

Agent based automated trust negotiation model

Li Kai Lu Zhengding Li Ruixuan Tang Zhuo

(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: To enhance the practicability of the trust negotiation system, an agent based automated trust negotiation model (ABAM) is proposed. The ABAM introduces an agent to keep the negotiation process with no human intervention. Meanwhile, the ABAM specifies the format of a meta access control policy, and adopts credentials with flexible format to meet the requirements of access control policies instead of disclosing the whole contents of a certificate. Furthermore, the ABAM uses asymmetric functions with a high security intensity to encrypt the transmitting message, which can prevent information from being attacked. Finally, the ABAM presents a new negotiation protocol to guide the negotiation process. A use case is studied to illuminate that the ABAM is sound and reasonable. Compared with the existing work, the intelligence, privacy and negotiation efficiency are improved in the ABAM.

Key words: automated trust negotiation; agent; credential; access control policy; negotiation protocol

As computer systems become more and more interconnected, many situations arise where different systems need to share data or resources^[1-3]. For example, a provider who wants to supply online service over the web must decide how much a remote user with a certain set of credentials is to be trusted.

Exchange of attribute credentials is a means to establish mutual trust relationship between strangers who wish to share resources or conduct business transactions. Automated trust negotiation (ATN) is invented as an effective method to regulate the exchange of sensitive information during such a process^[4-6].

While ATN provides a good means for the negotiators to share resources, it still has some pitfalls waiting to be solved. First, in order to establish the trust relationship during the negotiation process, the negotiators' certificates should be disclosed to each other. However, if a certificate contains sensitive information or individual privacy, the disclosure of a certificate will lead to information leakage^[5-6]. Secondly, during the negotiation process, the user and the resource provider are both required to take a real time part in the process, which degrades negotiation efficiency and causes a great waste in manpower. Thirdly, an access control policy itself may also contain sensitive information. The unconditional disclosure of policy contents may leak valuable business information or jeopardize

individual privacy^[7-9]. Fourthly, it is time-consuming and inefficient to check whether the credentials meet the requirements of the access control policy. Usually, in order to prevent information leakage, the negotiation protocol is designed too complicatedly for a requestor to attain access. Finally, if the encryption algorithm is limited in security intensity, the transmitting message over the insecure channel is easy to be attacked.

To solve the problems, an agent based automated trust negotiation model (ABAM) is proposed in this paper. The ABAM introduces an agent to ATN, which can make ATN much more intelligent. And the ABAM specifies the access control policy with a flexible format, which can reduce the size of the meta policy and help to relieve the burden towards networks. Thereto, the ABAM provides functions to read specific attribute values, using a flexible credential format to issue the credentials, which can avoid releasing the whole contents of a certificate to the other party. In addition, the ABAM adopts asymmetric algorithms to encrypt the transmitting message, which can effectively prevent the negotiation from being attacked.

1 Agent Based ATN Model

1.1 Assumption

In the ABAM, we give the basic assumption as follows:

① Cred: credential set, $\text{Cred} = \{c_1, c_2, \dots, c_n\}$, c_i ($1 \leq i \leq n$) is a credential. A credential is a kind of temporary certificate, whose content comes from the user's identity certificate.

② Policy: access control policy set, $\text{Policy} = \{p_1,$

Received 2007-05-18.

Foundation item: The National Natural Science Foundation of China (No. 60403027).

Biographies: Li Kai (1968—), male, graduate; Lu Zhengding (corresponding author), male, professor, zdlu@hust.edu.cn.

$p_2, \dots, p_n\}$, $p_i (1 \leq i \leq n)$ is a meta policy. The meta policy is treated as the public key to encrypt the protected resource. Respectively, the corresponding credential is regarded as the private key to decrypt the encrypted resource.

Definition 1 (negotiation strategy) The negotiation strategy specifies how the credentials and access control policies are released. In the ABAM, the resource is protected by being encrypted by the access control policies. If a credential can decrypt the resource cipher text, we say that the credential is disclosed. As far as the resource is concerned, we say the resource is revealed if its cipher text is decrypted.

Definition 2 (negotiation protocol) Negotiation protocol specifies what actions will be done during the negotiation process. In the ABAM, the negotiation process includes the following steps: ① Exchanging identity ID; ② Sending the access request; ③ Getting the encrypted resource; ④ Playing operations towards the resource.

1.2 Architecture of ABAM

The ABAM introduces an agent to make the trust negotiation intelligent. Fig. 1 shows the architecture of the ABAM. The ABAM consists of the following components.

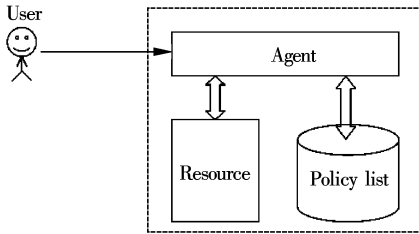


Fig. 1 Architecture of ABAM

- **Agent** The component provides online and automatic negotiation service. Its functions include receiving the access request from the user, filtering the invalid access request by checking the requested operations, getting the access control policy from the policy list according to the valid requested operations, and redirecting the access to the resource if the user satisfies the requirements of the access control policy.

- **Policy list** The component generates and stores the access control policies. Its functions include providing a retrieving service for the agent to get the access control policy, providing an interface for the resource provider to edit the access control policy, and storing the access control policies, i. e., to maintain an access control policy list according to the resource type. The detailed access control policy list will be described below.

- **Resource** The component is the main part to serve the legitimate users.

1.3 Access control policy

In order to make the negotiation process much more intelligent, we redefine the format of a meta policy as follows:

$$\text{Policy} = (\text{subject}) : (\text{recipient}) : (\text{attribute}) : (\text{operator}) : (\text{value})$$

In the format description, the first item subject is the owner of the policy, i. e., the resource provider's identity ID. The second item recipient specifies the policy's receiver, i. e., the user's identity ID. The third item attribute stands for the attribute type, such as role, age, and so on. The fourth item operator is the comparing predicate. Usually, $\text{operator} \in \{\supseteq, \subseteq, \in, \notin, >, <, =, \leq, \geq, \neq\}$. The fifth item value is the required attribute value. For example, a meta policy $p = (\text{Alice}) : (\text{Tom}) : (\text{role}) : (\in) : (\{\text{PhD}, \text{Ms}, \text{graduate}\})$ means that the resource provider Alice sends the policy to Tom, she requires that Tom's role belongs to {PhD, Ms, graduate}.

Usually, a complex access control policy consists of many meta policies. In the ABAM, we provide three predicates to combine meta policies into a complex one.

- \wedge : Connective predicate. For instance, $p_1 \wedge p_2$ means that the credentials should meet the requirements of both p_1 and p_2 .

- \vee : Selective predicate. For instance, $p_1 \vee p_2$ means that the credentials need only to meet the requirement of p_1 or p_2 .

- \neg : Negative predicate. $\neg p$ means the opposition of p , i. e., to negate the operator item of p . For instance, $p = (\text{Alice}) : (\text{Tom}) : (\text{age}) : (>) : (20)$, $\neg p = (\text{Alice}) : (\text{Tom}) : (\text{age}) : (\leq) : (20)$.

In the ABAM, we propose a function `policyGenerate()` to generate a meta policy. The function is described as follows:

- `policyGenerate(subject, recipient, attribute, operator, value)`: to generate a meta policy according to the policy format. The parameters subject, attribute, operator and value are set by the resource provider, and the parameter recipient comes from the user by exchanging the identity ID.

In the ABAM, all the access control policies are stored in the components of the policy list. Since the resource provider can supply different services for the user according to the user's attributes or identity, the different operation towards the resource requires the user to have different attributes.

1.4 Credential

A credential is a digital vehicle to carry all kinds of attribute information. In the ABAM, we provide a flexible format as follows:

cred = (holder): (recipient): (attribute): (value): (period)

In the format, the first item, holder, is the credential owner. The second item, recipient, specifies the credential's recipient, i. e., the resource provider. The third item, attribute, is the attribute type, such as major, role, age and so on. The fourth item, value, is the specified attribute value. The last item, period, stands for the life time of the credential, i. e., the valid period. Since every credential has its own life time, and the valid period is not long, we treat it as default. Usually, the life time is set as 24 h. For instance, the credential $c = (\text{Tom}): (\text{Alice}): (\text{role}): (\text{PhD}): (2007-02-01, 2007-02-02)$ means that Tom is the credential holder, his role is PhD, the credential is used to meet the access control policy from Alice, and the credential is valid during the period from 2007-02-01 to 2007-02-02.

Accordingly, we provide two functions to accomplish a credential's generation. The two functions are as follows:

- $\text{attrRead}(\text{attribute}, \text{cert})$: to get the attribute value from the identity certificate according to the attribute type. The parameter attribute is the attribute type, while the other cert is the certificate. For instance, $\text{attrRead}(\text{role}, \text{cert}_{\text{Tom}})$ means reading the role information from the certificate cert_{Tom} .

- $\text{credGenerate}(\text{holder}, \text{recipient}, \text{attribute}, \text{value})$: to generate a credential. The parameter holder is the user's identity ID. The parameter recipient is the resource provider's identity ID, which can be obtained from the identity exchange. The parameter attribute comes from the access control policy. The parameter value is attained by running the function $\text{attrRead}()$. Note that, the life time of a credential is automatically added according to the current time.

1.5 Negotiation protocol

Negotiation protocol specifies what the user and the resource provider should do during the negotiation process. In the ABAM, the negotiation process consists of three states: ① To exchange the identity IDs between the user and the resource provider; ② To send the access request and get the encrypted resource; ③ To decrypt the resource cipher text and access the resource. Fig. 2 shows the negotiation process of the ABAM.

Note that all the transmitting messages are cipher

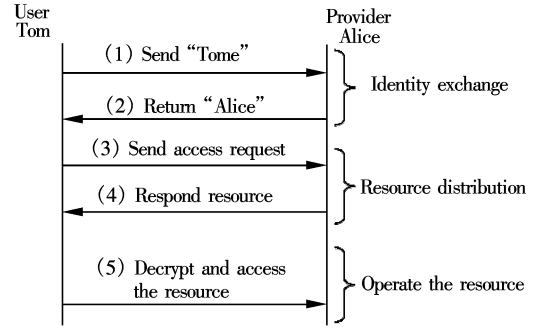


Fig. 2 The negotiation process of the ABAM

texts so as to protect the communication information from being attacked. In the ABAM, we provide a series of functions to accomplish the encryption and decryption as follows:

- $\text{Encrypt}(k, \text{msg})$: to encrypt msg using k . In the ABAM, Encrypt is an asymmetric function. Thus, the encryption cipher k_{en} is different from the decryption cipher k_{de} , and $k_{\text{en}} = k_{\text{de}}^{-1}$.

- $\text{Decrypt}(k, \text{msg})$: to decrypt msg using k . Decrypt and Encrypt satisfies: $\text{Decrypt}(A, \text{Encrypt}(B, \text{msg})) = \text{msg}$ ($A = B^{-1}$, or A and B are the public key and private key).

- $\text{CR} = \text{msg}_E(R, \text{recipient}, P)$: to use policy P to encrypt the resource R whose receiver is recipient. The resource R is protected by policy P . If $P = p_1 \wedge p_2$, $\text{CR} = \text{msg}_E(R, \text{recipient}, P) = \text{msg}_E(\text{msg}_E(R, \text{recipient}, p_1), \text{recipient}, p_2)$. If $P = p_1 \vee p_2$, $\text{CR} = \text{msg}_E(R, \text{recipient}, P) = \{\text{CR} = \text{msg}_E(R, \text{recipient}, p_1), \text{CR} = \text{msg}_E(R, \text{recipient}, p_2)\}$.

- $R = \text{msg}_D(\text{CR}, \text{Cred})$: to decrypt the cipher text CR using user's credential set Cred. If the credentials can meet the requirements of the policy P , CR will be recovered into R . To check whether a credential can match a meta policy, the process includes: ① To check the credential's life time so as to judge its validity; ② To check whether the credential's holder equals the policy's recipient; ③ To check whether the credential's attribute type equals the policy's attribute; ④ To check whether the attribute value meets the requirement of the policy's operator.

2 A Use Case for ABAM

2.1 Scenario description

Suppose Lily to be a computing center, which can provide storage service and online cluster computing service. If a user wants to use the storage service, he is required to be a member of Lily or a student of the ABC College. If a user wants to have the online cluster computing, he is required to meet: ① He is a member

of Lily and he majors in a scientific project; or ② He is a graduate of the ABC College and a team leader; or ③ He is a graduate of the ABC College and majors in a scientific project.

2.2 Policy description and policy list

Based on the above scenario, the access control policy can be depicted as

$$\begin{aligned} \text{Policy}_{\text{storage}} &= (p_1 \wedge p_2) \vee (p_3 \wedge p_4) \\ \text{Policy}_{\text{computing}} &= (p_1 \wedge p_2 \wedge p_7) \vee (p_4 \wedge p_5 \\ &\quad \wedge p_6) \vee (p_4 \wedge p_5 \wedge p_7) \end{aligned}$$

In the above, the meta policies are: $p_1 = \{\text{user. role} = \text{employee}\}$, $p_2 = \{\text{user. unit} = \text{Lily}\}$, $p_3 = \{\text{user. role} = \text{student}\}$, $p_4 = \{\text{user. unit} = \text{ABC}\}$, $p_5 = \{\text{user. role} = \text{graduate}\}$, $p_6 = \{\text{user. role} = \text{teamleader}\}$, $p_7 = \{\text{project. type} = \text{science}\}$.

Accordingly, the policy list can be described as in Tab. 1.

Tab.1 The policy list of the use case

Service type	Access control policy
Access	None
Storage	$(p_1 \wedge p_2) \vee (p_3 \wedge p_4)$
Cluster computing	$(p_1 \wedge p_2 \wedge p_7) \vee (p_4 \wedge p_5 \wedge p_6) \vee (p_4 \wedge p_5 \wedge p_7)$

2.3 Credentials description

Suppose Tom to be a graduate of the ABC College. In order to test an algorithm, he wants to deploy it in a cluster and have a cluster computing. That is to say, after getting the resource provider's identity ID, Tom can hold two credentials: c_A and c_B , while $c_A = (\text{Tom}) : (\text{Lily}) : (\text{role}) : (\text{graduate}) : (2007-02-01, 2007-02-02)$ and $c_B = (\text{Tom}) : (\text{Lily}) : (\text{project}) : (\text{science}) : (2007-02-01, 2007-02-02)$.

2.4 Negotiation process

In the use case, the negotiation process includes the following steps.

① Tom, as the requestor, encrypts his identity ID by using the resource provider's public key, i. e., $\text{encrypt}(\text{Pub}_{\text{Lily}}, \text{"Tom"})$. Then, Tom sends the cipher text to the resource provider.

② The agent receives the cipher text, and uses his own private key to decrypt it ($\text{decrypt}(\text{Sec}_{\text{Lily}}, \text{encrypt}(\text{Pub}_{\text{Lily}}, \text{"Tom"}))$) and recovers "Tom". Then, the agent encrypts his identity ID by using the Tom's public key (that is to say, $\text{encrypt}(\text{Pub}_{\text{Tom}}, \text{"Lily"})$) and sends the cipher text back.

③ Tom decrypts the cipher text ($\text{decrypt}(\text{Sec}_{\text{Tom}}, \text{encrypt}(\text{Pub}_{\text{Tom}}, \text{"Lily"}))$) and recovers "Lily". Then, Tom encrypts his access request and sends $\text{req} = \text{encrypt}(\text{Pub}_{\text{Lily}}, \text{"cluster computing to test an algorithm"})$ to the agent.

④ The agent decrypts the encrypted access re-

quest and checks whether the request is invalid. After checking the operation, the agent judges the request is valid. Then, the agent retrieves the policy list and gets the access control policy. Meanwhile, the agent generates an access entry, such as "The login ID is XXX, password is XXXX, and the port is XXXX", which is treated as the resource R . Now, the agent uses $\text{msg}_E()$ to encrypt R ($\text{msg}_E(R, \text{"Tom"}, \text{Policy}_{\text{computing}})$) and $\text{encrypt}()$ encrypt the policy ($\text{encrypt}(\text{Pub}_{\text{Tom}}, \text{Policy}_{\text{computing}})$). Finally, the agent sends the two cipher texts together to Tom.

⑤ Tom receives the two cipher texts. He uses his private key to recover the access control policy. According to the policy, he uses his certificate to generate the corresponding credentials. Then, he uses the credentials to decrypt the encrypted resource. After that, Tom can log on the resource server and execute his testing.

3 Discussion and Conclusion

In this paper, an agent-based ABAM is presented. The features of the ABAM are summarized into five aspects as follows: ① A new method to make the negotiation process intelligent; ② A new negotiation strategy to disclose credentials and access control policies; ③ A new negotiation protocol to guide the negotiation process; ④ A new means to prevent information leakage; ⑤ A flexible format for the policy and a corresponding format for the credential.

The ABAM introduces an agent to make the negotiation process much more intelligent. Thereto, the ABAM specifies the format of a meta policy, and adopts credentials with flexible format to meet the requirements of access control policies, instead of disclosing the whole contents of a certificate. In addition, the ABAM uses asymmetric functions with a high security intensity to encrypt the transmitting message, which can prevent information from being attacked. Furthermore, the ABAM presents a new negotiation strategy to enhance the negotiation efficiency. A use case is studied to illuminate that the ABAM is a sound and reasonable model for ATN.

References

- [1] Liao Zhensong, Jin Hai, Li Chisong, et al. Automated trust negotiation and its development trend[J]. *Journal of Software*, 2006, 17(9): 1933 – 1948. (in Chinese)
- [2] Winsborough W H, Li N. Towards practical automated trust negotiation[C]//*Proc of the 3rd International Workshop on Policies for Distributed Systems and Networks*. Los Alamitos: IEEE Computer Society, 2002: 92 – 103.

- [3] Winsborough W H, Seamons K E, Jones V E. Automated trust negotiation [C]//*DARPA Information Survivability Conference and Exposition*. Piscataway: IEEE Press, 2000: 88 – 102.
- [4] Jin H, Liao Z S, Zou D Q, et al. A new approach to hide policy for automated trust negotiation[C]//*Proc of the 1st International Workshop on Security*. Berlin: Springer-Verlag, 2006: 168 – 178.
- [5] Winsborough W H, Li N. Protecting sensitive attributes in automated trust negotiation[C]//*Proc of ACM Workshop on Privacy in the Electronic Society*. New York: ACM Press, 2002: 102 – 113.
- [6] Yu T, Winslett M. A unified scheme for resource protection in automated trust negotiation [C]//*IEEE Symposium on Security and Privacy*. Los Alamitos: IEEE Computer Society, 2003: 245 – 257.
- [7] Winsborough W H, Li N. Safety in automated trust negotiation[C]//*Proc of the IEEE Symposium on Security and Privacy*. Los Alamitos: IEEE Computer Society, 2004: 147 – 160.
- [8] Yu T. Dynamic trust establishment in open systems[D]. Urbana-Champaign: Department of Computer Science of University of Illinois, 2003.
- [9] Seamons K, Winslett M, Yu T. Limiting the disclosure of access control policies during automated trust negotiation [C]//*Proceedings of the Network and Distributed System Security Symposium*. San Diego, CA, 2001: 45 – 56.

一种基于代理的自动信任协商模型

李 开 卢正鼎 李瑞轩 唐 卓

(华中科技大学计算机科学与技术学院, 武汉 430074)

摘要:为增强信任协商系统的实用性,提出一种基于代理的自动信任协商模型(ABAM). ABAM 引进代理使协商过程免于人工干涉. 同时,ABAM 指定了元策略格式,使用这种格式灵活的信任凭证来满足访问控制策略要求,而不需披露数字证书的全部内容. 此外,ABAM 使用高强度的非对称函数加密传输消息,能防止消息遭受攻击. 最后,模型中提出一种新的协商协议来指导协商进行. 事例分析表明,ABAM 是健全和合理的. 与现有工作相比,ABAM 在智能性、保密性和协商效率方面得到了改进.

关键词:自动信任协商;代理;信任凭证;访问控制策略;协商协议

中图分类号:TP311