Artif Intell Rev (2012) 37:239–260
DOI 10.1007/s10462-011-9228-8

A survey of security issue in multi-agent systems

Youna Jung · Minsoo Kim · Amirreza Masoumzadeh · James B. D. Joshi

Published online: 4 June 2011

Abstract Multi-agent systems have attracted the attention of researchers because of agents' automatic, pro-active, and dynamic problem solving behaviors. Consequently, there has been a rapid development in agent technology which has enabled us to provide or receive useful and convenient services in a variety of areas such as banking, transportation, e-business, and healthcare. In many of these services, it is, however, necessary that security is guaranteed. Unless we guarantee the security services based on agent-based systems, these services will face significant deployment problems. In this paper, we survey existing work related to security in multi-agent systems, especially focused on access control and trust/reputation, and then present our analyses. We also present existing problems and discuss future research challenges.

Keywords Intelligent agents · Multiagent system · Security · Access control · Trust · Reputation

1 Introduction

An agent is an autonomous and goal-oriented software entity which collaborates and communicates with other software entities and humans (Greenberg et al. 1998). There are various definitions of an agent, but the most common characteristics of agents are: *autonomy*, *social ability*, *reactivity*, and *pro-activity* (Wooldridge and Jennings 1995). Because of such characteristics, the agent paradigm has become a promising technology for developing applications in open, distributed, and heterogeneous environments. Indeed, agent-based systems have been widely developed in open distributed environments, especially in electronic commerce, mobile computing, network management, and information retrieval areas.

Y. Jung (✉) · M. Kim · A. Masoumzadeh · J. B. D. Joshi
LERSAIS, School of Information Sciences, University of Pittsburgh, 410 IS Building, 135 N. Bellefield Avenue, Pittsburgh, PA 15260, USA
e-mail: younajung@gmail.com

In agent-based systems, agents may try to obtain information from other agents or gain access to remote service provider agents in order to achieve their goals. Unfortunately, in open environments where agents are able to freely move around, such activities would be unsafe and unreliable because it is hard to know which agents are trustworthy and which external accesses are not harmful. Without appropriate solutions to such security problems, some sensitive information can be leaked or a system can be easily compromised. Particularly, some critical transactions, such as those related to banking or personal healthcare systems, must be securely performed. However, agent's characteristics, such as *autonomy*, *heterogeneity*, and *openness*, make it hard to guarantee security of these systems. Nevertheless, it is important to secure MASs in order to fully benefit from agent technology for a wide range of applications. To protect agent-based systems against security threats, we need to guarantee key security properties like *confidentiality*, *integrity*, *availability*, *accountability* and *non-repudiation* through different mechanisms such as authentication, authorization, trust management, etc.

In recent years, many researchers have attempted to address security issues in agent-based systems. They have analyzed security vulnerabilities and identified security requirements and challenges. In addition, possible security attacks have been studied, and applicable security techniques for attacks have been suggested. Various security models, middleware, and security services have also been proposed. Among a variety of security issues, the security for *mobile agent* systems has been the main focus of many researchers. A *mobile agent* is a particular type of agent with the ability to migrate from one host to another where it can resume its execution (Borselius 2002). An agent's mobility is not a mandatory characteristic but it has recently attracted attention because of its advantages. The most significant benefit of using mobile agents is that they can help to reduce network traffic and overcome network latencies (Chess et al. 1996). Furthermore, if we can solve the security problems related to mobile agents, then these solutions can be easily applied to solve the security problems of any type of agent-based system (Ghanea-Hercock and Gifford 2001). In addition to research focused on mobile agent security in general, some researchers have studied solutions for specific requirements of access control and trust management. Furthermore, many existing security solutions have been analyzed for their applicability to agent-based systems. Besides security considerations for agent-based systems, some researchers have also proposed using agents to provide security services; we do not discuss these in this paper.

In this paper, we present a survey of existing work on security requirements analysis and security solutions for agent-based systems, especially focused on access control and trust models. We also discuss their limitations and future challenges. The rest of the paper is organized as follows. Section 2 introduces research addressing the security requirements for multi-agent systems (MASs) by analyzing related threats and vulnerabilities. In Sect. 3, we present existing security solutions to defend against various threats. Section 4 surveys authentication techniques and access control models suitable for MASs, including trust management solutions for the purpose of distributed access control. Section 5 presents existing research related to trust and reputation in MASs. Finally, in Sect. 6, we conclude and present future research directions.

2 Security requirements in MAS

In this section, we first define and characterize MASs; then we describe security vulnerabilities based on agents' characteristics and then identify security requirements for MASs.

2.1 Multi-agent systems

There are many different definitions for agents and MASs. For our purpose, we use the following commonly accepted definition (Jennings et al. 1998): "*An agent is a software entity, situated in some environment, that is capable of flexible, autonomous action in order to meet its design objectives*". Although there is no universally accepted definition, most researchers agree that the agent's common characteristics of *situatedness*, *autonomy*, and *flexibility* (Franklin and Graesser 1996; Jansen 2000; Jennings et al. 1998) distinguish its paradigm from other software paradigms.

Situatedness means that an agent is aware of its specific condition based on sensory input it receives from its environment. *Autonomy* means that agents are able to control their own actions and internal states without the direct intervention of humans or other agents. *Flexibility* is the ability to adapt to changing situations and perform actions seamlessly towards achieving an agent's goals. Flexibility has three properties: *responsiveness*, *proactiveness*, and *social-ability*. *Responsiveness* means agents can perform actions that change the environment or give feedback as a response when they are aware of their environment. *Proactiveness* means that agents do not simply act in response to their environment; rather, they are able to exhibit goal-directed behavior (Franklin and Graesser 1996). *Social-ability* means that agents are able to interact with other agents and humans in order to solve their own problems or to help others. In addition, there are additional characteristics, such as *mobility*, *rationality*, *veracity*, and *benevolence*. A mobile agent, as mentioned earlier, is an agent that also has the characteristic of mobility, that is, the ability to migrate across networks and between different hosts (Greenberg et al. 1998). While agent systems can significantly benefit from such *mobility* of agents, the *mobility* of agents also introduces significant security concerns. Several researchers have concentrated on the security problems of mobile agent systems. We describe the problems and existing solutions in the following sections in detail.

Note that, as discussed earlier, a MAS is an agent-based system consisting of interacting agents (Ferber 1999). Generally, MASs have some complex problems to solve that require cooperation among agents. In addition, many MASs have no global control, and their data is usually decentralized. In order to make a MAS secure, we should deal with problems that occur across a system as well as problems within a single agent.

2.2 Vulnerabilities and security requirements

The agent paradigm has been shown to be a promising approach to develop intelligent, heterogeneous, and open systems, because of its features, such as *autonomy*, *flexibility*, and *cooperative problem solving behavior*. However, such characteristics make it more difficult to assure the security of MASs. Nowadays, many applications are being developed as MASs, even in security critical areas such as online-business, banking, and the medical service areas. Many researchers have recognized the security vulnerabilities of MASs and identified possible attacks.

In Borselius (2002), Mouratidis et al. (2003), authors present the security requirements of the MASs based on agents' characteristics. With regards to the *situatedness* characteristic, the verification of origin of information is a critical issue. If environmental information comes from an agent's host, then the security concerns may be minimal. If agents, however, get information from the Internet, we should check whether that information is trustworthy or not. Basically, an agent should know the source and the trustworthiness of information it uses. These concerns are issues related to authentication and integrity of information. The agent's *autonomy* might bring serious security problems because malicious

agents can be propagated without any request from other agents or humans (Mouratidis et al. 2003). Therefore, agents should be able to prevent or repair damages that are potentially inflicted by unauthorized accesses; and a MAS should be protected against malicious intrusions caused by other autonomous agents. With regards to *social-ability*, we should be able to assure secure communication among agents and also between agents and humans. In order to do so, we need to guarantee several security goals in MASs such as *confidentiality*, *integrity*, *availability*, *accountability* and *non-repudiation*. In addition, the agent's *mobility* may cause serious security problems. A host can be damaged by a malicious mobile agent. On the other hand, a malicious host may be able to compromise the security of mobile agents. Accordingly, we need security solutions for protecting both the hosts and mobile agents. To assure a mobile agent's security, we need to pay attention to the interactions with other malicious agents and users, as well as malicious hosts. Furthermore, cooperation among agents may cause more serious security problems. In order to achieve their goals, sometimes agents may need to access resources maintained and owned by others or to know about the internal status of other agents. If cooperation is allowed without appropriate authentication and authorization mechanisms in place, then serious security problems may arise.

In addition to identifying specific vulnerabilities related to agents' characteristics, other possible attacks against MASs have been studied in the literature. In Poslad et al. (2002), security attacks related to an abstract MAS architecture have been discussed. This work analyzes the Foundation for Intelligent Physical Agents (FIPA) architecture. The FIPA Abstract Architecture (FIPA 2002a) defines how agents can locate and communicate with each other by registering themselves and exchanging messages at an abstract level. To do this, a set of architectural elements and relationships among them have been defined. Among these elements, Poslad et al. focus on the model of service discovery, the message transport interoperability, the Agent Communication Language (ACL) representations, the content language, and the multiple directory services representation. They describe several threats associated with the name service, the directory service, and the communication service of the FIPA MAS architecture. The name service component may allow agents' fake identification in a message exchange or service request. While providing a directory service, Denial of Service (DoS) or unauthorized modifications are possible. During communication among entities in a MAS, key concerns are the eavesdropping or corruption of transmitted data. The authors analyze possible attacks with regard to the FIPA Abstract Architecture but do not provide solutions to those attacks.

Aside from the security attacks caused by architectural vulnerabilities, there has been significant research about security attacks related to mobile agent systems. In Greenberg et al. (1998), attacks to mobile agent systems have been classified into seven different types: damage, DoS, breach of privacy or theft, harassment, social engineering, event-triggered attacks and compound attacks. The event-triggered attack, called the *logic bomb*, is an attack triggered by an external event such as time, location, or the arrival of a specific person. The compound attack is composed of multiple attacks, possibly by cooperating agents or hosts. In Wang et al. (2005), attacks associated with the agent's mobility have been described. The authors argue that a mobile agent is vulnerable to attacks such as masquerading, DoS, eavesdropping, and alteration. On the other hand, a host may be vulnerable to masquerading, DoS, unauthorized access, and copy-and-reply. Moreover, the authors describe attacks to cooperating agents such as masquerading, DoS, unauthorized access, and repudiation.

3 Existing security solutions for MASs

As mentioned above, recently, many security vulnerabilities have been pointed out and attacks to MASs have been studied. To protect systems and recover from attacks, various security techniques have been proposed in the literature. In this section, we provide a summary of existing solutions to the security problems in MASs; these include specific security services, as well as general models and frameworks.

3.1 Overview of security solutions

We categorize suggested solutions in terms of the security requirements of MASs driven by the agent's characteristics.

Regarding *situatedness* characteristic of agents, we should be able to identify environmental information obtained from untrustworthy parties. To do so, the origin of information must be authenticated. To recover agents from the improper accesses from other autonomous agents, it is useful to maintain a log of interactions, or otherwise, to have a mediator agent to interact with all other agents and manage a system's interaction log (Jansen 2000). For secure sharing of the individual agent's resources and internal states during cooperation, an agent should allow only authenticated and authorized agents to access its resources or internal status. In order to do so, in Jennings et al. (1998), Roth (1998) proposes a method to make each message unique by associating it with the agent's identification and time stamp, and uses the ownership and usage relationship between agents and their resources/internal status.

Various security solutions for protecting mobile agents and their hosts have been proposed in the literature (Borselius 2002; Greenberg et al. 1998; Jansen 2000; Wang et al. 2005). To protect hosts, it is important to authenticate and authorize mobile agents. Towards this, in Wang et al. (2005), several security techniques have been introduced, which include: *sandboxing*, *safe code interpretation*, *signed code*, *authorization and attribute certificates*, *state appraisal*, *path history*, *proof-carrying code*, and *model-carrying code*. The *sandboxing* method is to isolate agents into a limited domain enforced by software (Borselius 2002; Jansen 2000). The *safe code interpretation technique* (Borselius 2002; Jansen 2000) is required when the mobile agent's code is interpreted. During execution of the interpreted code, it "*cures*" an unsafe command or simply ignores the command. The *signed code technique* (Jansen 2000) is to sign agents with the digital signatures of the creator of the agent, the agent's owner, and/or a trustworthy third party to make sure of the agent's authenticity/integrity. *State appraisal* (Farmer et al. 1996) aims to guarantee that an agent's status has not been modified. The idea of the *path history technique* (Roth 1998; Vigna 1997) is to make agents have a record of the prior hosts that they have previously visited. The *proof-carrying code* (Necula and Lee 1998) makes the author of an agent generate a proof that guarantees the safety of the agent code, and then the host can verify the agent using the proof transmitted with the agent. However, there is a drawback in that it is not easy to generate a formal proof. Unlike the *proof-carrying code*, which places the burden of security entirely on the code producer, *model-carrying code* (Sekar et al. 2001) distributes the burden to producers and consumers. A mobile agent host forms a *model* that captures the security-relevant behavior of code, rather than a proof, by using information accompanying the untrusted code. Code consumers are able to know the security needs of untrusted code more precisely. In addition, the *model-carrying code* technique enables the consumers to try out different security policies to untrusted code before the execution, and then select one policy.

Several techniques for protecting mobile agents have also been proposed in the literature, which include: *contractual agreements*, *using trusted hardware or nodes*, *cooperating*

agents, *execution tracing*, *sliding encryption*, *environmental key generation*, *computing with encrypted functions*, *obfuscated code*, and *undetachable signatures* (Borselius 2002). The *contractual agreement method* makes host operators guarantee the security of the host by establishing and using contractual agreements. Using trusted hardware and trusted hosts is a simple and effective solution. Sometimes, the single-hop technique is used with the trusted node solution in some applications. For fault-tolerant MASs, it is useful to replicate significant information or code in cooperating agents. The *execution tracing technique* (Vigna 1997) keeps a record of the agent's executions in previous host platforms. Its objective is to detect an unauthorized modification in a mobile agent. However, the huge size of the log becomes a drawback of this technique. In case of the exchanges of small messages, the *sliding encryption technique* (Young and Yung 1997) can help to reduce the overhead to guarantee confidentiality. The *environmental key generation* method (Riordan and Schneier 1998) has been suggested as one of the security techniques to protect an agent's privacy. Here, the agent's encrypted data can only be decrypted under specific predefined environmental conditions. In order to ensure secure execution, the agent's owner needs to provide an encrypted agent function. The *obfuscated code technique* (Hohl 1998), referred to as *blackbox security*, tries to keep the secrecy of an agent by scrambling the agent's codes. However, this technique has some drawbacks in that there is no universal algorithm for blackbox security and it is effective only for a short period of time.

We categorize the aforementioned vulnerabilities, security issues, and security techniques according to the characteristics of MASs as shown in Table 1.

3.2 Comprehensive solutions: models, middlewares, and systems

In addition to various techniques aimed at specific needs, some comprehensive solutions have been carried out. In this section, we introduce some standards, security models, middlewares, and secure MASs.

As an effort to standardize secure MASs, FIPA has tried to capture security concerns into their specification. Some security requirements, including secure channels and authentication, have been reflected in the FIPA Abstract Architecture specification (FIPA 2002a). To provide specific security services to FIPA-based MASs, FIPA has specified security for the message transport service (FIPA 2002b) and agent management (FIPA 2004). The FIPA security technical committee has published the white paper (FIPA 2002c) of FIPA MASs Security in 2002. They have reviewed past activities to guarantee security of FIPA-based MASs and suggest some security issues that should be considered in such systems.

So far, there is no standard security model for MASs although various security models have been proposed in the literature. Poslad et al. (2002) propose the *asset security model*. In this model, security is defined as a set of safeguards that help protect the assets. They define the communication service, the name service, and the directory service as the core MAS assets, and then describe threats and safeguards for each. Mouratidis et al. (2003) propose security concepts which enable *Tropos* methodology to model security concerns throughout the entire MAS development process. *Tropos* (Bresciani et al. 2004a) has been proposed as an agent-oriented software engineering methodology, but it does not consider security. Hence, Bresciani et al. try to add security to *Tropos* methodology by employing various security concepts, such as *security constraint and security dependency*; *security entities* such as *secure goal* and *secure task*; and a *security reference diagram*. As an initial step, they introduce an algorithm that identifies and break security bottlenecks to reduce the complexity and criticality of MASs (Bresciani et al. 2004b). They have extended *Tropos* to propose the *secure Tropos* model (Mouratidis 2007). They have also suggested a way to enhance *Tropos* security

Table 1 Security vulnerabilities and techniques for MASs

Characteristics of MASs	Vulnerability	Security requirements	Useful security solutions
Situatedness	Origin of information	Identification and authentication of information origin	Authentication—PKI method, X.509 certificates, digital signature
Autonomy	Access control	Authorization mechanism	Access control models
			Accountability through use of interaction log of an individual agent or system interaction log of an mediator agent
Social-ability	Unsecure communication among agents and between agent and host	Communication security	Authentication—PKI method, X.509 certificates, Digital signature
			Confidentiality—encryption methods
Mobility	Malicious agents	Host protection	Authentication—signed code
			Confidentiality—sandboxing, safe code interpretation
			Integrity—path history
			Privacy—sandboxing, safe code interpretation
			Authorization—sandboxing, safe code interpretation, signed code, state appraisal, path history, proof carrying code
	Malicious host	Agent protection	Authentication—undetachable signatures
			Confidentiality—sliding encryption
			Integrity—trusted hardware or nodes, cooperating agents, execution tracing, sliding encryption, computing with encrypted functions, obfuscated code
			Privacy—trusted hardware or nodes, environmental key generation, obfuscated code
			Authorization—contractual agreements
			Fault tolerance—replication of information or code in cooperating agents
Cooperation	Access to resources and internal status of other agents	Identification, authentication and authorization of cooperating agents	Authentication—PKI method, X.509 certificates, digital signature
		Secure sharing of resources	Authorization—access control models (DAC, MAC, RBAC, etc)

by integrating three secure processes (Mouratidis and Giorgini 2009). A *selection* process is first added to choose a system's architectural style such as the client/server style or the mobile agent style by using the satisfiability calculation. After this selection, a process to *transform* the requirements to a design is performed, and then a *test* process validates an implemented security solution based on various attack scenarios.

Beydoun et al. (2009) propose a meta-model for MASs by adding security concerns into *FAML* (Beydoun et al. 2006), the *FAME* Agent-oriented Modeling Language. *FAML* is a general meta-model to describe features of MASs, but it does not consider security requirements. In that work, the authors have described some vulnerabilities caused by the MAS's characteristics, such as *cooperation*, *autonomy*, and *mobility*. To address such vulnerabilities, they extend the original *FAML* by inserting security techniques such as an interaction history log.

van't Noordende et al. (2004) suggest a security architecture for the *Mansion*-based mobile agent systems. *Mansion* is a multilayered middleware system designed to support large-scale mobile agent systems. *Mansion* provides security using authentication and authorization services. The *signed code technique* is used to authenticate mobile agents, and all objects have an access control list (ACL) indicating the agents authorized to access them. To control information flow, it uses *confined room*s. If a mobile agent enters a *confined room*, then its interaction with the outside is cut off. For mobile agent protection, *Mansion* provides secure *AMS* (Agent Management Service), such as location look-up service and auditor and notary processes. In addition, it maintains the audit trails, which help to protect agents from tampering with an agent's persistent state during multi-hop travel. It also provides agents the *handoff protocol* to guarantee secure migration of mobile agents.

Vuong and Fu (2001) propose a secure mobile agent system, called the Secure Actigen System (SAS). *SAS* provides several security services for hosts and mobile agents. For host protection, it offers *digital certificate* service and *digital signature* service based on SHA1-DSA algorithm in order to authenticate mobile agents. Moreover, a host must check its own security policies before it provides services to a mobile agent. To protect an agent's integrity, it uses the *syntactic integrity check* mechanism and the *append-only data log* approach.

Recently, security mechanisms employed by agent platforms has been evaluated by several researchers. Fischmeister et al. (2001) has provided a test result on three Java-based mobile agent platforms: *Aglets*, *Jumping Beans*, and *Grasshopper*. They focus on the attacks that can be launched by a mobile agent against the authorization mechanisms of these platforms. They report several vulnerabilities on the listed agent platforms. For example, Aglets (2002) allows unauthorized modification of security policies and the platforms to reveal a portion of code and information about users' identification. Jumping Beans (2006) can be disabled or shut down by graphic user interface attacks and the runtime system call attacks. In *Grasshopper* (Bäumer and Magedanz 1999), they show that it is possible to modify the system's properties without any approval and to bypass its authorization system. Bürkle et al. (2009) evaluate two different agent platforms: JADE (2007) and the *Secure Mobile Agents* (SeMoA 2007). They demonstrate attacks that can be launched by malicious mobile agents and use them to test the security of the two platforms. Towards this, they have built a testing system consisting of various hardware having different operating systems and then have successfully launched several attacks such as DoS, masquerading, eavesdropping, spamming, unauthorized access to the host's data, etc. The tests performed on *JADE* reveal a serious loophole in the security mechanisms and also show weaknesses against some attacks such as recursive cloning, non-blocking behaviors, and spamming. *SeMoA* has been shown to be successful in preventing unauthorized access and attacks to other agents. However, it has been shown to be vulnerable to DoS attacks using the endless loop execution or memory overload. As the

attacks become numerous and more complicated, the necessity of an attack-resistant agent platform increases significantly. Through such evaluation results, we can identify the extent of security that has been provided in the current platforms.

4 Access control approaches

As discussed in the previous section, many security techniques and solutions have been proposed to guarantee the security of MASs. Among them, we focus on two major security solutions in this paper: access control and trust approaches. In this section, we present a survey of existing literature on access control and trust management.

Securing access to systems in general involves two main steps: *authentication* that establishes the true identity of a subject, and access control or *authorization* that defines which subject has what type of access privileges to which resources. We first introduce authentication mechanisms suitable for and have been frequently suggested to be used in MASs in Sect. 4.1. Then, we present an overview of the generic access control models and the ones that are proposed specifically for MASs in Sect. 4.2. In Sect. 4.3, we introduce research on trust management which supports distributed access control.

4.1 Authentication

Authentication establishes the identity of one party to the other and is prerequisite for access control mechanisms (Sandhu and Samarati 1996).

One of the main challenges to support authentication and access control in MASs is to enforce them in a distributed manner. Researchers have proposed the use of public-key infrastructures (PKI) for this purpose. PKI schemes are classified as trust management protocols, which specifically deal with authentication and access control in a distributed environment.

PKIs are the mainstream trust management systems, although there exist other successful protocols and frameworks: e.g., Pretty Good Privacy (PGP) that provides communication privacy and distributed authentication, and Kerberos, which is a distributed authentication mechanism. Several research efforts in the multi-agent area have adopted the ideas of PKI, mainly the IETF X.509 standard and Simple Public Key Infrastructure/Simple Distributed Security Infrastructure (SPKI/SDSI) or have directly used them to provide protection in MASs.

X.509 is an IETF standard for distributed authentication, which is largely employed today in online business applications. In this protocol, a certificate authority issues a certificate that bind a public key to the unique ID of a principal. The authenticity of a certificate can be verified by the certificate authority, which itself relies on (implicitly trusted) root certificates. The important components of a X.509 certificate include the names of the issuer and subject, the subject's public key information, and the validity period of the certificate.

SPKI/SDSI is another standard developed with the primary intention of avoiding complexities of X.509; it provides distributed authorization and bases the principals on public keys instead of identities. Public keys can be more easily assured to be globally unique in comparison to identities. Since there is no need to verify an identity, there are no central certificate authorities in the sense that exists for X.509. Local names can be bound to public keys using name certificates only for issuer's reference. Name certificates are composed of the public key of the issuer and subject, an identifier, a describing term, and a validity period assigned by the issuer. Local names are used only for human convenience or for grouping multiple principals. SPKI/SDSI is, moreover, capable of managing authorizations by granting

authorization certificates signed by the issuers. The receiver can further delegate such authorizations if the issuer allows it in the original certificate. By the propagation of delegation, the authorization can be managed in a distributed manner. The authorization certificate is composed of the public keys of the issuer and the subject, a permission specification, a delegation bit, and a validation period. Upon receiving an access request for a resource, the provider principal can easily verify the authorization certificate (or chain of certificates).

4.2 Access control

Various types of access control requirements exist for different types of systems and applications. There exist several access control models and policy specification frameworks to address such needs. An access control policy defines the rules according to which the access is regulated (Samarati and De Capitani di Vimercati 2001). Traditionally, access control policies are grouped into *Discretionary Access Control* (DAC) and *Mandatory Access Control* (MAC) policies. In DAC, authorization is controlled at the discretion of users, who are the controller or owner of some resources, usually based on the identity of the requester. Access Control Matrix is the conceptual DAC model proposed by Lampson (1974). In an open environment where entities may be strangers to each other, attributes other than the identifier can be used as the basis for authorization; such an approach is used in attribute-based access control models. Contrary to DAC, MAC policies are based on mandated regulations determined by the system. For instance, the Bell LaPadula model (Bell and La Padula 1976) prevents unauthorized information flow between higher and lower security classes based on mandatory rules: the *no read-up* and *no write-down* rules. Role-Based Access Control (RBAC) has been proposed as an alternative to DAC and MAC, where a role is defined as a job function within the organization that describes the authority and responsibility conferred on a user assigned to the role (Sandhu et al. 1996). Although roles in RBAC are primarily defined for organizations, the generality of role concept makes it applicable to many systems including MASs. In RBAC, permissions are assigned to roles and users can exercise permissions through activation of assigned roles. Other access control models have been proposed in recent years to support more expressive and flexible policies. Context-aware access control models (Covington et al. 2002; Wilikens et al. 2002; Zhang and Parashar 2004) let the policy adapt to the changing context by specifying contextual conditions. Moreover, some access control models deal more specifically with particular context types such as time (Bertino et al. 2001; Joshi et al. 2005), and location (Damiani et al. 2007; Chandran and Joshi 2005).

Cremonini et al. (2000) propose an infrastructure, called TuCSoN, for coordinating support in agent-based systems. In TuCSoN, interactions among agents are mediated through tuple centers embedded in nodes. There are gateways for nodes that are structured hierarchically, and each gateway is responsible for protection of its domain. Access to tuple centers is controlled by using traditional access control matrix model enforced by gateways. To enable the gateway to control its domain, the authors expand access matrix dimensions with a component indicating nodes or sub-gateways that are included in a domain through a gateway interface; therefore, external accesses to a domain can be handled by the top gateway. In order to make such enforcement possible, access control privileges need to be delegated from lower gateways to the upper gateways in the hierarchy. Later, Omicini et al. (2005) have explored the integration of RBAC into the TuCSoN infrastructure. In order to control the coordination protocol, the authors define a prolog-like role policy definition language. The policies can specify the authorized actions considering the current state of roles and conditions, while determining the next state.

Boella and van der Torre (2004) distinguish between authorization and permission in a community policy. In their work, a community is a distributed MAS operating in a P2P environment. An authorization is enforced by the community authorization service based on the community's policy, while the actual permission is granted by the members.

From the perspective of employing access control policy models in the area of MASs, with the exception of the above-mentioned work, most of the proposed frameworks apply very basic policies like identity-based ACLs by implementing access matrix models. For instance, Such et al. (2009) propose using Linux's native access control model, which employs users, groups, and access control lists in a MAS platform where a Linux process is regarded as an agent. Wen and Mizoguchi (2000) base their authorization policy on roles but they do not consider adopting the notions from RBAC models; for example, how roles are assigned to users or how permissions are assigned to roles is not discussed.

Key challenges exist for access control in MASs that need to be further investigated. Agents in a MAS collaborate to achieve the proposed goals of the system. It is important to ensure that any collaboration among agents does not undermine those goals, which can be considered as a security threat. In order to maintain this requirement, collaborations among agents need to be controlled and secured. This aspect is not covered by the common practices of access control models that consider protection of only information objects. It is important that to develop proper access control models that also capture the authorized/unauthorized interactions among agents. Jung et al. (2008) have proposed preliminary work that captures authorization of interactions among roles in an extension to RBAC model for MASs.

Another research direction is to investigate distributed autonomous enforcement of access control policies in practical MASs. Most of the work in MAS access control is related to the requirement of distributed authorization (Wen and Mizoguchi 2000; Wangham et al. 2004; Hu and Tang 2003). However, the multi-domain nature of such systems is neglected. In a MAS, authorizations for resources that are controlled by agents, or even the agents themselves as community resources, are typically enforced by the agents. But since agents are autonomous entities, they may simply choose not to follow the community policy. Therefore, inconsistencies may occur between policy and actual enforcement. We believe that access control policies in MASs should be augmented with concepts related to agents, such as obligations and responsibilities of agents.

4.3 Trust management: distributed access control

MASs consist of multiple autonomous agents with limited knowledge of each other. To enable protection and secure operation or interaction in such systems, trust relationships must be established between them. Trust management has been previously investigated for building trust and controlling accesses in distributed systems. Therefore, it naturally attracts many researchers in MAS area as a proper approach for distributed authentication and authorization.

As distinguished from human PKI, which is used for authenticating human users, Hu and Tang (2003) propose an agent PKI that can set up a trust path between service providers and receiver agents and do certificate binding between humans and agents. The agent certificate authorities (CAs) are structured as trees consisting of root, general, and local CAs. They describe a protocol for agent identity certificate application, issuance, revocation, and verification. Attribute certificates are also issued for humans but can be used by their corresponding agents. The binding of a human to an agent is done by the human signing the public key of the agent in its identity certificate. The format of attribute certificates in this framework is similar to SPKI/SDSI.

Wen and Mizoguchi (2000) propose an authorization-based trust model (ABTM) which is based on SPKI/SDSI certificates, but the authorization is performed through authorization servers, making it a semi-decentralized scheme. Resource provider agents delegate the permission for authorization of their resources to the authorization server. Thus, the authorization server can employ coherent security policies to authorize the requests; here, they provide a role-based policy. For requesting a service, a user delegates its role to an executive agent. The agent would place a request with the authorization server providing its role certificate. Based on the policy, the authorization server grants the authorization certificate to the executive agent, which can be presented to the resource provider agent to access the resource. That way, they separate the authorization decision-making, which is performed by authorization servers, and enforcement, which is performed by resource providers.

Several researchers propose frameworks for specific applications. For instance, Wangham et al. (2004) propose a mobile agent security scheme based on SPKI/SDSI, specific to the application of searching and selecting partners in the formation of Virtual Enterprises and negotiating between partners. Few have basically redefined the challenges and solutions in SPKI/SDSI (Poggi et al. 2004), or proposed certificate-based schemes such as Novák et al. (2003).

Formal logic-based models for distributed security provide theoretical approaches to capture the state of the entities of the systems and possible inferences in a distributed system (Lampson et al. 1992; Abadi et al. 1992). Accordingly, some researchers have proposed new logic or built on existing logic and the trust semantics in agent systems. Liau (2003) propose logic that models and relates belief, information exchange, and trust in MASs. The author provides doxastic logic with modalities for representing trusting attitudes and information transmission actions between agents. The logic supports belief, trust, and information acquisition operators to model an agent's belief, trust in another agent about something, and how it acquires information from another agent. Different properties of trust and information acquisition are discussed and shown to be useful in formulating the properties of MASs. Such logic can be used to verify the semantics of an agent system by linking system states with agent mental states. Berkovits et al. (1998) propose trust relations in mobile agent environments based on Lampson et al.'s logic for authentication (Lampson et al. 1992). They focus on three goals: certification of executing a mobile agent by a host, providing necessary privileges for agents to carry out their tasks, and ensuring the non-malicious state of an agent due to alterations imposed by the host.

5 Trust and reputation

In addition to access control, trust and reputation are also key issues to provide secure and trustworthy services for modern systems, especially e-business and other web systems (Sabater and Sierra 2005). So far, a number of models and systems for trust and reputation have been proposed in the literature. In this section, we present an analysis surveying previous work on trust and reputation in MASs. In Sect. 5.1, we first introduce the concept of trust and reputation, and then summarize existing models and systems in Sect. 5.2. Finally, in Sect. 5.3, we present comparisons of existing work and discuss the results.

5.1 Definition

Trust and reputation systems have been recognized as key factors for successful adoption of electronic commerce (Resnick et al. 2000). These systems have been frequently employed

by intelligent systems including MASs as a mechanism to search for trustworthy interaction partners and decide whether or not to honor contracts. Before exploring existing work on trust and reputation, we first define trust in the context of this paper. Trust is defined in a variety of ways. Grandison and Sloman (2000) examine the various definitions of trust and then provide a working definition of trust for internet applications: "Trust is the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context". However, we adopt the definition of Mui et al. (2002), because their definition can be considered as 'reputation-based' trust, and we consider trust in MASs as a concept which strongly relates to reputation: "Trust is a subjective expectation an agent has about another's future behavior based on the history of their encounters".

Reputation is a subject's opinion or view on objects. Reputation of an agent in MASs is an assessment of the agent, based on the history of interactions with it or observations of it by another agent. Such reputation values can be directly evaluated by the agent itself or reported by others. In MASs, reputation has been mostly used to establish trust between agents (Ramchurn et al. 2004). The record of past interactions and an evaluation of an agent's performance are a basis for building the reputation of the agent, and such information is used to expect the agent's future behaviors.

Trust and reputation have played a role in facilitating interaction among agents. It is hard for agents to decide whether other agents are malicious or not. In this circumstance, it is really useful to know an agent's reputation in order to find a safe partner for interaction. Accordingly, trust mechanisms have taken an important role to ensure trusted interactions in MASs.

Several survey papers have been published related to trust and reputation in MASs. Ramchurn et al. (2004) broadly discuss the trust issues in open MASs. By analyzing the characteristics of MASs, authors point out some trust issues. Sabater and Sierra (2005) review various computational trust and reputation models using seven comparison factors; the *general model*, the *information source*, *visibility*, the *model's granularity*, agent *behavioral assumptions*, and the *behavior model*. Arts and Gil (2007) discuss existing work on trust in the semantic web. They categorize trust into four groups by how trust is established: *policy-based trust*, *reputation-based trust*, *general model of trust*, and *trust in information resources*. However, these surveys do not focus on MASs, so they fail to provide comprehensive analysis on the practical use of trust/reputation models and systems in MASs.

5.2 Models and systems

In this section, we introduce some existing work on trust and reputation models and systems that are utilized in MASs.

Zacharia and Maes (2000) propose two reputation models, SPORAS and HISOTS. These models are extensions of online reputation models, such as those used in eBay and Amazon auctions, in that they investigate a new rating aggregation method. SPORAS is a reputation mechanism for loosely-connected communities that share the same interest. It is a centralized model with more sophisticated characteristics than other online reputation models that simply decide the reputation values by collecting user's opinions. In this model, the rating values are aggregated by considering only the two most recent users (agents). For predicting a user's reputation, SPORAS provides a way to measure its reliability based on the standard deviation of reputation values. On the other hand, HISTOS is a more personalized reputation system than SPORAS, so it is used as a reputation mechanism for highly connected communities. Unlike SPORAS, it calculates the reputation of a user by considering who makes the query and how he/she has rated other users in the online community.

While SPORAS and HISTOS are centralized models, ReGreT is a decentralized trust and reputation system oriented towards complex and midsize e-commerce environments where social relations among individuals play an important role (Sabater and Sierra 2001, 2002). It evaluates trust by considering social relations and ontological considerations as well as an agent's direct perception of the reliability of the target agent. The ReGreT reputation model includes three specialized reputation types which are differentiated by the information source: *witness reputation*, *neighborhood reputation*, and *system reputation*. Each agent rates its partner's performance after every interaction and records the ratings in a local database. An agent can use the stored ratings information to evaluate another agent's trust by querying its local database. An agent derives such a trust value by calculating the weighted mean of all ratings, called *direct trust*. In this model, each rating is weighed according to its recent activities. Like SPORAS, ReGreT also provides a measurement of reliability for each trust value to show its predictive power. Unlike previous models, FIRE integrates a number of information sources in open systems to produce a comprehensive assessment of an agent's performance (Huynh et al. 2004, 2006a,b). It categorizes information sources into four types: *interaction trust*, *role-based reputation*, *witness reputation*, and *certified reputation*. Then, it integrates them to calculate a more precise value of trust and reputation. First, the interaction trust is built from direct interaction experience between two agents. Second, the role-based reputation is modeled by the role-based relationships between two agents. To assign the role-based trust values, some rules need to be defined. Third, the witness reputation of target agent is built based on the observation of other agents about the target agent's behavior. To find some witnesses, FIRE adopts the referral system presented in Huynh et al. (2006b). Finally, the certified reputation of target agent consists of a number of certified references about the target's behavior on particular tasks provided by third-party agents (Huynh et al. 2006a). The certified reputation is defined to overcome the shortcomings of the interaction trust and the witness reputation.

Abdul-Rahman and Hailes (Abdul-Rahman 2005; Abdul-Rahman and Hailes 2000) propose an Ntropi model in which trust and experience are differentiated by levels. For instance, levels for the trust can be 'Very Trustworthy', 'Trustworthy', 'Moderate', 'Untrustworthy', and 'Very Untrustworthy'. This model uses not only direct trust and reputation, but also the recommending party's trust to assess witness credibility in computing a final trust degree for a target. Ntropi models two types of trust: *situational trust* and *basic trust*. This model represents trust by classifying it into five levels, or strata. Its disadvantage is that the trust values are too coarse-grained, thereby, losing both sensitivity and accuracy. Although comparisons are easier, the update of values is more complex than using continuous values.

Multi-Dimensional Trust (MDT) is proposed for multi-dimensional trust (Griffiths 2005). Agents model the trustworthiness of others according to various criteria, such as cost, timeliness, or success, depending on which criteria the agent considers important. In this model, agents use their own direct experience of interactions with others. To improve the model, MDT-R has been proposed with the concept of recommendation (Lim Choi Keung and Griffiths 2008). MDT-R stratifies trust into several levels for ease of comparison. In this model, the summaries of relevant past interactions are shared instead of the explicit values for trust, since sharing of information among agents often suffers from the differences in subjective interpretation of each agent. A recent work extends the MDT-R's mechanism to obtain recommendations by including indirect recommendations also.

This mainstream approach of using social networks in trust and reputation models is attracting many researchers. Yu and Singh tackle the problem of retrieving ratings from a social network through the use of referrals, pointing to other sources of information similar to web links (Yu and Singh 2002, 2003). They propose a method of representing a social

network and gathering information through the network. They show how agents can explore a network and use referrals gathered to build up a model of a social network. Schillo et al. (2000) propose to enhance the representation of existing social networks by annotating the particular characteristics of network nodes. Each node holds two values, *trust value* and the *degree of altruism*. In this model, the trust values describe the degree of an agent(node)'s honesty. Both values are used to deduce the trustworthiness of witnesses queried at the time of calculating the reputation of potential interaction partners. Pujol et al. (2002) propose the Node Ranking algorithm for creating a ranking of an agent's reputation in a community by means of a corresponding social network. The main idea is that each node has authority and a part of this authority is propagated to the out-nodes via out-edges. ReGreT, as mentioned above, also exploits social network technologies (Sabater and Sierra 2001). By using the social dimension in its reputation system, agents start to take into account social relations and such social information becomes important to build a good reputation as well as to know other agents' reputation.

Agents can change their behavior unpredictably, hence the trust and reputation models need to be aware of such changes and be able to adjust corresponding values dynamically. This is a reason why establishing a dynamic models of trust and reputation is one of the most challenging requirements today. Some recent research has proposed approaches to address such requirements. For example, Li et al. (2008) propose a dynamic trust model for MASs. This integrated trust model, similar to ReGreT and FIRE, is used to calculate the trust and reputation values of agents, such as recent trust, historical trust, expected trust, and confidence factor. As a major contribution, this model introduces a filtering method that removes inaccurate values.

5.3 Comparison and discussion

In this section, we compare existing models described above. We first investigate several comparison factors that can differentiate MASs from other systems and then compare and categorize existing work.

5.3.1 Comparison factors

Information sources. Information sources can be considered as a main factor in comparing trust and reputation models/systems. Each of them gathers information from the environment or agents and then calculates the trust and reputation values. Usually, three kinds of information are used to do this, which are: *direct experiences*, *witness information*, and the *relationship* among agents.

Obviously, direct experience is the most relevant and reliable information source, so traditional systems mainly depend on this information. Witness information is knowledge gathered from other agents in the same society. If you want to use witness information, you need to verify it first, because it is relatively less reliable than the direct experience.

The information about relationships among agents indicates a social relationship (Sabater and Sierra 2005) or a role-based relationship (Huynh et al. 2006a). When two agents interact with each other, the trust and reputation value can be more reliable by considering their relationship.

Reliability measurement of trust and reputation. This factor shows whether a model provides a method to measure the reliability of trust and reputation values. For example, let's assume the following situation: agent a_1 usually trusts agent a_2, but sometimes, the trust value

is not reliable if a_1 does not have sufficient experience. To evaluate the reliability of trust or reputation, some models employ the certified agents or define a formula for calculating its reliability.

Distributed reputation. MASs are essentially distributed systems, therefore, their reputation systems should be able to support a distributed mechanism to generate and maintain the trust and reputation values. From this perspective, it is important to classify models by their ability in distributed reputation. Most recent models are the distributed models, but some of them are not fully distributed. Besides, the centralized reputation methods are still used in online e-commerce systems.

Agent behavior assumption. Some existing models assume that agent's behavior is used to build trust and reputation. To classify the level of assumptions regarding agent behavior, we adopt the three levels introduced in Sabater and Sierra (2005):

- Level 0—A model relies on a large number of agents who offer honest ratings to counteract the potential effect of the ratings provided by malicious agents. It does not consider an agent's malicious behavior on rating.
- Level 1—A model assumes that agents can hide specific information or provide biased information but they never lie. It means that agents are honest in exchanging information.
- Level 2—A model has specific mechanisms to deal with liars.

Filtering inaccurate trust. Agents do not act in a static way, but they perform their actions depending on their belief or social commitment. However, sometimes their belief can be wrong even though they are not liars; that is, they might have inaccurate information and the corresponding trust value can, consequently, be inaccurate. Therefore, the trust and reputation systems need to catch such inaccurate values and then fix them to assure trusted interactions.

Composition. Wang and Singh (2006a,b, 2007) introduce the combination of trust and reputation. It is clear that trust cannot be trivially propagated, but in some cases, it is better to compose different trust values. For example, a_1 may trust a_2 who trusts a_3, but a_1 may not trust a_3. In this case, to calculate a_1's trust of a_3, a_1's and a_2's trust of each other must be considered.

5.3.2 Discussion

In Table 2, we list various approaches described in this paper and compare them on the basis of six factors. With regards to distributed reputation, most existing work supports this factor. With regards to reliability, ReGreT and FIRE build the trust and reputation values from a variety of information sources by integrating each value from a source, and hence, the trust values become more reliable. However, they have to cover the high cost and overhead to maintain all information sources and relationships among agents. Furthermore, there is a scalability issue with regards to maintaining witnesses.

Some recent work has proposed methods to evaluate and measure the reliability of trust values. The certified reputation model in FIRE evaluates the combined trust values, and then certifies the value or fixes it by increasing or decreasing it. The confidence factor introduced in Li's work plays a similar role in the certified model of FIRE, but it requires a centralized server or some trustworthy agents to do such measurements. Due to this limitation, it is hard to use this model in dynamic MASs.

Table 2 Comparison between various trust and reputation models/systems

	IS	RM	DR	BA	FI	TC
SPORAS	WI	✓	×	Level 0	×	×
HISTOS	DE+WI	×	×	Level 0	×	×
REGRET	DE+WI+RI	✓	✓	Level 2	✓	×
FIRE	DE+WI+RI	✓	✓	Level 1	×	×
Ntropi	DE	✓	✓	Level 0	×	×
MDT	DE, WI	✓	✓	Level 0	×	×
Li et al.	DE	✓	✓	Level 2	✓	×
Wang and Singh	DE+RI	×	✓	Level 2	✓	✓
Schillo et al.	DE, WI	×	✓	Level 1	×	×
Pujol et al.	N/A	×	×	Level 2	×	×
Yu and Singh	DE, WI	×	✓	Level 1	×	✓

IS information sources, *DE* direct experience, *WI* witness information, *RI* relationship information, *RM* reliability measure, *DR* distributed reputation, *BA* agent behavior assumptions, *FI* filtering inaccurate value, *TC* trust composition

Since agents are autonomous entities and there is no central authority, agents' opinions and behaviors cannot be anticipated. In this sense, the assumptions for agent behaviors described above might not be practical. ReGreT, and Lie et al.'s and Want and Singh's methods employ level-2 assumptions, but none provide a complete solution. In fact, this problem is a highly complicated one requiring artificial intelligence techniques to overcome it.

6 Security challenges and conclusion

In this section, we summarize the key security challenges on MASs. Then we conclude this paper.

6.1 Security challenges

Although existing research provides useful solutions to guarantee the security in MASs, unsolved problems still remain. Furthermore, new challenges arise as new technologies are developed. In this section, we discuss the future challenges on which we should focus.

6.1.1 Authorized collaboration

Collaboration is one effective means to achieve agents' goals in MASs. However, collaboration with malicious agents may make agents deviate from achieving their goals. For instance, a malicious agent may ask other agents to provide services that it really doesn't need with the intent to make some community resources unavailable. To overcome such undesirable interactions, a solution to the problem of how to control authorized collaborations is required. A proper access control model is needed to enable secure cooperation among agents.

6.1.2 Autonomous access control enforcement

In scalable MASs, access control must be enforced in a distributed manner. An agent, either as a controller of environmental objects or as an interaction partner of other agents, can

be an actual enforcer of authorization policy in such systems. However, agents are autonomous entities so they may simply choose not to follow authorization policies previously agreed to. To cope with such enforcement problems, some concepts such as commitment and responsibilities of agents beyond authorization are required.

6.1.3 Trust composition

Wang and Singh (2007) have introduced a method called trust composition, which combines several trust values from different agents. Let's look at an example. Agent a_1 trusts agent a_2, and a_2 trusts agent a_3. We cannot say that a_1 trusts a_3, because trust relationship is not necessarily transitive. Most existing approaches do not contribute to this issue but a few researchers have tried various techniques to combine trust values. Unfortunately, there is lack of practical or comprehensive solutions.

However, the importance of trust composition is obvious when considering the organization of agent groups. In a group, agents generally interact with each other to achieve their common goals. To recruit the best agents for a group, each agent should be a trustworthy partner with others in the group. However, this is definitely a complicated problem since it is hard for an agent to get complete knowledge about other agents. In such circumstances, the trust composition can play a critical role for determining the trust and reputation values for unknown agents.

6.1.4 Filtering inaccurate trust

Sometimes, agents' trust value might be inaccurate; hence, the trust and reputation systems should be able to filter out the inaccurate reports to maintain its trustworthiness. Li et al. (2008) have addressed this issue by using past interactions and witness information, but the method to find inaccurate reports and to recalculate the value is still naive and not very practical. If we eliminate the inaccuracy of trust and reputation values, then we can guarantee more reliable interactions in MASs.

6.1.5 Trust and reputation customization (personalization)

It is not easy to select the most suitable trust/reputation model for each application and adapt it for the particular requirements of the application, since trust and reputation are very subjective. Consequently, it is difficult for users cannot to easily adapt a model to match their needs without resorting to re-programming it. A work called PTF (Personalized Trust Framework) proposed by Huynh (2009) can be considered as a first work towards this direction.

6.2 Conclusion

Agent technology can benefit many application systems with advanced characteristics such as autonomy, intelligence, and dynamic and cooperative problem solving abilities. Many MASs have been developed for different application areas. However, ensuring security of such agent based environments is very critical. To address security requirements, many researchers have proposed various types of security approaches for MASs. In this paper, we have reviewed and categorized the existing related work on security issues. Among the many security issues, we have focused in more detail on access control and trust/reputation issues. Despite many efforts, several problems still remain and new challenges are continuously being identified as new technologies are developed.

Acknowledgments This work has been supported by the US National Science Foundation award IIS-0545912.

References

Abadi M, Burrows M, Lampson B, Plotkin G (1992) A calculus for access control in distributed systems. In: Feigenbaum J (ed) CRYPTO 1991, LNCS 576. Springer, Berlin, pp 1–23

Abdul-Rahman A (2005) A framework for decentralised trust reasoning. Ph.D. Thesis, Department of Computer Science, University College London, UK

Abdul-Rahman A, Hailes S (2000) Supporting trust in virtual communities. In: Proceedings of the 33rd international conference on system sciences. IEEE Computer Society, p 6007

Aglets (2002) Aglets. http://www.trl.ibm.com/aglets/. Accessed 1 Apr 2011

Arts D, Gil Y (2007) A survey of trust in computer science and the semantic web. J Web Semant 5:58–71

Bäumer C, Magedanz T (1999) Grasshopper—a mobile agent platform for active telecommunication networks. In: Albayrak S (ed) Intelligent agents for telecommunication applications, LNCS 1699. Springer, Berlin, pp 19–32

Bell DE, La Padula L (1976) Secure computer system: unified exposition and multics interpretation. ESD-TR-75-306 ESA/AFSC, The MITRE Corporation

Berkovits S, Guttman JD, Swarup V (1998) Authentication for mobile agents. In: Vigna G (ed) Mobile agents and security, LNCS 1419. Springer, Berlin, pp 114–136

Bertino E, Bonatti PA, Ferrari E (2001) TRBAC: a temporal role-based access control model. ACM Trans Inf Syst Security 4(3):191–233

Beydoun G, Gonzalez-Perez C, Henderson-Sellers B, Low G (2006) Developing and evaluating a generic metamodel for MAS work products. In: Garcia A (ed) et al Software engineering for multi-agent systems IV, LNCS 3914. Springer, Berlin, pp 126–142

Beydoun G, Low G, Mouratidis H, Henderson-Sellers B (2009) A security-aware metamodel for multi-agent systems (MAS). Inf Softw Technol 51:832–845

Boella G, van der Torre LWN (2004) Permission and authorization in policies for virtual communities of agents. In: Proceedings of the 3rd international workshop on agents and peer-to-peer computing, pp 86–97

Borselius N (2002) Mobile agent security. Electron Commun Eng J 14(5):211–218

Bresciani P, Giorgini P, Giunchiglia F, Mylopoulos J, Perini A (2004a) TROPOS: an agent-oriented software development methodology. J Auton Agents Multi Agent Syst 8(3):203–236

Bresciani P, Giorgini P, Mouratidis H, Manson G (2004b) Multi-agent systems and security requirements analysis. In: Proceedings of software engineering for multi-agent systems, LNCS 2940. Springer, Berlin, pp 35–48

Bürkle A, Hertel A, Müller W, Wieser M (2009) Evaluating the security of mobile agent platforms. Auton Agents Multi Agent Syst 18(2):295–311

Chandran SM, Joshi JBD (2005) LoT RBAC: a location and time-based RBAC model. In: Proceedings of the 6th international conference on web information systems engineering (WISE 2005)

Chess D, Harrison C, Kershenbaum A (1996) Mobile agents: are they a good idea? In: Proceedings of 2nd international workshop on mobile object systems, LNCS 1222. Springer, Berlin, pp 25–47

Covington M, Fogla P, Zhan Z, Ahamad M (2002) A context-aware security architecture for emerging applications. In: Proceedings of the 18th annual computer security applications conference, pp 249–258

Cremonini M, Omicini A, Zambonelli F (2000) Coordination and access control in open distributed agent systems: the TuCSoN approach. In: Porto A, Roman GC (eds) Coordination languages and models, LNCS 2906. Springer, Berlin, pp 369–390

Damiani M, Bertino E, Catania B, Perlasca P (2007) GEO-RBAC: a spatially aware RBAC. ACM Trans Inf Syst Security (TISSEC) 10(1):1–42

Farmer WM, Guttman J, Swarup V (1996) Security for mobile agents: authentication and state appraisal. In: Proceedings of the 4th European symposium on research in computer security, pp 118–130

Ferber J (1999) Multi-agent system: an introduction to distributed artificial intelligence. Addison Wesley Longman, Harlow

FIPA (2002a) FIPA abstract architecture specification, SC00001L. http://www.fipa.org/specs/fipa00001/SC00001L.pdf

FIPA (2002b) FIPA agent massage transport service specification, SC00067F. http://www.fipa.org/specs/fipa00067/SC00067F.pdf

FIPA (2002c) FIPA MAS security white paper, f-out-000113

FIPA (2004) FIPA agent management specification, SC00023K. http://www.fipa.org/specs/fipa00023/SC00023K.pdf
Fischmeister S, Vigna G, Kemmerer RA (2001) Evaluating the security of three Java-based mobile agent systems. In: Proceedings of the 5th international conference on mobile agents, pp 31–41
Franklin S, Graesser A (1996) Is it an agent, or just a program? a taxonomy for autonomous agents. In: Proceedings of the workshop on intelligent agents III, LNCS 1193. Agent Theories, Architectures, and Languages, London, pp 21–35
Ghanea-Hercock RA, Gifford I (2001) Top-secret multi-agent systems. Electron Notes Theor Comput Sci 63:77–90
Grandison T, Sloman M (2000) A survey of trust in internet applications. IEEE Commun Surv Tutor 3(4):2–16
Greenberg MS, Byington JC, Harper DG (1998) Mobile agents and security. IEEE Commun Mag 36(7):76–85
Griffiths N (2005) Task delegation using experience-based multi-dimensional trust. In: Proceedings of AAMAS'05, Utrecht, Netherlands, pp 489–496
Hohl F (1998) Time limited blackbox security: protecting mobile agents from malicious hosts. In: Vigna G (ed) Mobile agents and security, LNAI 1419. Springer, Berlin, pp 92–113
Hu Y, Tang C (2003) Agent-oriented public key infrastructure for multi-agent E-service. In: Proceedings of the 7th international conference on knowledge-based intelligent information and engineering systems, pp 114–136
Huynh D (2009) A personalized framework for trust assessment. In: Proceedings of ACM symposium on applied computing, Honolulu, pp 1302–1307
Huynh D, Jennings NR, Shadbolt NR (2004) FIRE: developing an integrated trust and reputation model for open multi-agent Systems. In: Proceedings of 16th ECAI, pp 18–22
Huynh D, Jennings NR, Shadbolt NR (2006a) An integrated trust and reputation model for open multi-agent systems. Auton Agent Multi Agent Syst 13:119–154
Huynh D, Jennings NR, Shadbolt NR (2006b) Certified reputation: how an agent can trust a stranger. In: Proceedings of AAMAS'06, Hakodate, Japan, pp 1217–1224
JADE (2007) http://jade.tilab.com/
Jansen WA (2000) Countermeasure for mobile agent security. Comput Commun 23(17):1667–1676
Jennings NR, Sycara K, Wooldridge M (1998) A roadmap of agent research and development. Auton Agents Multi Agent Syst 1(1):7–38
Joshi JBD, Bertino E, Latif U, Ghafoor A (2005) A generalized temporal role-based access control model. IEEE Trans Knowl Data Eng 17(1):4–23
Jumping Beans (2006) http://jumpingbeans.com/, Accessed 1 Apr 2011
Jung Y, Masoumzadeh A, Joshi JBD, Kim M (2008) RiBAC: role interaction based access control model for community computing. In: Proceedings of 4th international conference on collaborative computing: networking, applications and worksharing
Lampson BW (1974) Protection. SIGOPS Oper Syst Rev 8(1):18–24
Lampson B, Abadi M, Burrows M, Wobber E (1992) Authentication in distributed systems: theory and practice. ACM Trans Comput Syst 10:265–310
Li B, Xing M, Zhu J, Che T (2008) A dynamic trust model for the multi-agent systems. In: Proceedings of international symposiums on information processing, pp 500–504
Liau C (2003) Belief, information acquisition, and trust in multi-agent systems—a modal logic formulation. Artif Intell 149(1):31–60
Lim Choi Keung SN, Griffiths N (2008) Towards improved partner selection using recommendations and trust. In: Falcone R (ed) et al Trust in agent societies, LNAI 5396. Springer, Berlin, pp 43–64
Mouratidis H (2007) Secure tropos: a security-oriented extension of the tropos methodology. Int J Softw Eng Knowl Eng (IJSEKE) 17(2):285–309
Mouratidis H, Giorgini P (2009) Enhancing secure tropos to effectively deal with security requirements in the development of multiagent systems. In: Barley M, Mouratidis H, Unruh A, Spears D, Scerri P, Massacci F (eds) Safety and security in multiagent systems, LNAI 4324. Springer, Berlin, pp 8–26
Mouratidis H, Giorgini P, Manson G (2003) Modeling secure multiagent system. In: Proceedings of the 2nd international joint conference on autonomous agents and multiagent systems, Melbourne, pp 859–866
Mui L, Mohtashemi M, Halberstadt A (2002) A computational model of trust and reputation. In: Proceedings of the 35th international conference on system science, pp 280–287
Necula GC, Lee P (1998) Safe, untrusted agents using proof-carrying code. In: Vigna G (ed) Mobile agents and security, LNCS 1419. Springer, Berlin, pp 61–91
Novák P, Rollo M, Hodík J, Vlcek T (2003) Communication security in multi-agent systems. In: Proceedings of the 3rd international central and eastern European conference on multi-agent systems, pp 454–463
Omicini A, Ricci A, Viroli M (2005) RBAC for organisation and security in an agent coordination infrastructure. Electron Notes Theor Comput Sci 128(5):65–85

Poggi A, Tomaiuolo M, Vitaglione G (2004) A security infrastructure for trust management in multi-agent systems. In: Falcone R, Barber S, Sabater-Mir J, Singh M (eds) Trusting agents for trusting electronic societies, LNCS 3577. Springer, Berlin, pp 162–179

Poslad S, Charlton P, Calisti M (2002) Specifying standard security mechanisms in multi-agent systems. In: Proceedings of autonomous agents and multi-agent systems (AAMAS 2002)

Pujol JM, Sanguesa R, Delgado J (2002) Extracting reputation in multi agent systems by means of social network topology. In: Proceedings of autonomous agents and multi-agent systems (AAMAS 2002), Bologna, pp 467–474

Ramchurn SD, Huynh D, Jennings NR (2004) Trust in multi-agent systems. Knowl Eng Rev 19(1):1–25

Resnick P, Kuwabara K, Zeckhauser R, Friedman E (2000) Reputation systems. Commun ACM 43(12):45–48

Riordan J, Schneier B (1998) Environmental key generation towards clueless agents. In: Vigna G (ed) Mobile agents and security, LNCS 1419. Springer, Berlin, pp 15–24

Roth V (1998) Secure recording of itineraries through cooperating agents. In: Proceedings of 4th workshop on mobile object systems: secure internet mobile computations. INRIA, France, pp 147–154

Sabater J, Sierra C (2001) REGRET: a reputation model for gregarious societies. In: Proceedings of 4th workshop on deception, fraud and trust in agent societies, Montreal, Canada, pp 61–69

Sabater J, Sierra C (2002) Reputation and social network analysis in multi-agent systems. In: Proceedings of 1st AAMAS, Bologna, pp 475–482

Sabater J, Sierra C (2005) Review on computational trust and reputation models. Artif Intell Rev 24:33–60

Samarati P, De Capitani di Vimercati S (2001) Access control: policies, models, and mechanisms. In: Focardi R, Gorrieri R (eds) Foundations of security analysis and design, LNCS 2171. Springer, Berlin, pp 137–196

Sandhu R, Samarati P (1996) Authentication, access control, and audit. ACM Comput Surv 28(1):241–243

Sandhu R, Coyne EJ, Feinstein HL, Youman CE (1996) Role-based access control models. Computer 29(2):38–47

Schillo M, Funk P, Rovatsos M (2000) Using trust for detecting deceitful agents in artificial societies. Appl Artif Intell 14((8):825–848

Sekar R, Ramakrishnan CR, Ramakrishnan IV, Smolka SA (2001) Model-carrying code (MCC): a new paradigm for moblie-code security. In: Proceedings the new security paradigms workshop (NSPW2001), pp 23–30

SeMoA (2007) http://semoa.sourceforge.net/

Such J, Alberola J, Garcia-Fornes A, Espinosa A, Botti V (2009) Kerberos-based secure multiagent platform. In: Hindriks KV, Pokahr A, Sardina S (eds) Programming multi-agent systems, LNCS 5442. Springer, Berlin, pp 197–210

van't Noordende GJ, Brazier FMT, Tanenbaum AS (2004) Security in a mobile agent system. In: Proceedings of the 1st IEEE symposium on multi-agent security and survivability, pp 35–45

Vigna G (1997) Protecting mobile agents through tracing. In: Proceedings of the 3rd ECOOP workshop on mobile object systems, Jyvälskylä

Vuong ST, Fu P (2001) A security architecture and design for mobile intelligent agent systems. ACM SIGAPP Appl Comput Rev 9(3):21–30

Wang Y, Singh MP (2006a) Trust representation and aggregation in a distributed agent system. In: Proceedings of 21st AAAI, pp 1425–1430

Wang Y, Singh MP (2006b) Trust via evidence combination: a mathematical approach based on certainty. TR 2006-11, North Carolina State University, Raleigh

Wang Y, Singh MP (2007) Formal trust model for multiagent systems. In: Proceedings of the 20th international joint conference on artificial intelligence (IJCAI'07), pp 1551–1556

Wang S, Hu J, Liu A, Wang J (2005) Security frame and evaluation in mobile agent system. In: Proceedings of 2nd international conference on mobile technology, applications, and systems, pp 1–6

Wangham MS, da Silva Fraga J, Schmidt R, Rabelo RJ (2004) MASS: a mobile agent security scheme for the creation of virtual enterprises. In: Proceedings of the 1st international workshop on mobility aware technologies and applications, pp 234–243

Wen W, Mizoguchi F (2000) An authorization-based trust model for multiagent systems. Appl Artif Intell 14(9):909–925

Wilikens M, Feriti S, Sanna A, Masera M (2002) A context-related authorization and access control method based on RBAC. In Proceedings of the 7th ACM symposium on access control models and technologies, pp 117–124

Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. Knowl Eng Rev 10:115–152

Young A, Yung M (1997) Sliding encryption: a cryptographic tool for mobile agents. In: Proceedings of the 4th international workshop of fast software encryption, LNCS 1267, pp 230–241

Yu B, Singh MP (2002) Distributed reputation management for electronic commerce. Comput Intell 18(4): 535–549
Yu B, Singh MP (2003) Searching social networks. In: Proceedings 2nd of AAMAS, pp 65–72
Zacharia G, Maes P (2000) Trust through reputation mechanism. Appl Artif Intell 14:881–907
Zhang G, Parashar M (2004) Context-aware dynamic access control for pervasive applications. In: Proceedings of communication networks and distributed systems modeling and simulation conference