 1975.

[36] L. Snyder, "Formal Models of Capability-Based Protection Systems," *IEEE Transactions on Computers*, vol. C-30, no. 3, pp. 172 – 181, March 1981.

[37] S. Jajodia, P. Samarati, and V. S. Subrahmanian, "A Logical Language for Expressing Authorizations," in *IEEE Symposium on Security and Privacy*, 1997, pp. 31 –42.

[38] T. Ahmed and A. Tripathi, "Specification and Verification of Security Requirements in Decentralized CSCW Systems," Department of Computer Science, University of Minnesota, Tech. Rep., December 2003, available at http://www.cs.umn.edu/Ajanta/publications.html.

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

[40] R. Sandhu, “The Typed Access Matrix Model,” in *IEEE Computer Society Symposium on Research in Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1992, pp. 122 –136.

[41] J. McLean, “Security Models,” in *Encyclopedia of Software Engineering*, J. Marciniak, Ed. John Wiley & Sons, 1994.

[42] B. W. Lampson, “A note on the confinement problem,” *Communications of the ACM*, vol. 16, no. 10, October 1973.

[43] A. Zakinthinos and E. Lee, “A General Theory of Security Properties,” in *IEEE Symposium on Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1997, pp. 94 –102.

[44] D. M. Volpano, “Safety versus Secrecy,” in *Static Analysis Symposium*. Springer-Verlag, 1999, pp. 303–311.

[45] S. Schneider, “Security properties and CSP,” in *IEEE Symposium on Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1996, pp. 174 –187.

[46] R. Focardi and R. Gorrieri, “The Compositional Security Checker: A Tool for the Verification of Information Flow Security Properties,” *IEEE Transactions on Software Engineering*, vol. 23, no. 9, pp. 550–571, 997.

[47] T. Fine, “Defining Noninterference in the Temporal Logic of Actions,” in *IEEE Symposium on Security and Privacy*, 1996, pp. 12–23.

[48] W. R. Bevier and W. D. Young, “A state-based approach to noninterference,” in *Proceedings of 7th Computer Security Foundations Workshop*, June 1994, pp. 11–21.

[49] G. R. Andrews and R. P. Reitman, “An Axiomatic Approach to Information Flow in Programs,” *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 2, no. 1, pp. 56–76, January 1980.

[50] F. Martinelli, “Partial Model Checking and Theorem Proving for Ensuring Security Properties,” in *Proceedings of 11th IEEE Computer Security Foundations Workshop*. Los Alamitos, CA: IEEE Computer Society Press, 1998, pp. 44 –52.

[51] D. E. Denning and P. J. Denning, “Certification of programs for secure information flow,” *Communications of the ACM*, vol. 20, no. 7, pp. 504–513, July 1977.

[52] J. K. Millen, “Information Flow Analysis of Formal Specifications,” in *IEEE Symposium on Security and Privacy*, 1981, pp. 3 – 8.

[53] D. M. Volpano and G. Smith, “Verifying Secrets and Relative Secrecy,” in *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*. New York: ACM, 2000, pp. 268–276.

[54] V. Simonet, “Fine-grained Information Flow Analysis for a lambda-calculus with Sum Types,” in *Proceedings of. 15th IEEE Computer Security Foundations Workshop*, 2002, pp. 209 –223.

[55] P. Ørbæk and J. Palsberg, “Trust in the λ-calculus,” *Journal of Functional Programming*, vol. 7, no. 6, pp. 557–591, Nov. 1997.

[56] A. C. Myers and B. Liskov, “Protecting privacy using the decentralized label model,” *ACM Transactions on Software Engineering and Methodology*, vol. 9, no. 4, pp. 410–442, 2000.

[57] T. Ahmed and A. R. Tripathi, “Static Verification of Security Requirements in Role Based CSCW Systems,” in *Proceedings of 8th ACM Symposium on Access Control Models and Technologies (SACMAT 2003)*. New York: ACM, June 2003, pp. 196–203.

[58] R. Sandhu, “Role activation hierarchies,” in *Proceedings of the third ACM workshop on Role-based access control*, 1998, pp. 33 – 40.

[59] B. J. Biddle, *Role theory: concepts and research*. New York, Wiley, 1966, edited by Bruce J. Biddle and Edwin J. Thomas.

[60] J. Bacon, K. Moody, and W. Yao, “A Model of OASIS Role-Based Access Control and its Support for Active Security,” *ACM Transactions on Information and System Security*, vol. 5, no. 4, pp. 492 – 540, November 2002.

[61] E. C. Lupu and M. Sloman, “Reconciling Role-Based Management and Role-Based Access Control,” in *ACM Workshop on Role-based Access Control*. New York: ACM, 1997, pp. 135–141.

[62] R. Baldwin, “Naming and grouping privileges to simplify security management in large databases,” in *IEEE Computer Society Symposium on Research in Security and Privacy*, 1990, pp. 116 –132.

[63] R. Sandhu, V. Bhamidipati, and Q. Munawer, “The ARBAC97 model for role-based administration of roles,” *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 105 – 135, February 1999.

[64] R. K. Thomas, “Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments,” in *ACM Workshop on Role-based Access Control*. New York: ACM, 1997, pp. 13 – 19.

[65] I. Mohammed and D. M. Dilts, “Design for dynamic user-role-based security,” *Computers & Security*, vol. 13, no. 8, pp. 661–671, 1994.

[66] W.-K. Huang and V. Atluri, “SecureFlow: A Secure Web-enabled Workflow Management System,” in *ACM Workshop on Role-based Access Control*. New York: ACM, 1999, pp. 83 – 94.

[67] E. Bertino and E. Ferrari, “The Specification and Enforcement of Authorization Constraints in Workflow Management Systems,” *ACM Transactions on Information and System Security*, vol. 2, no. 1, pp. 65 – 104, February 1999.

[68] A. Tripathi, T. Ahmed, and R. Kumar, “Specification of Secure Distributed Collaboration Systems,” in *IEEE International Symposium on Autonomous Distributed Systems*. Los Alamitos, CA: IEEE Computer Society Press, April 2003, pp. 149–156.

[69] P. Epstein and R. Sandhu, “ Engineering of Role/Permission Assignments,” in *17th Annual Computer Security Applications Conference* , December 2001.

[70] E. Bertino, E. Ferrari, and V. Atluri, “A Flexible Model Supporting the Specification and Enforcement of Role-based Authorizations in Workflow Management Systems,” in *ACM Workshop on Role-based Access Control*, 1997, pp. 1–12.

[71] G.-J. Ahn and R. Sandhu, “Role-based authorization constraints specification,” *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 207 – 226, November 2000.

[72] S. L. Osborn, “Information Flow Analysis of an RBAC System,” in *ACM Symposium on Access Control Models and Technologies*. New York: ACM, 2002, pp. 163 – 168.

[73] M. Koch, L. V. Mancini, and F. Parisi-Presicce, “A graph-based formalism for RBAC,” *ACM Transactions on Information and System Security*, vol. 5, no. 3, pp. 332 – 365, August 2002.

[74] D. E. Denning, “A new paradigm for trusted systems,” in *Proceedings on the 1992-1993 Workshop on New Security Paradigms*. New York: ACM, August 1993, pp. 36 – 41.

[75] R. Dobry and M. Schanken, “Security concerns for distributed systems,” in *Proceedings of Computer Security Applications Conference*, Dec 1994, pp. 12 –20.

[76] A. Jsang, “The right type of trust for distributed systems,” in *Proceedings of the UCLA Conference on New Security Paradigms Workshops*, 1996, pp. 119 – 131.

[77] T. J. Casey, S. Vinter, D. Weber, R. Varadarajan, and D. Rosenthal, “A secure distributed operating system,” in *IEEE Symposium on Security and Privacy*, April 1988, pp. 27 –38.

[78] R. Yahalom, B. Klein, and T. Beth, “Trust relationships in secure systems-a distributed authentication perspective,” in *IEEE Computer Society Symposium on Research in Security and Privacy*. Los Alamitos, CA: IEEE Computer Society Press, 1993, pp. 150 –164.

[79] Y. Kawatsura, “RFC 3538 - Secure Electronic Transaction (SET) Supplement for the v1.0 Internet Open Trading Protocol (IOTP),” Available at URL http://www.faqs.org/rfcs/rfc3538.html, June 2003.

[80] OASIS Security Services TC , “Security Assertion Markup Language (SAML) v1.1,” Available at URL http://www.oasis-open.org/specs, August 2003.

[81] OASIS Extensible Access Control Markup Language TC , “Extensible Access Control Markup Language (XACML) v1.0,” Available at URL http://www.oasis-open.org/specs, February 2003.

[82] G. Karjoth and M. Schunter, “A privacy policy model for enterprises,” in *Proceedings of 15th IEEE Computer Security Foundations Workshop*, June 2002, pp. 24–26.

[83] P. Ashley, S. Hada, G. Karjoth, C. Powers, and M. Schunter, “IBM Enterprise Privacy Authorization Language (EPAL 1.2), W3C Member Submission,” Available at URL http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/, November 2003.