Journal of Network and Computer Applications 35 (2012) 480–490

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

# Extracting trust information from security system of a service

Şerif Bahtiyar*, Mehmet Ufuk Çağlayan

*Computer Networks Research Laboratory, Department of Computer Engineering, Boğaziçi University, Bebek, Istanbul 34342, Turkey*

ARTICLE INFO

*Article history:*
Received 17 May 2011
Received in revised form
7 September 2011
Accepted 2 October 2011
Available online 8 October 2011

*Keywords:*
Trust
Security
Information extraction

ABSTRACT

The issue of trust is a research problem in emerging open environments, such as ubiquitous networks. Such environments are highly dynamic and they contain diverse number of services and autonomous entities. Entities in open environments have different security needs from services. Trust computations related to the security systems of services necessitate information that meets needs of each entity. Obtaining such information is a challenging issue for entities. In this paper, we propose a model for extracting trust information from the security system of a service based on the needs of an entity. We formally represent security policies and security systems to extract trust information according to needs of an entity. The formal representation ensures an entity to extract trust information about a security property of a service and trust information about whole security system of the service. The proposed model is applied to Dental Clinic Patient Service as a case study with two scenarios. The scenarios are analyzed experimentally with simulations. The experimental evaluation shows that the proposed model provides trust information related to the security system of a service based on the needs of an entity and it is applicable in emerging open environments.



## 1. Introduction

Having sufficient information is a precondition for making decisions about any property of services in emerging open environments. Emerging open environments are expected to have a large number of services and entities. Entities should obtain or extract sufficient information for making decisions about services. Depending on the goal of each entity, the amount of sufficient information for making decisions may change. Therefore, entities should obtain information based on their needs.

Since the diversity of services increases in open environments, trust to the security of services has become a significant issue. Security properties in computer science are defined as authentication, confidentiality, integrity, and availability (Chivers, 1994; Sun et al., 2008; Subashini and Kavitha, 2011). On the other hand, trust has been investigated in various fields of science, such as philosophy and computer science (Massa, 2007; Hussain et al., 2006) however there is no agreement about the definition and properties of trust.

Trust to the security system of a service by an entity is a significant problem in open environments. The security system of a service is a set of security mechanisms that are implemented according to the security policy of the service. A security policy is a collection of rules that allow or disallow security related actions and events in a service (Kagal et al., 2001; Li et al., 2007; Patz et al., 2001). On the other hand, a security mechanism implements security policies in the system.

### 1.1. Motivation

An entity should trust to security systems of services to interact with them in emerging open environments. Therefore, assessing the trust of the security system of a service according to needs of an entity is becoming a significant issue. Additionally, each entity needs a trust assessment model and information for making trust assessments related to the security system of a service.

In literature, there are many trust computation models that can be applied for assessing the trust of the security system of a service based on the needs of an entity. On the other hand, obtaining information related to the security system of a service according to needs of an entity is not clearly addressed. One entity may gather information for trust computations from other entities and services. The entity may also extract information directly from the security system of a service.

Existing trust models do not provide a solution to extract trust information related to the security system of a service according to needs of a specific entity. Our motivation is the lack of a model for extracting trust information from the security system of a service based on the needs of a specific entity in open environments.

### 1.2. Contributions

In this paper, we propose a model for extracting trust information from the security system of a service in emerging open

* Corresponding author. Tel.: +90 2123597781.
*E-mail addresses:* serif.bahtiyar@boun.edu.tr, serifbahtiyar@gmail.com (Ş. Bahtiyar), caglayan@boun.edu.tr (M. Ufuk Çağlayan).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.10.002

environments. Trust information is extracted from the security system of a service based on the needs of an entity, the security policy of the entity, and the security policy of the service. The proposed model has been applied to the security system of a management service of patients' account as a case study. The proposed model has been evaluated experimentally with simulations in the case study. We can summarize the contributions of our work as below.

- We represent the security policy of an entity and the security policy of a service with sets of atomic units according to needs of the entity. The set representation of security policies provides a way to demonstrate the needs of a specific entity from the security system of a particular service in emerging open environments.
- We propose a novel model for extracting trust information from the security system of a service. The model considers needs of an entity from the security system of a specific service. An entity can extract trust information related to a specific security property and the whole security system of a service by using the model.

The rest of the paper is organized as follows. Section 2 is a brief overview of trust and trust models. We examine some trust related works in Section 3. We present our model for extracting trust information in Section 4. In Section 5, Dental Clinic Patient Service is presented as a case study to show contributions of model. The paper is concluded in Section 6.

## 2. Trust and trust models

In this section, we review how trust is defined in different contexts and examine the issue of trust management in computer science. We also examine main trust models. Further, we present our trust definition related to the security system of a service based on the needs of a specific entity to reflect the significance of extracted information for trust computations.

### 2.1. The definition of trust

Trust is investigated in different fields of science and the definition of trust highly depends on the context and the research field. On the other hand, there is no agreement about the definition of trust (Massa, 2007; Raya et al., 2008; Gollmann, 2006; Bertino et al., 2006). The concept of trust is initially used in social sciences, such as sociology and philosophy. Sociological trust is defined as a particular level of the subjective probability of an agent that assesses a future action of another agent in Gambetta (1988). Subjective factors, risk, confidence, and security are used in the definition of trust in social sciences, where trust shows the expected behavior of an entity (Hoven, 1997; Deutsch, 1958; Misztal, 1996; Grandison and Sloman, 2000).

Despite the nature of trust is subjective, the concept of trust has been widely used in many contexts of computer science because it provides diverse decision making options in different circumstances. Similar to social sciences, trust is defined in a different manner in computer science (Raya et al., 2008; Jøsang et al., 2007; Weeks, 2001; Kuter and Golbeck, 2007; Ryutov, 2007). For example, trust is defined as the judgment expressed by one user about another user, often directly and explicitly, sometimes indirectly through an evaluation of the artifacts produced by that user or her activity on the system in Massa (2007). Jøsang et al. defines two general trust notations, namely reliability trust and decision trust (Jøsang et al., 2007). For instance, reliability trust is the subjective probability by which an individual expects that another individual performs a given action on which its welfare depends on. These definitions of trust show that trust models are highly depend on context.

### 2.2. Trust management in computer science

Trust management contains trust assessment and trust based decision making. The term trust management is first introduced by Blaze et al. (1996), where they define the trust management problem. All components of network services are referred as the trust management problem. Trust management is based on a unified mechanism, flexible to represent all trust relationships, local control of trust relationships, and the separation of a mechanism from a policy. The earliest trust management system is PolicyMaker that is also introduced by Blaze et al. (1996).

After the definition of trust management was introduced, many trust definitions related to trust management have been made in computer science. Trust is defined as a bridge between social needs and security solutions to cope with trust management vulnerabilities for distributed networks (Sun et al., 2008). In service oriented computing, informal definition of trust is the manifestation of reasoning and judgment processes whereas formal definition of trust is a relationship between agents A and B (Trček, 2009). Soft trust and hard trust are defined for mobile computing platforms in Yan (2007). Soft trust considers subjective trust standards and facts whereas hard trust builds up trust through structural and objective regulations, standards, as well as widely accepted rules, mechanisms and sound technologies. Additionally, social trust is analyzed and defined from a computational perspective in Kuter and Golbeck (2007). Trust management is studied in emergency networks for a node revocation in ad hoc networks of cell phones by using a threshold cryptography based scheme in Durresi et al. (2009). These researches show the diverse usage of the term trust management in computer science.

### 2.3. Trust models in computer science

In computer science, trust has three general models that determine the degree of trust relationship between two entities. The first model is the direct trust model that is established through observations (Andert et al., 2002). In the second trust model, trust is transmitted through third parties that is called transitive trust model (Sun et al., 2008; Andert et al., 2002). The transitive trust model is also referred as the indirect trust model (Sun et al., 2008; Krukow, 2006; Yan, 2007). The assumptive trust model is the formal name of the spontaneous trust model and does not necessitate any validation process (Andert et al., 2002). In this paper, our model is a direct trust model.

### 2.4. Our definition of trust

While diverse number of trust definitions exists in computer science, this may lead us to confusion about trust relationships in the context of security (Gollmann, 2006). We define *trust* as the security expectation of an entity from a service according to available security evaluation information of that entity. In this paper, the available security evaluation information is the extracted trust information related to the security system of a service according to needs of a specific entity.

## 3. Overview of trust related work

Trust has been investigated over many years and still attracting academicians as an emerging research field. In this section, we examine some existing trust research related to security in

computer science. Particularly, we concentrate to examine existing works that contain models for extracting or obtaining information for trust computations. The goal of this section is to show the difference of our model from existing trust research.

Trust propagation is significant for an entity to obtain information for recent trust computations. Generally, trust is determined primarily in a source and then it is transmitted to a destination according to different models. For example, a framework for propagation of trust and distrust is proposed by considering different circumstances in Guha et al. (2004). A more specific study about propagation of trust and distrust considers co-citations (Zhu et al., 2009). Semantic based information trust computation and propagation algorithm for semantic web is proposed in Zhang et al. (2009). Moreover, security information flow for trust computations is proposed in Bahtiyar et al. (2010). These models show how an entity can gather information from other entities or services for trust based decision making.

Contemporary services are highly dynamic and untrustworthy. Sensitive data of the services are continuously exposed to the risk of being delivered to final users. Intermediary actors who do not have access rights to data are taking part to data transactions. An approach to manage data privacy for data exchange is proposed in Canfora et al. (2008). The approach considers the front-end trust filter paradigm. The paradigm aims to guarantee high flexibility, reduce the resources required, and limit pervasiveness into applications and devices.

The problem regarding security properties of communicating agents is analyzed in Ma and Orgun (2006). Temporal belief logic is used to show how to establish dynamic trust theories for communication protocols. A trust theory for a given security mechanism of communication systems is presented and a global assumption set is construct. Both the trust theory and the global assumption set are used to show a security property. In another research, an information gain metric is used to dynamically extract tendencies of failure of target agents (Urbano et al., 2010). Autonomic trust extraction is also studied for trustworthy service discovery in urban computing in Jung and Lee (2009). The value of trust is automatically determined according to interactions between a user and a service.

Trust Computing Group (TCG) presents a secure computing environment and a testing prototype to solve trust problems of the secure computing environment (Zhang et al., 2008). Specifically, the prototype intends to eliminate the gap between TCG specifications and product implementations. An automata theory is introduced as a test mechanism to achieve TPM specification compliance test, validate chain of trust compliance by analyzing TCG-BIOS, and use reflection mechanism to test each layer of TSS. The significance of this research is that it divides the entire system into pieces and begins to calculate trust according to these pieces.

Ad-hoc networks are another popular context for the trust research. For instance, trust metrics related to ad-hoc networks are evaluated in Theodorakopoulos and Baras (2006). An example related to trusted routing in mobile ad-hoc networks is presented in Peng et al. (2010). A more specific study that considers essentials for developing a good trust management system for wireless sensor networks is presented in Lopez et al. (2010).

Although these researches contain many contributions to trust, none of them presents a solution that describes the way to extract trust information from the security system of a service based on the needs of a specific entity. In our model, each entity can extract trust information from the security system of a service based on its current needs from that service.

## 4. Extracting trust information

In open environments, each entity has its own needs from the security system of a service to establish trust. Entities in such environments are expected to compute the trust of the security system of a service based on their present needs. On the other hand, trust computations necessitate information so entities have to extract information. Moreover, the entities have to represent their needs from a security system formally and the security system has to be represented in a formal manner.

### 4.1. Environment

Software applications and autonomous agents that represent users in open environment are *entities* in our model. An entity may be a web application on the Internet, where the web application is used by a person. An entity may also be a software agent that is responsible to accomplish a given task autonomously. On the other hand, a service can provide many different services to entities. In this paper, we are interested in security aspects of a service so a *service* represents a security policy and a security system.

The proposed model is for emerging open environments. An emerging open environment is expected to be a dynamic environment and contain diverse number of entities and services, where each entity and service may run on many different platforms. For example, such entities and services may run on a PDA, a cell phone, a data center, etc. Additionally, they can communicate over a ubiquitous network.

### 4.2. Security policy and security mechanism

A *security policy* is a collection of rules that allow or disallow possible actions, events, or something related to the security of an entity or a service. A *security system* is a set of security mechanisms. The security system of a service is a set of security mechanisms enforced according to the security policy of the service. A *security mechanism* of a service is an application to enforce rules of a security policy or could also be interpreted as a security property.

Security policies have different granularities. High level security policies are close to natural languages and they are sometimes represented with formal natural languages. Briefly, high level security policy languages are derivatives of natural languages that are modified according to needs of an application. On the other hand, low level security policies are formal specifications of high level security policies. Therefore, low level security policy specifications are expected to be enforced directly to systems. Figure 1 shows general security policy enforcement hierarchy.

Security policy of an entity represents its security needs from the security system of a service. However, the security system of a service is an enforcement of the security policy of the service.

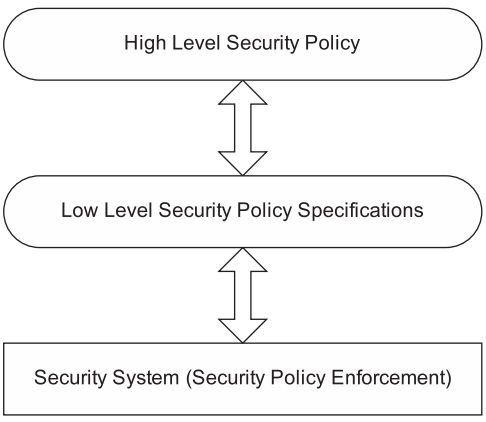


**Fig. 1.** Security policy enforcement hierarchy.

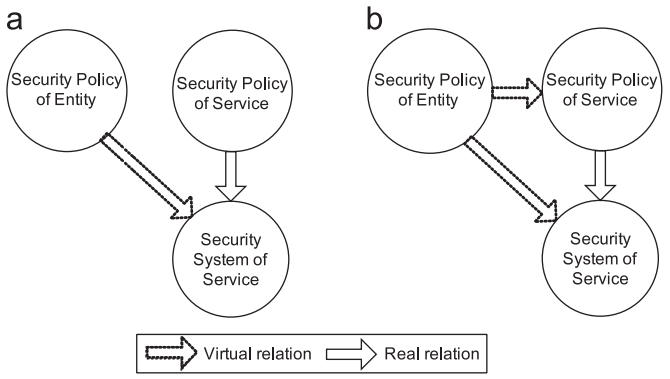


**Fig. 2.** Relations among the security policy of an entity, the security policy of a service, and the security system of the service according to the entity point of view: (a) expected relations, (b) real relations.

Therefore, the security system of a service may not comply with the security policy of an entity. The relation between the security policy of an entity and the security system of a service is shown in Fig. 2a. Briefly, the security system of a service is an enforcement of the security policy of the service, but an entity actually needs to determine the degree the security system enforces its security policy.

The security system of a service may not apply all features of the security policy of the service. The amount of the enforcement is an indicator of trust for an entity so the amount of the enforcement is significant to extract information for trust computations.

The security policy of a service represents the expected behavior of the security system of the service from the service point of view. Additionally, the security policy of an entity represents expected security relations of the entity with services. Therefore, the relation between the security policy of an entity and the security policy of a service is a part of information for trust computations as shown in Fig. 2b.

High level security policies of entities and services have to be represented formally according to needs of each specific entity to extract trust information. Additionally, the security system of a service has to be represented in a formal manner. In our model, an entity represents security policies and security mechanisms with sets of atomic units according to its own needs.

In the proposed model, each entity has access to security policies and security systems of services. Specifically, services send their security policies to entities on the request of the entities. Entities then process and represent the received security policy and the security system of a service according to their own needs. Therefore, each entity may have different representations of security policies and security systems.

Entities and services may have different security policies depending on their specific needs. For example, some entities and services may have access control policies while some others may have communication security policies. On the other hand, some other entities and services may have both access control policies and communication security policies.

### 4.3. Representation of security policy by atomic units

We represent a security policy with atomic units according to needs of an entity from a specific service. An *atomic unit* may be a rule of a security policy or a set of rules of the security policy. For instance, assume that the security policy of an entity has two rules that define the expected access control behavior of a service with which services the entity interacts. The first rule defines access control to a service from intranet and the second rule defines access control from the Internet, where intranet is expected to have more trustworthy entities than the Internet has. Therefore, the first rule necessitates more strict access control than the second rule. If the trustworthiness of intranet differs from the trustworthiness of the Internet on an entity, the entity represents two rules with separate atomic units. On the other hand, another entity may not distinguish the Internet and intranet so the entity may represents the two rules with a single atomic unit.

Set $P_c^s(t)=\{p_1,\ldots,p_m\}$ represents the security policy of service $c$ in an entity, where $m\in\mathbb{Z}^+$ and $p_j$ denotes an atomic unit of set $P_c^s(t)$. A security policy of an entity is also represented with a set of atomic units in the entity. Set $P_c^e(t)=\{p_1,\ldots,p_n\}$ represents the security policy of an entity related to service $c$ in that entity, where $n\in\mathbb{Z}^+$ and $p_k$ denotes an atomic unit of set $P_c^e(t)$.

Assume that the security policy of an entity has two rules related to security expectations from the online ticked reservation service of an airlines. The first rule is *Passengers' information are encrypted and then stored*, whereas the second rule is *All encryptions are carried out on a Trusted Platform Module on the service*. Additionally, assume that the entity represents the first rule with atomic unit *CRYPTO* and the second rule with atomic unit *TPM*. In this case, the set representation of the security policy of the entity is $P_{air}^e(t)=\{CRYPTO,TPM\}$. On the other hand, assume that the security policy of an airline online ticked reservation service has the first rule only so the security policy of the service is represented with $P_{air}^s(t)=\{CRYPTO\}$. Note that security policies may be dynamic and needs of an entity may change in course of time so the elements of a set may change with time.

Either an entity or a service may have complex security policies. The complexity of a security policy depends on the amount of rules and dependencies among the rules. An entity represents only some parts of its security policy, which are related to services depending on its needs from that services. For example, the security policy of an entity may contain rules that are about relationships with other entities, which rules are not related to services. Therefore, the entity does not represent these rules to extract trust information.

In this paper, an entity has one security policy that security policy defines all requirements of the entity with rules. Similarly, a service has a unique security policy that defines requirements of the service with rules from its security system.

### 4.4. Representation of security system by atomic units

We represent the security system of a service from the point of view of an entity with atomic units. Each entity can extract information about all atomic units of the security system of a specific service. Because the security system of a service is a set of security mechanisms, the atomic unit representation of the security system of a service is an atomic unit representation of security mechanisms according to needs of a specific entity.

An *atomic unit* can be a property of a security mechanism or some properties of the security mechanism. Additionally, an atomic unit can be a security mechanism or a set of security mechanisms. For instance, assume that the security system of a service has a password based authentication mechanism and a digital certificate authentication mechanism. The password based authentication mechanism has two properties namely, the minimum length of a password constraint and the password content constraint. An entity may represent the length of a password constraint and the content of a password constraint with different atomic units. On the other hand, another entity may represent the password based authentication mechanism with a single atomic unit. Moreover, some other entities may represent both the password based authentication mechanism and the digital signature based authentication mechanism with one atomic unit.

The security system of service $c$ in an entity is represented with a set of atomic units $\Phi_c(t)=\{\varphi_1,\ldots,\varphi_u\}$, where $u\in\mathbb{Z}^+$. An atomic unit of set $\Phi_c(t)$ is denoted with $\varphi_i$. For example, assume that the security system of the online reservation service of an airline uses an encryption mechanism to store its internal data, which is represented with $\Phi_{air}(t)$. The service may use *DES* symmetric key algorithm, *AES 128* or *AES 256* asymmetric key algorithms for encryptions. In addition, the service may use one of the two different mechanisms for authentications, namely the digital signature based authentication mechanism *DSS* or the password based authentication mechanism *PW*. In this case, set $\Phi_{air}(t)$ may have five atomic units according to needs of an entity, $\{DES,AES128,AES256,DSS,PW\}$.

Each entity may have different granularities for representations of atomic units due to resource costs of entities. It is expected that if all entities have different granularities, the atomic unit representation of a security system will lead to better computations of trust. For instance, symmetric encryptions may perform better than asymmetric ones for a resource limited entity, therefore each encryption mechanism has to be better represented with an atomic unit. On the other hand, another entity may not have a resource limitation so that the entity may use any encryption mechanism and all encryption mechanisms may be represented with one atomic unit, such as $\Phi_{air}(t)=\{CRYPTO,DSS,PW\}$, where in this case, the atomic unit *CRYPTO* represents *DES*, *AES128*, and *AES256* in an aggregated form.

### 4.5. Trust information

An entity expects that the security policy of a service is the same as with its security policy. The entity has rules in its security policy that describe its security needs from the security system of a service. A service has also rules in its security policy that describe its needs from its security system. Briefly, the security policy of a service may differ from the security policy of an entity. If the security policy of service $c$ is different from the security policy of an entity and the entity needs to obtain information from that service for trust computations, the security policy of the service has fewer rules than the security policy of the entity. Formally, $|P_c^e(t)|>|P_c^p(t)|$. The case where security policy of the service has more rules than the security policy of the entity is trivial and does not concern us.

The rules of the security policy of an entity that represent needs of the entity from the security system of a service are a subset of all rules of the entity's security policy. Similarly, the rules of the security policy of a service that are related to needs of a specific entity may be a subset of all rules of the security policy of the service.

#### 4.5.1. Expected sets

Although an entity expects the security policy of a service to be the same with its security policy, the security policy of the service is usually different from the security policy of the entity. Therefore, an entity has an *expected security policy* related to a specific service. The expected security policy related to service $c$ is represented with set $P_c^{xs}(t)$. Set $P_c^{xs}(t)$ has equal number of members with set $P_c^e(t)$ that means $|P_c^e(t)|=|P_c^{xs}(t)|$.

Actually, an entity uses binary relations between set $P_c^e(t)$ and set $P_c^{xs}(t)$ to extract trust information. Therefore, missing atomic units are used to complete the security policy of a service as shown with triangles for set $P_c^{xs}(t)$ in Fig. 3. A *missing atomic unit* stands for an absent rule in the security policy of the service. Missing atomic units do not exist in set $P_c^s(t)$.

The security system of a service may not satisfy its security policy. Because the security policy of a service is dynamic, the security system of the service is also expected to be dynamic. However, the security system of a service may not be updated immediately when the security policy is updated. Additionally, the security system may be implemented incorrectly or it may be incomplete. Therefore, the security system may not represent correct enforcement of its security policy. In other words, set $\Phi_c(t)$ may not have atomic units that are needed to represent all relations between atomic units of set $\Phi_c(t)$ and atomic units of set $P_c^s(t)$. Moreover, set $\Phi_c(t)$ may not have atomic units that are needed to represent all relations between atomic units of set $\Phi_c(t)$ and atomic units of set $P_c^{xs}(t)$. Therefore, an entity has the expected security system of service $c$ that meets requirements for the representation of relations between atomic units of set $\Phi_c(t)$ and atomic units of set $P_c^{xs}(t)$.

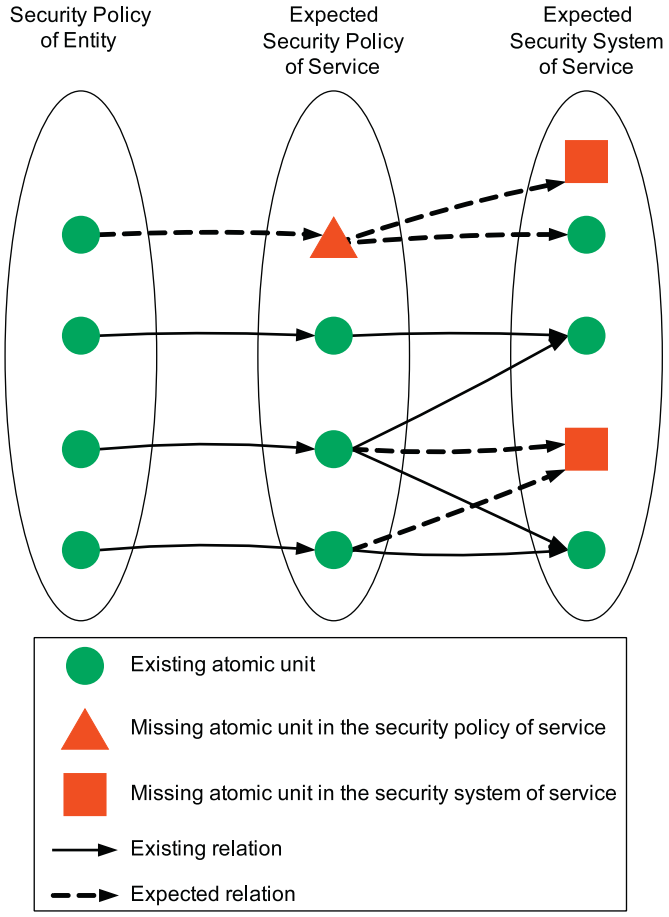


**Fig. 3.** Atomic relations among the security policy of an entity, the expected security policy of a service and the expected security system of the service.

The *expected security system* of service $c$ is represented with $\Phi_c^x(t)$ and the relations between set $P_c^{xs}(t)$ and set $\Phi_c^x(t)$ are shown in Fig. 3. Similar to the expected security policy representation of a service, we use missing atomic units to complete set $\Phi_c^x(t)$. A *missing atomic unit* in $\Phi_c^x(t)$ is represented with a square as shown in Fig. 3. Note that $\Phi_c(t)$ does not contain missing atomic units.

An atomic unit of set $P_c^{xs}(t)$ may be related to one atomic unit or more than one atomic unit of set $\Phi_c^x(t)$ as shown in Fig. 3. Moreover, each atomic unit of set $\Phi_c^x(t)$ may have different weights on an atomic unit of set $P_c^{xs}(t)$. The *weight of atomic unit* $\varphi_i\in\Phi_c^x(t)$ on atomic unit $p_j\in P_c^{xs}(t)$ is represented with $w_{j,i}(t)$, where $w_{j,i}(t)\in[0,1]$ and $w_{j,i}(t)$ satisfies the following condition:

$$\sum_{\forall\varphi_i\in\Phi_c^x(t)} w_{j,i}(t)=1,\quad p_j\in P_c^{xs}(t). \tag{1}$$

If there is no relation between two atomic units, then $w_{j,i}(t)=0$. The value of $w_{j,i}(t)$ represents the significance of atomic unit $\varphi_i$ related to atomic unit $p_j$ for extracting trust information. For instance, if $w_{j,i}(t)=0.6$ and $w_{j,r}(t)=0.4$, then atomic unit $\varphi_i$ is more significant than atomic unit $\varphi_r$ to extract trust information related to atomic unit $p_j$, where $\varphi_i,\varphi_r \in \Phi_c^x(t)$. Additionally, if atomic unit $p_j$ is related only to one atomic unit, such as to atomic unit $\varphi_i$, then $w_{j,i}(t)=1$.

#### 4.5.2. Satisfaction factors

An atomic unit of a security system may not be able to satisfy its specification fully because of many reasons, such as implementation problems and security policies of organizations. For example, assume that an atomic unit of the security system of a service specifies an acceptable password for the authentication mechanism of the service, where a password has to contain at least two capital letters and the password length has to be at least eight characters. If a service allows an entity to determine a password with less than two capital letters but does not allow determining the password length to be less than eight characters, the atomic unit of the security system partially satisfies its specifications.

The *satisfaction factor of an atomic unit of a security system* represents the satisfaction ratio of the atomic unit based on the needs of a specific entity. The satisfaction factor related to the expected security system of service $c$ is represented with $sts_i(t)$, where $sts_i(t)\in[0,1]$ and $\varphi_i \in \Phi_c^x(t)$. If atomic unit $\varphi_i$ fully satisfies its specifications, then $sts_i(t)=1$. On the other hand, if atomic unit $\varphi_i$ does not satisfies any of its specifications or it is a missing atomic unit, $sts_i(t)=0$. Otherwise, $0<sts_i(t)<1$. A satisfaction factor depends on needs of an entity and the security policy of a service on which the security policy is implemented. Therefore, the value of a satisfaction factor is dynamic.

The *satisfaction factor of atomic unit of a security policy* shows how much the security policy of service $c$ satisfies the rule in the expected security policy of the service. We represent the satisfaction factor of atomic unit $p_j \in P_c^{xs}(t)$ with $stp_j(t)\in[0,1]$. Satisfaction factor $stp_j(t)$ is the weighted sum of satisfaction factors of corresponding atomic units in $\Phi_c^x(t)$, which is computed as below:

$$stp_j(t)=\sum_{\forall \varphi_i \in \Phi_c^x(t)} w_{j,i}(t)sts_i(t), \quad p_j \in P_c^{xs}(t). \tag{2}$$

#### 4.5.3. Histories

The security policy of a service and its corresponding security system is in general time varying. Modeling such time variability is significant to extract additional information for more sensitive trust computations. An entity may have more trust to the security system of a service if the security system is improved according to needs of the entity. On the other hand, a change in the security policy of a service or a change in the security system of the service may result in lower trust to the security system on the entity. Specifically, atomic units of the security policy of a service and atomic units of the security system of the service may change with time. Histories of these changes provide information for trust computations.

The *history of atomic unit* $\varphi_i \in \Phi_c^x(t)$ represents changes of the atomic unit in relation to atomic unit $p_j \in P_c^{xs}(t)$ according to needs of a specific entity related to service $c$. We represent the effect of history of atomic unit $\varphi_i \in \Phi_c^x(t)$ with $h_{j,i}^{sss}(t)$, where $p_j \in P^{xs}(t)$ and $h_{j,i}^{sss}(t)\in[-1,1]$. The history effect may be positive or negative and it changes with time. Moreover, all history effects that are related to atomic unit $p_j$ are combined as following, where $h_j^{sss}(t)$ represents the combined histories of atomic units in $\Phi^x(t)$ that is related to atomic unit $p_j$:

$$h_j^{sss}(t)=\sum_{\forall \varphi_i \in \Phi_c^x(t)} h_{j,i}^{sss}(t), \quad p_j \in P_c^{xs}(t). \tag{3}$$

Similar to histories of atomic units of set $\Phi_c^x(t)$, an atomic unit of set $P_c^{xs}(t)$ has a history that represents changes in that atomic unit according to needs of a specific entity. We represent the effect of the history of atomic unit $p_j \in P_c^{xs}(t)$ with $h_j^{ssp}(t)\in[-1,1]$.

Since atomic units of set $\Phi_c(t)$ are enforcements of atomic units of set $P_c^s(t)$, an atomic unit of set $P_c^s(t)$ is also affected by histories of related atomic units of set $\Phi_c(t)$. Therefore, the *overall history effect* on atomic unit $p_j \in P_c^{xs}(t)$ depends on histories of related atomic units of set $\Phi_c^x(t)$ and the history of the atomic unit of set $P_c^{xs}(t)$. The overall history effect related to atomic unit $p_j$ is represented with $h_j(t)\in[-1,1]$ and is computed as below:

$$h_j(t)=\min(1,\max[-1,h_j^{sss}(t)+h_j^{ssp}(t)]). \tag{4}$$

The combination of the overall history effect and the weighted sum of satisfaction factors of an atomic unit in set $P_c^{xs}(t)$ is the information extracted for trust computations from the security policy of a service and related security system. We call *trust information of an atomic unit* in set $P_c^{xs}(t)$ for such information. Trust information related to atomic unit $p_j \in P_c^{xs}(t)$ is represented with $ta_j(t)\in[0,1]$ and is computed as following:

$$ta_j(t)=\begin{cases}1, & stp_j(t)+h_j(t)>1\\ 0, & stp_j(t)+h_j(t)<0\\ stp_j(t)+h_j(t), & \text{otherwise}\end{cases} \tag{5}$$

#### 4.5.4. Perception factor

An entity can extract trust information related to a specific atomic of its set $P_c^e(t)$ or all atomic units of the set according to its present needs from the security system of service $c$. In our model, each atomic unit of set $P_c^e(t)$ depends on an atomic unit of set $P_c^s(t)$. Therefore, trust information of an atomic unit of set $P_c^e(t)$ depends on the trust information of an atomic unit of set $P_c^{xs}(t)$. Moreover, each entity can perceive trust information of an atomic unit of set $P_c^{xs}(t)$ differently so entities have a perception factor for each atomic unit of set $P_c^e(t)$. The *perception factor* shows the effect of atomic unit $p_j \in P_c^{xs}(t)$ to atomic unit $p_k \in P_c^e(t)$ and is represented with $\pi_{k,j}(t)\in[0,1]$. If atomic unit $p_j$ fully affects atomic unit $p_k$, $\pi_{k,j}(t)=1$. If $\pi_{k,j}(t)=0$, atomic unit $p_j$ does not affect atomic unit $p_k$ or atomic unit $p_j$ is a missing atomic unit. Briefly, a perception factor shows the belief of an entity to information received from a service related to a specific atomic unit of set $P_c^e(t)$. A perception factor is private of an entity so it may differ from one entity to another one.

#### 4.5.5. Trust metrics

Extracted trust information related to an atomic unit in set $P_c^e(t)$ consists of all trust information extracted from the security system of a service based on needs an entity. Specifically, *extracted trust information* is a combination of information extracted from the security policy of a service and the security system of the service. Additionally, it depends on the perception of the entity. Extracted trust information from service $c$ in an entity related to atomic unit $p_k \in P_c^e(t)$ is represented with $\iota_k(t)\in[0,1]$ and is computed as following.

$$\iota_k(t)=\pi_{k,j}(t)ta_j(t), \quad p_k \in P_c^e(t), \quad p_j \in P_c^{xs}(t). \tag{6}$$

While extracted trust information related to an atomic unit of set $P_c^e(t)$ shows the trustworthiness of the atomic unit, extracted trust information related to all atomic units of set $P_c^e(t)$ shows the trustworthiness of the security system of service $c$. Extracted trust information related to all atomic units is a weighted sum of

extracted trust information related to each atomic unit of set $P_c^e(t)$. The *impact factor* of an atomic unit in set $P_c^e(t)$ shows how much the extracted trust information related to the atomic unit contributes to all extracted trust information. The impact factor of atomic unit $p_k \in P_c^e(t)$ is represented with $imp_k(t) \in [0,1]$ and it satisfies the condition shown with Eq. (7). If $imp_k(t) = 1$, extracted trust information related to atomic unit $p_k$ has maximum impact to the computation of all extracted trust information. Whereas, if $imp_k(t) = 0$, extracted trust information related to atomic unit $p_k$ has no impact:

$$\sum_{\forall p_k \in P_c^e(t)} imp_k(t) = 1. \tag{7}$$

We represent extracted trust information related to all atomic units of set $P_c^e(t)$ according to needs of an entity from the security system of service $c$ with $\imath(t) \in [0,1]$. Extracted trust information related to all atomic units is computed as following.:

$$\imath(t) = \sum_{\forall p_k \in P_c^e(t)} imp_k(t)\imath_k(t). \tag{8}$$

The value of $\imath_k(t)$ shows the trustworthiness of atomic unit $p_k \in P_c^e(t)$ and the value of $\imath(t)$ shows the trustworthiness of the security system of service $c$ based on the needs of an entity. For instance, $\imath_k(t) = 1$ means that atomic unit $p_k$ has maximum trustworthiness on the entity. On the other hand, if $\imath(t) = 0$, the security system of the service is not trustworthy according to needs of a specific entity.

Consequently, our contribution is to show a way to extract trust information for complicated trust computations. Trust information is extracted from the security system of a service based on the needs of a specific entity. The security policy of the entity and the security policy of the service contribute to extract trust information. We have two types of trust information. The first type of information is related to a specific security property of a service, whereas, the second type is related to all security system of the service.

## 5. Case study: dental clinic patient service

We have simulated the proposed model with a case study and have conducted several experiments. The case study and experiments have two objectives. The first objective is to illustrate the applicability of the proposed model on a realistic application. The second objective is to show the effects of changes in a security policy and in a security system to extracting trust information. Therefore, the case study is about extracting trust information from a dental clinic patient service according to needs of a person. Experiments are conducted with two scenarios. In the first scenario, we have analyzed the effect of changes in security system of a service. In the second scenario, we have investigated effects of changes in the security policy of the service and in the security policy of an entity.

We simulated the scenarios and showed the performance results based on our proposed model. Simulations were carried out by using MATLAB R2009b version 7.9.0.529 that run on a PC with Intel Core 2 Duo E8400 3.00 GHz processor and 3GB of RAM.

### 5.1. Case study overview

Suppose that security policies and security systems of dental clinics are public for potential patients who wish to get appointments from the Internet. Suppose also that a patient has dental problems and she needs to get an appointment from a dental clinic close to her location. There are many dental clinics close to the patient's location. Dental clinics store patients' records on data storages that are accessible from the Internet. However, the patient knows that some clinics have weak security systems of their patients' record management service. Therefore, medical records may be revealed by an adversary.

Medical privacy is significant for the patient so that a dental clinic has to keep patients' medical records from being revealed to other people. If medical records of the patient are revealed, the personal life of the patient will be affected. Moreover, the patient may have financial problems, such as increasing of insurances costs. On the other hand, the patient has some positive recommendations about a dentist in dental clinic BDENT. Before getting an appointment from the dentist in BDENT, the patient needs to assess the trust of the security system of BDENT Patient Service.

The patient has a software agent (PA) that represents herself on the Internet, where the software agent is the entity in this case study. The software agent can get information related to BDENT Patient Service and it can assess the trust of the security system of the service according to privacy needs of the patient. Then, the patient can decide whether to get an appointment or not by considering the trust assessments.

The *security policy of entity PA* has the following rules related to the security system of a service:

1. Patients' records have to be stored in an encrypted form and the encryption keys have to be kept secure in a hardware device.
2. Dentists have to access only to their patients' records by using password based authentication, where passwords are stored in an encrypted form.
3. The security system of a service has to contain a monitoring mechanism for auditing all accesses to patients' records.

According to the rules, PA represents its security policy with three atomic units, where each atomic unit corresponds to a rule in the security policy. The first rule is represented with atomic unit *ECE*. The second and the third rules are represented with atomic units *ATE* and *ADE* respectively. Therefore, the set representation of the security policy of entity PA is $P_{BDENT}^e(t) = \{ECE, ATE, ADE\}$.

The security policy of BDENT has following rules related to Patient Service:

1. Patients' records are stored in an encrypted form.
2. Dentists access only to their patients' records by using password based authentication, where passwords are stored in an encrypted form.

PA knows the security policy of BDENT because the security policy is public. PA represents the security policy with two atomic units, where *ECS* corresponds to the first rule and *ATS* corresponds to the second rule. In this case, the set representation of the security policy of BDENT is $P_{BDENT}^s(t) = \{ECS, ATS\}$. Normally, PA expects to see a rule in the security policy of BDENT that is related to the third rule in its security policy, but the security policy does not contain such a rule. Therefore, PA has a missing atomic unit in $P_{BDENT}^{xs}(t)$ that is related to the third rule of its security policy. The missing rule is represented with *ADS* so $P_{BDENT}^{xs}(t) = \{ECS, ATS, ADS\}$.

The security system of BDENT has a password based authentication mechanism. Both passwords and patients' records are encrypted with an encryption mechanism. The encryption mechanism uses AES algorithm for encryptions. However, encryption keys are not stored in a hardware device. Additionally, the security system does not contain any monitoring mechanism for logging accesses to patients' records. Therefore, the set representation of the security system of BDENT has two atomic units. Atomic unit *AES* represents the encryption mechanism whereas

atomic unit *PW* represents the password based authentication mechanism. Atomic unit *TPM* represents a possible hardware for storing encryption keys and atomic unit *LOG* represents a possible monitoring mechanism of the security systems. Atomic units *TPM* and *LOG* are missing atomic units in this case so the set representation of the security system is $\Phi_{BDENT}(t)=\{AES,PW\}$ whereas the set representation of the expected security system is $\Phi^{x}_{BDENT}(t)=\{AES,PW,TPM,LOG\}$. Relations among the sets in PA are show in Fig. 4.

### 5.2. Scenario 1: effects of changes in security system

In this scenario, we examine impacts of changes in the security system of a service related to extracted trust information. Specifically, we update some security mechanisms of the security system of BDENT and show effects of the update. Moreover, we analyze effects of $sts_i(t)$, such that $\varphi_i \in \Phi^{x}_{BDENT}(t)$.

Because we are interested in effects of $sts_i(t)$ and changes in the number of atomic units of the security system in this scenario, we chose some parameters to be constant. We assume that effects of histories are zero, $h^{sss}_{j,i}(t)=0$ and $h^{ssp}_{j}(t)=0$, where $p_j \in P^{xs}_{BDENT}(t)$ and $\varphi_i \in \Phi^{x}_{BDENT}(t)$. The perception factors are $\pi_{ECE,ECS}(t)=0.8$, $\pi_{ATE,ATS}(t)=1$ and $\pi_{ADE,ADS}(t)=0$. The impact factors are $imp_{ECE}(t)=0.3$, $imp_{ATE}(t)=0.5$ and $imp_{ADE}(t)=0.2$. The weight factors are $w_{ECS,TPM}(t)=0.3$, $w_{ECS,AES}(t)=0.7$, $w_{ATS,AES}(t)=0.4$, $w_{ATS,PW}(t)=0.6$, and $w_{ADS,LOG}(t)=1$.

Normally, it is expected that a satisfaction factor does not change unless there are changes in the related security mechanisms. However, needs of entities may change so satisfaction factors are expected to vary with time. Therefore, values of $sts_{AES}(t)$ vary between 0.85 and 0.95 in this scenario as shown in Fig. 5.

Initially, the minimum password length has to be at least eight characters in BDENT so that values of $sts_{PW}(t)$ are between 0.65 and 0.75 for $0<t<10$. However, the minimum password length is changed to be at least four characters at $t=10$ because some patients do not remember their passwords and have to contact to the security management desk, which circumstance brings additional cost to BDENT. However, IT management department of BDENT observes that patients have low trust to services with short password lengths therefore the minimum password length is updated to be at least six characters at $t=20$. The security system of BDENT is updated according to these changes. PA also changes values of the satisfaction factor related to PW to be between 0.25 and 0.35 for $10<t<20$ and between 0.45 and 0.5 for $t>19$ as shown in Fig. 5. Additionally, the security policy of BDENT and the security system are updated simultaneously according to these changes.

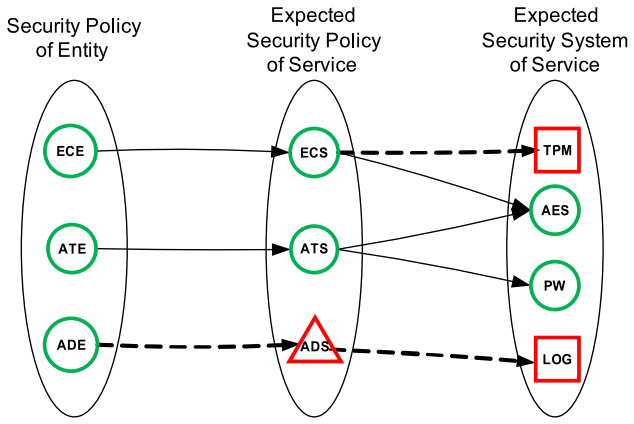


**Fig. 4.** Atomic relations among the sets in PA related to BDENT.

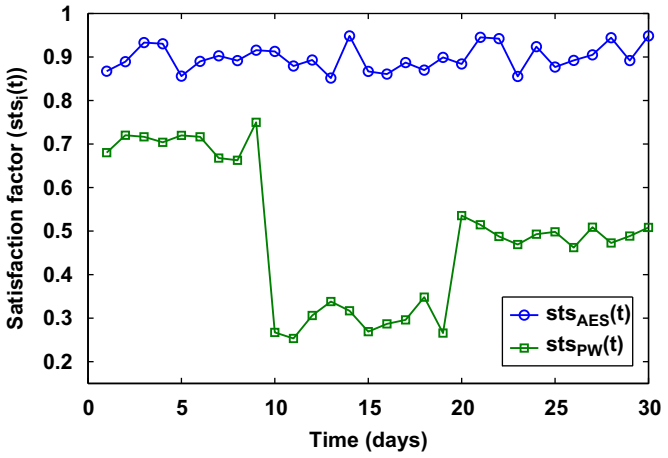


**Fig. 5.** The change of satisfaction factors when the security system of BDENT is changed.

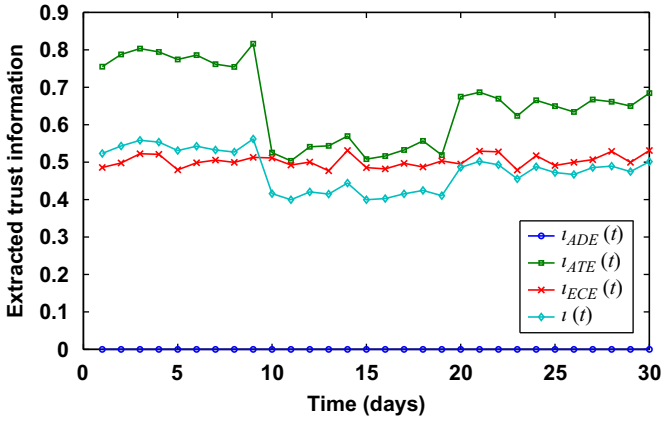


**Fig. 6.** Extracted trust information when the security system of BDENT is changed.

Actually, the security system of BDENT has monitoring software that logs accesses to patients' records. However, the monitoring software has been turned off until $t=10$. When the required length of passwords are updated at $t=10$, the monitoring software is turned on. On the other hand, the security policy of BDENT is never updated so PA does not consider this change and $\iota_{ADE}(t)=0$ as in Fig. 6. Briefly, changes in a security system of a service do not affect the extracted trust information if the changes do not have corresponding rule in the security policy of the service.

In this scenario, extracted trust information varies for some atomic units as shown in Fig. 6. Since perception factor $\pi_{ADE,ADS}(t)=0$, extracted trust information $\iota_{ADE}(t)$ is zero. On the other hand, extracted trust information related to other atomic units varies all the time. However, the variance of $\iota_{ATE}(t)$ is greater than the variance of $\iota_{ECE}(t)$ when the security system is updated. For example, the atomic unit *PW* is updated at $t=10$ and $t=20$, where $\iota_{ATE}(t)$ changes considerably. $\iota_{ECE}(t)$ does not vary as much as $\iota_{ATE}(t)$ because there is no change in atomic units *AES* and *TPM* in $\Phi_{BDENT}(t)$. These examples show that extracted trust information related to an atomic unit of a security system depends highly on updates of the atomic unit.

Extracted trust information related to all security system of BDENT depends on impact factors and extracted trust information related to each atomic unit of the security policy of PA as shown in Fig. 6. Therefore, $\iota(t)$ does not oscillate as $\iota_{ATE}(t)$ does and the value of $\iota(t)$ varies more than the value of $\iota_{ATE}(t)$. This scenario

shows that the proposed model reflects changes in the security system of a service according to needs of an entity.

### 5.3. Scenario 2: effects of changes in security policies

Our goal in this scenario is to show effects of changes in the security policy of a service and in the security policy of an entity. Moreover, we present effects of overall history. Specifically, we update the security policy of BDENT and the security policy of entity PA by considering the facts in the previous scenario to accomplish the goal.

The monitoring software is turned on without updating the security policy of BDENT in Scenario 1. The security policy of BDENT is updated when the monitoring software is turned on at $t=10$ in this scenario. A new rule related to the change is added to the security policy of BDENT. The new rule is *Accesses to patients' records by dentists are logged by the monitoring software*, which is represented with atomic unit *ADS*. PA updates $P^s_{BDENT}(t)$ at $t=10$, where $P^s_{BDENT}(t)=\{ECS,ATS,ADS\}$. Additionally, perception factor $\pi_{ADE,ADS}(t)$ is changed to be 0.6 after $t=10$.

On the other hand, managers of BDENT believe that storing patients' records in an encrypted form brings additional cost to manage the database. They also believe that an adversary cannot access to the database of BDENT from the Internet or from the physical environment. Therefore, the first rule of the security policy of BDENT is removed at $t=25$. Additionally, PA updates $P^s_{BDENT}(t)$ by removing *ECS* from $P^s_{BDENT}(t)$ at $t=25$. The entity also changes perception factor to be $\pi_{ECE,ECS}(t)=0$ for $t\geq 25$.

Figure 7 shows changes in satisfaction factors after the security policy updates and Fig. 8 shows changes in extracted trust information related to the security policy change. Extracted trust information related to a rule of the security policy of BDENT is better reflected when there is an update in the rule. Additionally, extracted trust information related to all atomic units behaves as expected. For instance, the value of $\iota(t)$ depends on extracted trust information related to each individual atomic unit as shown in Fig. 8.

The owner of PA observes that many of patients' services of dental clinics do not contain monitoring mechanisms. Moreover, the dental clinics do not share their logging data related to accesses to their security systems with entities. Therefore, the patient updates the security policy of PA by removing the third rule from the security policy at $t=15$.

PA updates its set representations of security policies and set representation of the security system according to the change of its security policy. In the updated form, set $P^e_{BDENT}(t)=\{ECE,ATE\}$ after $t=15$. Because atomic unit *ADE* is removed from set $P^e_{BDENT}(t)$, sets $P^s_{BDENT}(t)$ and $\Phi_{BDENT}(t)$ do not contain atomic units associated with atomic unit *ADE* after $t=15$. Therefore, set representations are $P^s_{BDENT}(t)=\{ECS,ATS\}$ and $\Phi_{BDENT}(t)=\{AES,PW\}$ for $15<t\leq 25$. Expected set representations are $P^{xs}_{BDENT}(t)=\{ECS,ATS\}$ and $\Phi^x_{BDENT}(t)=\{TPM,AES,PW\}$ for $t>15$. Since the first rule is removed from the security policy of BDENT at $t=25$, set representation of $P^s_{BDENT}(t)$ becomes $\{ATS\}$ for $t>25$.

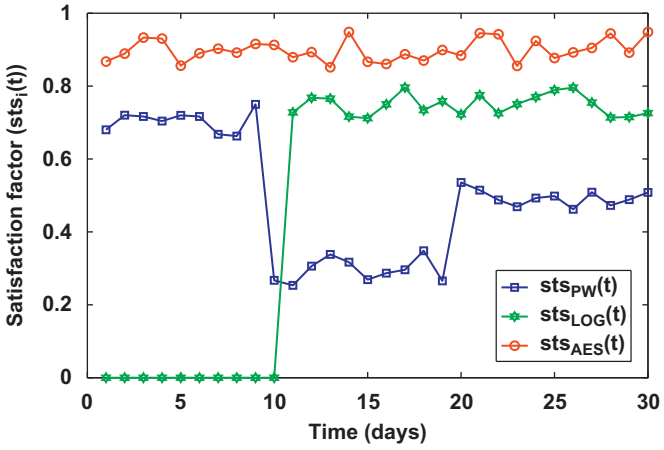


**Fig. 7.** The change of satisfaction factors when the security policy of BDENT is changed.

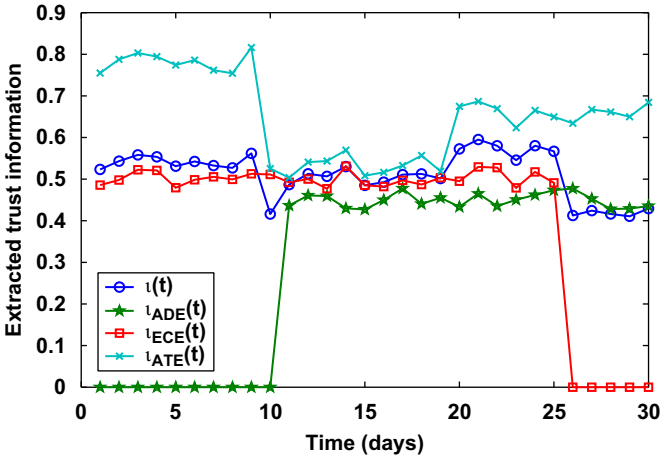


**Fig. 8.** Extracted trust information when the security policy of BDENT is changed.

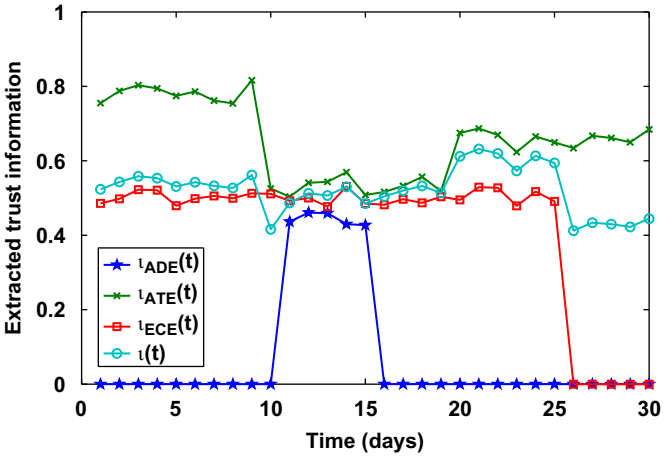


**Fig. 9.** Extracted trust information after removing the first rule of the security policy of PA.

Because of the change in the number of atomic units of $P^e_{BDENT}(t)$, PA updates importance factors at $t=15$. Specifically, we assume that the second rule of the security policy of the entity is more significant than the first rule. Therefore, $imp_{ECE}=0.35$ and $imp_{ATE}=0.65$ for $t>25$ in this scenario.

Effects of changes in the security policy of PA is shown in Fig. 9. The entity removes atomics units from its sets representing security policies and the security system that are related to the third rule of its security policy at $t=15$. Therefore, $\iota(t)$ depends only on $\iota_{ECE}(t)$ and $\iota_{ATE}(t)$ after $t>15$. Specifically, the value of $\iota(t)$ slightly increases because the value of $\iota_{ADE}(t)$ is always smaller than values of $\iota_{ECE}(t)$ and $\iota_{ATE}(t)$ in this scenario. In short, our model reflects changes in the security policy of an entity to extract trust information related to the security of a service.

The experimental results that are shown in Figs. 5–9, do not contain effects of histories. However, extracted trust information usually depends on the histories. The overall history $h_j(t)$ related to atomic unit $p_j\in P^{xs}_{service}(t)$ is determined according to the history

of the security policy $h_j^{ssp}(t)$ and the history of the security system of the service $h_j^{sss}(t)$. The effects of $h_j^{ssp}(t)$ and $h_j^{sss}(t)$ to $h_j(t)$ for all possible values of $h_j^{ssp}(t)$ and $h_j^{sss}(t)$ are shown in Fig. 10. Additionally, trust information $ta_j(t)$ related to atomic unit $p_j \in P^{xs}_{BDENT}(t)$ is computed according to the overall history $h_j(t)$ and the satisfaction factor $stp_j(t)$. Possible effects of $h_j(t)$ and $stp_j(t)$ to $ta_j(t)$ are shown in Fig. 11.

Trust information of atomic unit $p_j \in P^{xs}_{service}(t)$ contributes to the computation of extracted trust information related to atomic unit $p_i \in P^{e}_{service}(t)$. On the other hand, each entity can evaluate histories of security policies of services and their corresponding histories of security system depending on their needs. Therefore, one can apply results in Fig. 11 to the results in this case study to see possible effects of histories.

This scenario shows that our model reflects changes in the security policy of an entity and the security policy of a service for extracting trust information related to the security system of the service based on the needs of the entity. Moreover, the scenario presents possible history effects.

The case study shows that the proposed model is applicable to entities in open environments. Each entity can extract trust information related to the security system of a service according to its security policy and the security policy of the service even though the security policies and the security system may be dynamic. Moreover, an entity can evaluate the security system of a service based on its present needs.

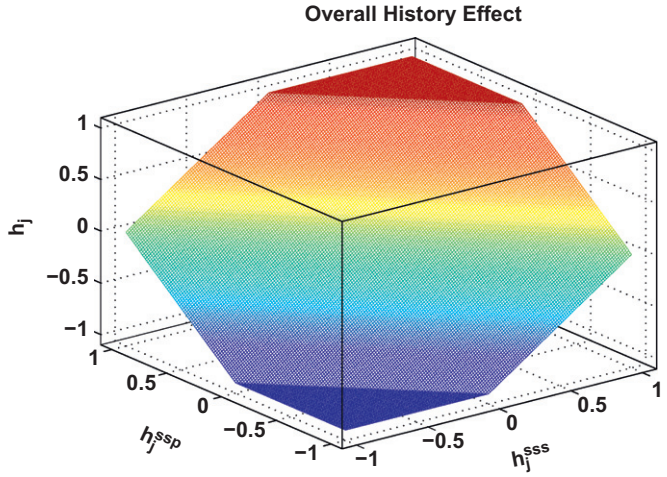


**Fig. 10.** The change of overall history of atomic unit $p_j$ in security policy of the service.

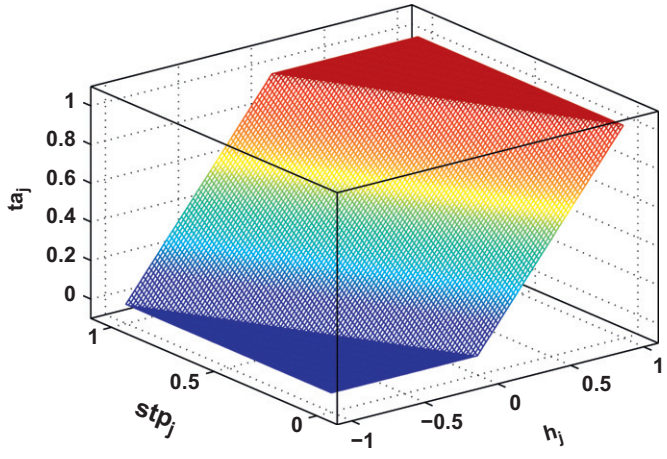


**Fig. 11.** The change of trust information of atomic unit $p_j$ related to the overall history and the satisfaction factor.

## 6. Conclusion

Open environments are expected to support a large number of various services that interact with many different autonomous entities. Such diversity of services leads to trust problems in entities related to security systems of services. Moreover, the trust problems create new research challenges in emerging open environments. One such challenge is to obtain information related to the security system of a service for trust computations. In this paper, we have studied the challenge of obtaining trust information from the security system of a service in emerging open environments.

We have proposed a model for extracting trust information from the security system of a service based on the needs of a specific entity in emerging open environments. The security needs of an entity from a service are represented in the security policy of the entity. The security system of a service is an enforcement of the security policy of the service. Therefore, we have considered the security policy of an entity, the security policy of a service and the security system of the service for extracting trust information.

In the proposed model, security policies and security systems are represented with sets of atomic units. A security policy consists of rules, where each rule is an atomic unit. On the other hand, a security system consists of security mechanisms that are represented with atomic units. Therefore, an entity has a set for atomic units of its security policy, a set for atomic units of the security policy of a service, and a set for atomic units of the security system of the service. An entity only represents rules of security policies that are related to needs of the entity from the service. Furthermore, the entity represents only security mechanisms of the service that are related to needs of the entity.

In our model, an entity can extract trust information about a specific atomic unit and trust information about all atomic units. Specifically, an entity can extract trust information related to a specific rule of its security policy that means the entity may need to determine the trust of a specific security property of the security system of a service. The entity may also need to have trust information related to whole security system of the service so it can extract trust information about whole security system.

A case study for Dental Clinic Patient Service of BDENT has been presented with two scenarios to show the way to extract trust information from the security system of a service. We simulated the scenarios to evaluate the proposed model. In the first scenario, we analyzed effects of changes in the security system of the service. In the other scenario, we investigated effects of changes in the security policy of an entity and changes in the security policy of the service. The evaluation results show that the proposed model can provide information about the security system of a service based on specific needs of an entity for trust computations in emerging open environments.

Finally, as one of the future works to be proceeded is to analyze the proposed model with much complicated case studies.

## Acknowledgments

This work is supported by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610.

## References

Andert D, Wakefield R, Weise J. Trust modeling for security architecture development. Technical Report, Sun Microsystems, Inc.; 2002.

Bahtiyar Ş, Cihan M, Çağlayan MU. A model of security information flow on entities for trust computation. In: The third IEEE international symposium on

trust, security and privacy for emerging applications (TSP-10). Bradford, UK: IEEE Computer Society; 2010. p. 803–8.
Bertino E, Khan LR, Sandhu R, Thuraisingham B. Secure knowledge management: confidentiality, trust, and privacy. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 2006;36:429–38.
Blaze M, Feigenbaum J, Lacy J. Decentralized trust management. In: IEEE symposium on security and privacy; 1996.
Canfora G, Costante E, Pennino I, Visaggio CA. A three-layered model to implement data privacy policies. Computer Standards & Interfaces 2008;30:398–409.
Chivers H. Security and systems engineering. Technical report. Department of Computer Engineering and University of York, Heslington, York, UK; 1994.
Deutsch M. Trust and suspicion. The Journal of Conflict Resolution 1958;2:265–79.
Durresi A, Durresi M, Paruchuri V, Barolli L. Trust management in emergency networks. In: The 23rd IEEE international conference on advanced information networking and applications (AINA-09); 2009. p. 167–74.
Gambetta D. Can we trust trust? In: Trust: making and breaking cooperative relations. Basil Blackhead; 1988. p. 213–37.
Gollmann D. Why trust is bad for security. Electronic Notes in Theoretical Computer Science 2006;157:3–9.
Grandison T, Sloman M. A survey of trust in internet applications. IEEE Communications Survey 2000;3:2–16.
Guha R, Kumar R, Raghavan P, Tomkins A. Propagation of trust and distrust. In: Proceedings of the 13th international conference on World Wide Web. ACM; 2004. p. 403–12.
Hoven DJD. Computer ethics and moral methodology, vol. 28. The Methaphilosophy Foundation and Blackwell Publishers Ltd.; 1997. p. 234–47.
Hussain F, Chang E, Dillon T. Comparative analysis of trust and security. In: IEEE international conference on service operations and logistics, and informatics, SOLI 2006; 2006.
Jøsang A, Ismail R, Boyd C. A survey of trust and reputation systems for online service provision. Decision Support Systems 2007;43:618–44.
Jung K, Lee Y. Autonomic trust extraction for trustworthy service discovery in urban computing. In: Proceedings of the 2009 eighth IEEE international conference on dependable, autonomic and secure computing; 2009. p. 502–7.
Kagal L, Finin T, Joshi A. Trust-based security in pervasive computing environments. IEEE Computer 2001;34:154–7.
Krukow K. Towards a theory of trust for the global ubiquitous computer. PhD thesis. Department of Computer Science, University of Aarhus, Denmark; 2006.
Kuter U, Golbeck J. Sunny: A new algorithm for trust inference in social networks using probabilistic confidence models. In: Proceedings of the twenty-second AAAI conference on artificial intelligence; 2007.
Li J, Huai J, Hu C. Peace-vo: A secure policy-enabled collaboration framework for virtual organizations. In: Twenty-sixth IEEE international symposium on reliable distributed systems, SRDS 2007; 2007.
Lopez J, Roman R, Agudo I, Gago MCF. Trust management systems for wireless sensor networks: Best practices. Computer Communications 2010;33: 1086–93.
Ma J, Orgun M. Trust management and trust theory revision. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 2006;36: 451–60.
Massa P. A Survey of Trust Use and Modeling in Real Online Systems. In: Trust E-services: Technologies, practices and challenges. Idea Group Inc.; 2007. p. 51–83.
Misztal B. Trust in modern societies: the search for the bases of social order. Polity Press; 1996.
Patz G, Condell M, Krishnan R, Sanchez L. Multidimensional security policy management for dynamic coalitions. In: DARPA information survivability conference & exposition II, DISCEX 2001; 2001.
Peng S, Jia W, Wang G, Wu J, Guo M. Trusted routing based on dynamic trust mechanism in mobile ad-hoc networks. IEICE Transactions on Information and Systems 2010;E93-D:510–7.
Raya M, Papadimitratos P, Gligor V, Hubaux J. On data-centric trust establishment in ephemeral ad hoc networks. In: IEEE INFOCOM 2008, The 27th conference on computer communications; 2008.
Ryutov T. A socio-cognitive approach to modeling policies in open environments. In: Eighth IEEE international workshop on policies for distributed systems and networks, POLICY 2007; 2007.
Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 2011;34: 1–11.
Sun YL, Han Z, Liu KJR. Defense of trust management vulnerabilities in distributed networks. IEEE Communications Magazine 2008;46:112–9.
Theodorakopoulos G, Baras JS. On trust models and trust evaluation metrics for ad hoc networks. IEEE Journal on Selected Areas in Communications 2006;24: 318–28.
Trček D. A formal apparatus for modeling trust in computing environments. Mathematical and Computer Modelling 2009;49:226–33.
Urbano J, Rocha AP, Oliveira E. Trustworthiness tendency incremental extraction using information gain. In: Proceedings of the 2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology, vol. 02. IEEE Computer Society; 2010. p. 411–4.
Weeks S. Understanding trust management systems. In: IEEE symposium on security and privacy, S&P 2001; 2001.
Yan Z. Trust Management for Mobile Computing Platforms. PhD thesis. Department of Electrical and Communication Engineering, Helsinki University of Technology, Network Laboratory; 2007.
Zhang B, Xiang Y, Xu Q. Semantics based information trust computation and propagation algorithm for semantic web. In: Fifth international conference on wireless communications, networking and mobile computing, 2009. WiCom 2009; 2009. p. 1–4.
Zhang H, Luo J, Yan F, Xu M, He F, Zhan J. A practical solution to trusted computing platform testing. In: Third Asia-Pacific trusted infrastructure technologies conference, APTC 2008; 2008.
Zhu Y, Li Y, Ren Y. Research on propagation of trust and distrust by means of co-citation. In: Sixth international conference on service systems and service management, 2009. ICSSSM 2009; 2009. p. 943–8.