Protecting Sensitive Attributes in Attribute Based Access Control

Guoping Zhang, Jing Liu, and Jianbo Liu

School of Computer & Communication Engineering
China University of Petroleum, Qing Dao, China
zhanggp@upc.edu.cn, {liujing415jsj,liu1jian2bo3}@163.com

Abstract. Attribute Based Access Control (ABAC) has gradually become a hot research topic in distributed systems. While frequent disclosure of subject attributes, resource attributes or environment attributes may lead to leaks of sensitive information. This article mainly focuses on protecting privacy of resource requester in the process of ABAC, and presents a trust based sensitive attributes protection mechanism which can disclose attributes through comparing resource requester's attribute sensitivity with resource provider's trust level. After experiments comparison with Beth model, we get a conclusion that this mechanism has higher accuracy, without violating resource requester's privacy.

Keywords: Attributes, Sensitivity, Trust Level, Privacy Protection, Malicious Recommendation.

1 Introduction

Most distributed systems are operated in an open and dynamic network environment, across multiple security domains. Because Attribute Based Access Control (ABAC) [1] can offer a fine-grained, dynamic and cross-domain access control mechanism, it has gradually become a hot research topic in distributed systems. There arises an important issue with ABAC model that frequent disclosure of attributes will inevitably lead to leaks of sensitive information.

In order to solve the problem above, this article proposes a trust based sensitive attributes protection mechanism (TSAP). This mechanism includes the concept of trust, and divides trust into direct trust and recommended trust [2]. Meanwhile TSAP can effectively get rid of malicious recommendation. TSAP mechanism decides which attributes to disclose, through comparing resource the requester's attribute sensitivity and the resource provider's trust level.

In this article, we begin with relevant definitions of TSAP mechanism, and basic work flow of TSAP in section 2. In section 3 we describe the concept of trust evaluation about resource provider in detail, and give computational formula of direct trust and recommended trust respectively. Then we verify the validity of TSAP by a simple case study and experiments in section 4 and introduce related work in section 5. At last, we reach the conclusion of this article and point out the future work.

A. Ghose et al. (Eds.): ICSOC 2012, LNCS 7759, pp. 294–305, 2013.

2 Trust Based Sensitive Attributes Protection

Generally, an ABAC system is made up of subjects, objects, and access control policies. For simplicity, we replace resource requester (resource provider) with subject (object), and both are uniformly called entities in the following.

2.1 The Relevant Definitions

- *Attribute Sensitivity*: Attribute Sensitivity means the importance level of subject attributes. We represent the attribute sensitivity by Sens∈[0,1].
- *Trust Evaluation*: Trust Evaluation means a trust estimate about an object made by a subject. Here, the trust means the confidence level of the object and we represent object's trust level by *T*.
- *Direct Trust*: Direct Trust expresses subjective evaluation of a subject to an object, which is gained from history records of direct access from the subject to the object.
- *Recommended Trust*: Recommended Trust is also called indirect trust, which can be gained from a trusted third party or other entities' recommendation
- *Honest Level*: We represent the authentic degree of entities' evaluation about target objects by Honest Level which is called *H* for short.
- *Attribute Access Policy*: Attribute Access Policy is a kind of access control policy that can protect one's own sensitive attributes, in which an object is supposed to own certain attributes values before it requests for some attributes of the subject.

2.2 TSAP Basic Flow

The whole protection process of TSAP is seen in Fig.1, and it includes division of attribute sensitivities, formulation of attribute access policies, trust evaluations to objects, acquirement of sensitive attributes and access to objects.

2.2.1 Division of Attribute Sensitivity

Generally, different subject attributes reflect different information about a subject, so sensitivities of attributes should also be divided into different values. Attributes of low sensitivity can be published in most network environments. However, attributes of high sensitivity can be published only in a trusted network environment. In addition, a subject defines sensitivities not only for attributes it owns, but also for the attributes it does not own to prevent disclosing information about the subject in case objects learn that the subject does not own certain attributes.

2.2.2 Formulation of Attribute Access Policies

In the process of an access control, a subject may need to disclose some sensitive attributes generally so as to access an object. If a subject distrusts the target object in the beginning, the subject can disclose the corresponding attributes only when the object submits the attributes that formulated in the Attribute Access Policy. For example, an Attribute Access Policy is as follows:

If $ra_1=y_1$ and $ra_2=y_2$; imply disclosure SA_1;

In the description above, ra_1 and ra_2 represents two attribute values of the object, and y_1 and y_2 mean the specific value of corresponding attributes. When this policy is met, the object can receive a response about the attribute SA_1. Otherwise, the attribute request of the object will be refused.

In addition, the subject can also formulate corresponding Attribute Access Policies for attributes of higher sensitivity which are not owned by the subject itself. Only when the object meets these policies, the subject can tell whether it owns those attributes or not.

2.2.3 Trust Evaluation to Objects

In the process of ABAC, an important basis for a subject to disclose attributes is it has trust in objects, which also means trust evaluations to objects. Here, trust evaluations to objects derive from two aspects, one aspect is subjective evaluations from subjects to target objects, namely Direct Trust, and the other aspect is a recommendation from a trusted third party to target object, namely Recommended Trust. Here, we use T to represent trust level from subjects to objects, T_D to direct trust, and T_R to recommended trust.

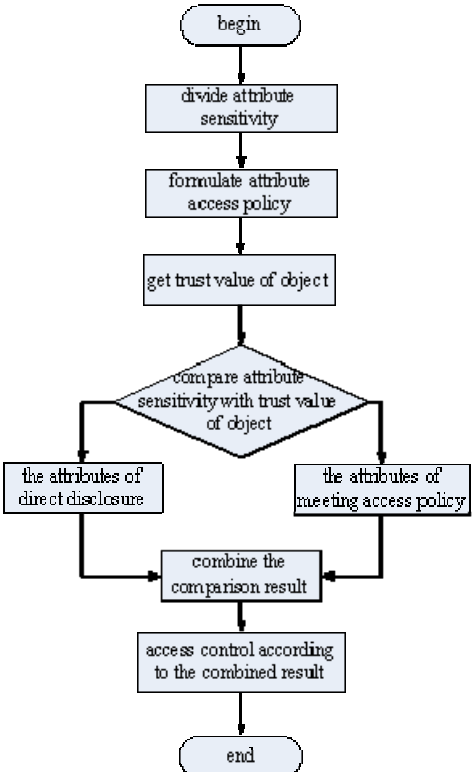

Fig. 1. Work flow of TSAP

2.2.4 Acquirement of Sensitive Attributes

When an object submits its request for subject attributes, the subject will make judgments based on the trust level of the object and the sensitivities of requested attributes. Then the subject will judge which attributes could be directly disclosed to the object, and which attributes can be disclosed to the object only when the following conditions are met:

If the sensitivity of a subject attribute is lower than or equal to the trust level of the target object, the subject will send this attribute to the target object directly or declares that it does not own certain attribute.

If the sensitivity of a subject attribute is larger than or equal to the trust level of the target object, the subject will send out attributes only when the object meets the corresponding Attribute Access Policies.

2.2.5 Access to Objects

After the above comparison judgments by subjects, ordinarily there will be two parts of results. The first part contains attributes that can be disclosed directly, and the other part contains attributes that can be disclosed indirectly after meeting the Attribute Access Policy. Subjects will merge the two parts together and send out the result to PDP (policy decision point) in ABAC. Then PDP will make authorization decisions and decide whether the subject is allowed to access the object or not.

3 Trust evaluation

Currently, there are many trust management models providing trust level calculation methods, such as Beth model, RFSN model and so on. In this section, we will first introduce the Beth model, as we are going to make a comparison between it and TSAP, and then we will describe the TSAP model in detail.

3.1 Beth Model

The Beth model adopts experience to describe and value trust, which mainly comes from history records of interactions between entities. When one interaction is successful, it is called a positive experience; otherwise it is called a negative experience. The experience can also be acquired from other entities' recommendations. Thus trust is divided into direct trust and recommended trust. The direct trust is formulated as follows:

$$V_i(p) = 1 - a^p \tag{1}$$

In Formula 1, p represents the number of successful interactive times, which also is the number of positive experiences, and a represents the expected value of successful interaction times from subjects to objects.

The recommended trust is formulated as follows:

$$V_r(p,n) = \begin{cases} 1 - a^{p-n}, if(p > n) \\ 0, else \end{cases} \quad (2)$$

In Formula 2, n means the number of failure interactions times while p and a means the same as above.

3.2 TSAP Model

Trust-based Sensitive Attributes Protection mechanism (TSAP) proposed in this paper divides trust into direct trust and recommended trust. TSAP can also exclude malicious recommendations, which is not taken into consideration in the Beth model.

3.2.1 Direct Trust

Direct Trust comes from historical access records of entities. Suppose each entity owns an access list which is composed of successful interactive number of times, and failure times. The list records interactions of one entity with other entities, as is shown in Table 1.

Table 1. Access list of entity m

	Entity1	Entity 2	...	Entity n
Success Times (a)	S_{m1}	S_{m2}	...	S_{mn}
Failure Times (b)	F_{m1}	F_{m2}	…	F_{mn}

S_{mn} means the number of successful interaction times between entity m and entity n, and F_{mn} means the number of failed interaction. Here, notice that S_{mn} is unequal to S_{nm}, and S_{mn} indicates that entity m sends request to entity n. Direct Trust of entity m to entity n can be seen as a subjective expectation for entity m that how entity n is able to fulfill a target task. In other words, it is a guess of probability that whether entity m can access entity n successfully. Each interaction between entity m and entity n can be seen as an independent event, and this event only has two results: success (represented by 1) or failure (represented by 0).

We use θ to represent the predication of next interaction (0 or 1) between entity m and entity n. According to the analysis and models mentioned in paper [10], we find that posterior probability of θ behaves according to Beta distribution (shown in Formula 3). Direct Trust of entity m to entity n can be represented by the expectation of θ (shown in Formula 4).

$$P(\theta) = \frac{Bin(S_{mn}+F_{mn},S_{mn}) * Beta(1,1)}{Normalization} = Beta(S_{mn}+1, F_{mn}+1) \quad (3)$$

$$T_D = E\big(P(\theta)\big) = E\big(Beta(S_{mn}+1, F_{mn+1})\big) \quad (4)$$

3.2.2 Recommended Trust

Trust evaluation of entity m to entity n is not accurate enough only from the subjective evaluation of entity m. It also needs recommendations provided by a trusted third

party. Recommended Trust is from historical evaluation records between entities. Suppose that the third party owns a number of evaluation lists which record the evaluative value of target entities after interactions. As is shown in Table 2, where V_{kn} represents the evaluative value of entity k to entity n, and $V_{kn} \in [0,1]$.

Table 2. Evaluation list of entity n

Entity	1	2	...	k
Evaluative Value (V)	V_{1n}	V_{2n}	...	V_{kn}

However, there may be some malicious evaluations from some entities, such as the competitors of homogeneous services, network viruses and so on. Therefore, to get effective recommended trust, we should eliminate entities with malicious recommendations first, and then adopt evaluative values of honest entities.

Here, we use deviation degree of each entity's evaluative value to exclude entities' malicious recommendations. Deviation degree shows the deviation extent of a single entity's evaluative value to a real value. Firstly, we calculate the average value (AV) of all evaluative values about the target entity given by other entities, as is shown in Formula 5. Secondly, certain deviation degree (Δ) is set by the trusted third party. Then we get a D-value of a single entity by comparing its evaluative value with the average value of all entities, and we use DV_k to represent the D-value of entity k, as is shown in Formula 6. Finally, we compare DV_k with Δ. If DV_k is beyond Δ, we exclude the evaluative value of this entity. Otherwise the entity will be included in a set R of entities providing honest evaluation.

$$AV(V_{kn}) = \frac{\sum V_{kn}}{N} \text{ , where } k \in \{\text{all entities that have made evaluation on entity } n\} \quad (5)$$

$$DV_k = |V_{kn} - AV|, \text{ where } k \in \{\text{all entities that have made evaluation on entity } n\} \quad (6)$$

Meanwhile, in order to avoid malicious evaluations as much as possible, we use honest level to represent the real extent of evaluations about target objects. Here, the honest level of an entity is obtained from the trusted third party who saves an honesty list for all entities that have made their evaluations, as shown in Table 3.

Table 3. Honesty list of entities

	Entity 1	Entity 2	...	Entity n
Real Evaluative Times (HN)	HN_1	HN_2	...	HN_n
Total Evaluative Times (SN)	SN_1	SN_2	…	SN_n

After entity k making evaluation to the object, the total number of evaluation (SN_k) correspondingly adds 1. If this evaluation does not belong to malicious evaluations, the number of real evaluations (HN_k) also needs to add 1. Honest level of entity k (H_k)

is represented by the proportion of the number of real evaluations in the total number of evaluations, as shown in Formula 7.

$$H_k = HN_k / SN_k \tag{7}$$

Through effectively excluding malicious recommendation from entities and getting honest level of various entities, the recommended trust (T_R) can be obtained by the calculation of Formula 8, where V_{kn} represents the evaluative value of entity k to entity n, H_k shows the honest level of entity k, R denotes a set of entities that give honest evaluation of entity n, and N means the number of honest entities.

$$T_R = \frac{\sum H_k * V_{kn}}{N} \tag{8}$$

3.2.3 Comprehensive Trust Evaluation

Trust evaluation of entity m to entity n is composed of direct trust (T_D) and recommended trust (T_R), as shown in Formula 9, where β represents the subjective evaluation's trust extent of entity m, and $\beta \in [0,1]$. If entity m has more faith in its own subjective evaluation, β takes a higher value; otherwise β takes a smaller value.

$$T = \beta T_D + (1 - \beta) T_R \tag{9}$$

4 Case analysis and Experiment Comparison

In this section, we use a simplified recommendation relationship graph as follows to describe the trust relationship between two entities i and j.

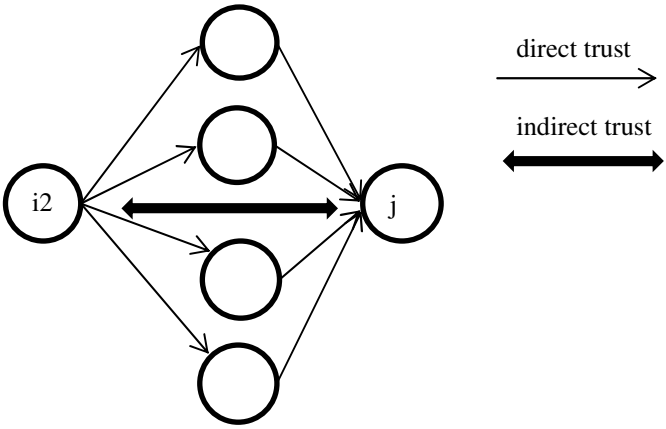

Fig. 2. The trust relationships between entities

As we can see from Fig. 2, entity i trusts entity j indirectly and the other entities directly. The other entities offer one-path recommendations of j.

Suppose that entity i has attributes including {name, age, date of birth, ID number, family address, telephone, marital status, hobbies, work unit, medical history}, and the corresponding sensitivities respectively are {0.25, 0.18, 0.20, 0.80, 0.50, 0.40,

0.20, 0.35, 0.50, 0.90}. Subject formulates the same Attribute Access Policies for ID number, work unit, medical history which have higher sensitivities. For example:

If Security grade= high and Certificate issuer = country institution;
imply disclosure ID number;

In our case, entity *i* needs to access object *j*, and entity *i* gets that the successful interactive number between entity *i* and entity *j* is 29 and the failed interactive number is 9 based on the history access record of entity *i* to entity *j*, so Direct Trust (T_D) of entity *i* to entity *j* is 0.75 according to Formula 2. The evaluation list of entity *j* provided by the trusted third party is shown in Table 4, and the value range of deviation degree (Δ) is set as[0,0.25]. We can obtain that the average value (AV) of all evaluative values is 0.57 based on the evaluation list of entity *j* and Formula 3. The D-values (DV) of various entities are shown in Table 5 based on Formula 4. Through comparing DV of every entity with Δ, we exclude evaluative value of entity 5 and entity 8. Honest level (H) of other entities can respectively be calculated to get {0.40, 0.50, 1.0, 0.75, 0.85, 0.73, 0.80, 0.72} via Table 6 provided by the trusted third party. Recommended Trust (T_R) is 0.47 based on Formula 6.

Suppose that entity *i* has more faith in its own subjective evaluation on entity *j*. Therefore, when calculating comprehensive trust of entity *i* on entity *j*, β takes a higher value, say 0.70. Finally, the comprehensive trust value (T) of entity *i* on entity *j* is 0.667 calculated by Formula 7.

Through comparing attribute sensitivities of entity *i* with comprehensive trust value of entity *i* on entity *j*, the attributes which can be disclosed to entity *j* are {name, age, date of birth, family address, telephone, marital status, hobbies, work unit}. If the attribute values of Security_grade of entity *j* is high and the attribute values of Certificate_issuer of entity *j* is country institution, entity *j* will get the attributes of ID number and medical history of entity *i* at the same time.

Table 4. Evaluation list of entity j

Entity	1	2	3	4	5	6	7	8	9	10
V	0.70	0.50	0.60	0.80	0.20	0.60	0.70	0.30	0.80	0.50

Table 5. D-value list of entities

Entity	1	2	3	4	5	6	7	8	9	10
DV	0.13	0.07	0.03	0.23	0.37	0.03	0.13	0.27	0.23	0.07

Table 6. Honesty list of entities

Entity	1	2	3	4	6	7	9	10
HN	20	15	20	30	17	44	32	54
SN	50	30	20	40	20	60	40	75

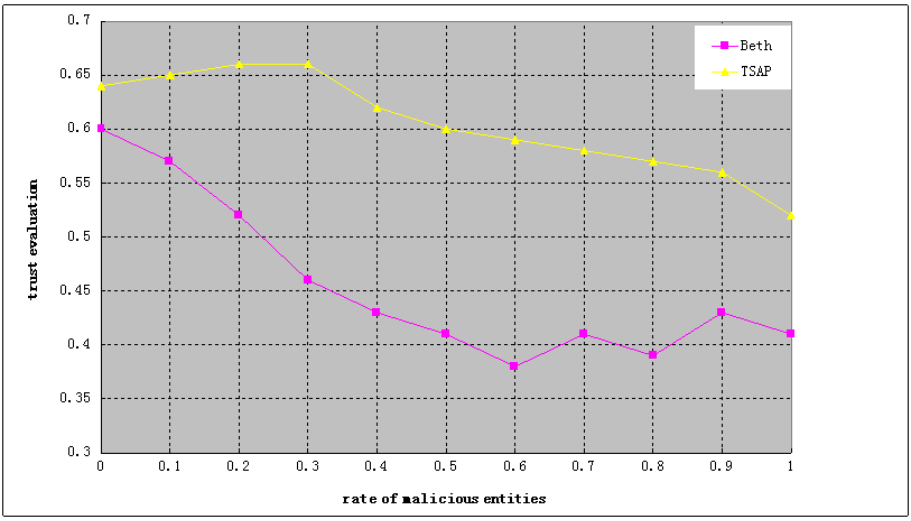

Fig. 3. Result of the first experiment

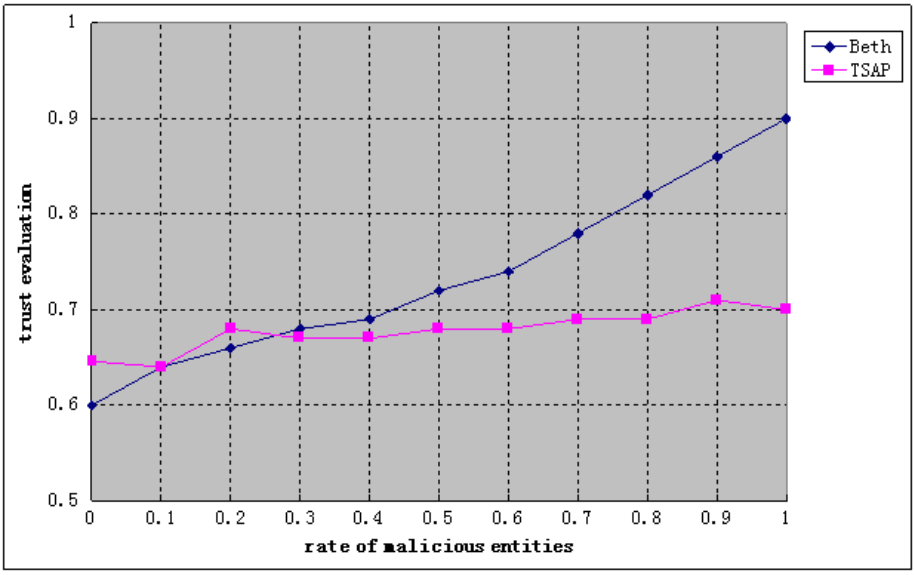

Fig. 4. Result of the second experiment

Here we perform two groups of experiments. In the first group, each malicious entity gives *i* a much lower trust value than the real value. The result is shown in Fig.3. As we can see from it, the trust valuation of *j* to *i* becomes lower as the rate of malicious entities increases both in Beth model and our TSAP model. However, the result of trust evaluation in TSAP model is always between 0.5 and 0.7, thus it has less impacts on which specific attributes to disclose as we describe in the example in last

section while the Beth model makes some attribute with lower sensitivity still cannot be disclosed to entity *j*.

In the second group, each malicious entity gives *i* a much higher value. As we can see from Fig.4, the result in Beth model changes a lot as the rate of malicious entities increases. It can even reach the value of 0.9, and this means that there is no sensitive attribute at all, thus *j* can get all the attributes of *i*. However, the result of TSAP model does not change much and the attributes to disclose stay the same as in the example in last section.

By comparing the results of the two experiments, we can see that the TSAP model has higher accuracy than the Beth model especially where there are malicious entities offering recommendations. And the TSAP model itself can well bear the changes resulting from the malicious entities which give unusual recommended trust values.

5 Related Work

For the protection of sensitive attributes, Seamons et al. proposed an idea on considering both content-sensitivity of attributes (attribute value) and possession sensitivity of attributes (whether there are certain attribute or not). One can predicate whether the others possess certain attributes by their different behavior with different attributes [3].

Holt et al. proposed hidden credentials based on the principle of elliptic curve encryption, to protect credentials from attacking and prevent leaking sensitive information via setting system parameter [4]. Oblivious signature-based envelope (OSBE) proposed in Paper [6] can avoid the malicious disclosure of attribute information by means of digital signature of attributes. Paper [7] made use of zero knowledge protocol between subjects and objects in which both sides can obtain different information by inputting different attribute value, and both sides cannot know any attribute information about each other. Although plenty of papers presented effective methods to protect sensitive attributes, these methods have fussy encryption or decryption process, which cost a lot of computing time and memory space, and go against some resource limited equipments.

Winsborough et al. proposed an acknowledgement policy (ACK) aiming at possession-sensitivity of attributes [8]. But this method has a problem of abundant workload management and a coarse-grained protection. In paper [9], attribute owners establish the corresponding access control policies for sensitive attributes, and send the access control policies to policy database anonymously. This mechanism effectively protects possession-sensitivity of attribute, while it fails to consider the network environment of system, and may destroy the flexibility of system access. Thus attribute owners may also disclose some sensitive attributes directly in a safe environment.

Through the analysis of existing works, this paper presents a trust based sensitive attributes protection mechanism (TSAP). TSAP mechanism can protect both content-sensitivity and possession-sensitivity of attributes, avoid fussy encryption or decryption process, and flexibly protect sensitive attributes of subjects in different network environments.

6 Conclusions and Future Work

Although the ABAC model can solve the resource sharing and collaboration problems of distributed systems, it can easily cause disclosures of private information. This paper proposes a trust based sensitive attributes protection mechanism. This mechanism discloses sensitive attributes based on attribute sensitivities of a subject and the trust level of an object. It can protect both content-sensitivity and possession-sensitivity of attributes. And an object can independently decide to disclose sensitive attributes, increasing the flexibility of system access.

In the future, we will integrate TSAP mechanism into ABAC access control architecture, in order to provide more secure access control, and will verify the effectiveness of the whole access control architecture at the same time.

References

1. Eric, Y., Jin, T.: Attributed Based Access Control (ABAC) for Web Services. In: Proceedings of the IEEE International Conference on Web Services (ICWS 2005), pp. 560–569 (2005)
2. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. LNCS, vol. 875, pp. 1–18 (1994)
3. Seamons, K.E., Winslett, M., Yu, T., Yu, L., Jarvis, R.: Protecting Privacy during On-Line Trust Negotiation. In: Dingledine, R., Syverson, P.F. (eds.) PET 2002. LNCS, vol. 2482, pp. 129–143. Springer, Heidelberg (2003)
4. Holt, J.E., Bradshaw, R.W., Seamons, K.E., Orman, H.: Hidden credentials. In: Proceedings of the ACM Workshop on Privacy in the Electronic Society, pp. 1–8 (2003)
5. Bradshaw, R., Holt, J., Seamons, K.E.: Concealing complex policies with hidden credentials. In: Proceedings of the 11th ACM Conference on Computer and Communications Security, pp. 146–157 (2004)
6. Li, N.H., Du, W.L., Boneh, D.: Oblivious signature-based envelope. In: Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003), pp. 182–189 (2003)
7. Li, J., Li, N.: OACerts: Oblivious Attribute Certificates. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 301–317. Springer, Heidelberg (2005)
8. Winsborough, W.H., Li, N.H.: Protecting sensitive attributes in automated trust negotiation. In: Proceedings of the ACM Workshop on Privacy in the Electronic Society, pp. 41–51 (2002)
9. Irwin, K., Yu, T.: Preventing Attribute Information Leakage in Automated Trust Negotiation. In: Proceedings of the 12th ACM Conference on Computer and Communications Security, pp. 41–51 (2005)
10. Sang, A.: A Subjective Metric of Authentication. In: Proceedings of European Symposium on Research in Security, pp. 329–344 (1998)
11. Yu, T., Winslett, M.: Policy migration for sensitive credentials in trust negotiation. In: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society (WPES 2003), pp. 9–20 (2003)
12. Gevers, S., De Decker, B.: Privacy Friendly Information Disclosure. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 636–646. Springer, Heidelberg (2006)

13. Esmaeeli, A., Shahriari, H.R.: Privacy Protection of Grid Service Requesters through Distributed Attribute Based Access Control Model. In: Bellavista, P., Chang, R.-S., Chao, H.-C., Lin, S.-F., Sloot, P.M.A. (eds.) GPC 2010. LNCS, vol. 6104, pp. 573–582. Springer, Heidelberg (2010)
14. Kolter, J., Schillinger, R., Pernul, G.: A Privacy-Enhanced Attribute-Based Access Control System. In: Barker, S., Ahn, G.-J. (eds.) Data and Applications Security 2007. LNCS, vol. 4602, pp. 129–143. Springer, Heidelberg (2007)
15. EI-Khatib, K.: A Privacy Negotiation Protocol for Web Services. In: Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments Halifax (October 13, 2003)
16. Guajardo, J., Mennink, B., Schoenmakers, B.: Anonymous Credential Schemes with Encrypted Attributes. In: Heng, S.-H., Wright, R.N., Goi, B.-M. (eds.) CANS 2010. LNCS, vol. 6467, pp. 314–333. Springer, Heidelberg (2010)