1-1-2010

Dynamic trust model for federated identity management

Hao Gao

University of Wollongong, hg016@uow.edu.au

Jun Yan

University of Wollongong, jyan@uow.edu.au

Yi Mu

University of Wollongong, ymu@uow.edu.au

Follow this and additional works at: <https://ro.uow.edu.au/infopapers>

Part of the [Physical Sciences and Mathematics Commons](#)

Recommended Citation

Gao, Hao; Yan, Jun; and Mu, Yi: Dynamic trust model for federated identity management 2010, 55-61.
<https://ro.uow.edu.au/infopapers/1734>

Dynamic trust model for federated identity management

Abstract

The goal of federated identity management is to allow principals, such as identities and attributes, to be shared across trust boundaries based on established policies. Since current Single Sign-On (SSO) mechanism excessively relies on the specifications of Circle of Trust (CoT), the need for service collaboration from different domains is being addressed on CoT. For the motivating issue of the cross-domain SSO mechanism, we need an emergent dynamic trust list for calculating the trust parties, thus, the CoT specifications require an initial effort on enrolling members automatically to adapt to the dynamic open environment. In this paper, we propose a Dynamic Trust Policy Language to support trust negotiation. The formal syntax of this language is presented in Backus Naur Form (BNF) based on the concept of role membership. We also systematically develop the Dynamic Trust Model (DTM) to allow Untrusted SP to join the existing CoT by trust negotiation. Finally, we identify the process and algorithm for communication between negotiation entities.

Keywords

Dynamic, trust, model, for, federated, identity, management

Disciplines

Physical Sciences and Mathematics

Publication Details

Gao, H., Yan, J. & Mu, Y. (2010). Dynamic trust model for federated identity management. 4th International Conference on Network and System Security, NSS 2010 (pp. 55-61). Piscataway, New Jersey, USA: IEEE.

Dynamic Trust Model for Federated Identity Management

Hao Gao¹, Jun Yan¹ and Yi Mu²

¹*School of Information Systems and Technology*

²*School of Computer Science and Software Engineering*

University of Wollongong

Wollongong, AUSTRALIA

{hg016, jyan, ymu}@uow.edu.au

Abstract—The goal of federated identity management is to allow principals, such as identities and attributes, to be shared across trust boundaries based on established policies. Since current Single Sign-On (SSO) mechanism excessively relies on the specifications of Circle of Trust (CoT), the need for service collaboration from different domains is being addressed on CoT. For the motivating issue of the cross-domain SSO mechanism, we need an emergent dynamic trust list for calculating the trust parties, thus, the CoT specifications require an initial effort on enrolling members automatically to adapt to the dynamic open environment. In this paper, we propose a Dynamic Trust Policy Language to support trust negotiation. The formal syntax of this language is presented in Backus Naur Form (BNF) based on the concept of role membership. We also systematically develop the Dynamic Trust Model (DTM) to allow Untrusted SP to join the existing CoT by trust negotiation. Finally, we identify the process and algorithm for communication between negotiation entities.

Keywords—Federated Identity Management, Dynamic Trust, Single Sign-On, Circle of Trust

I. INTRODUCTION

Federation has become an emergent concept for current identity management. The goal of federated identity management is to allow principals, such as identities and attributes, to be shared across trust boundaries based on established policies. The identity federation enables users of one security domain to gain access to resources seamlessly from other domains without repeating login processes. A number of different frameworks and approaches were presented for federated identity management, which could be considered as formal Internet standards or openly published specifications, such as *OASIS SAML*, *OpenID*, *ID-FF* (Liberty Alliance Identity Federation Framework), *Shibboleth*, and *WS-Federation* [3], [4], [5], [6], [7]. Three main actors in a federation are defined as follows.

- Identity Providers (IdP) : entities that focus on authorization and authentication of users, as well as managing and sharing identity information with various trusted SPs.
- Service Providers (SP): entities that provide particular services to users who are authenticated by trusted IdPs.
- Users: principals who are willing to obtain services from multiple SPs.

Single Sign-On (SSO) mechanism is a popular approach to reduce repeated login challenges. SSO protocols enable users to sign in a federated environment once and yet be able to access several services offered by different SPs. While SSO service hardly poses any difficulty within a single administrative domain, there are several problems in cross-domain scenarios. Firstly, what kind of services could be trusted? Secondly, how could a service provider trust identity information from another administrator? Currently, two entities could establish trust relationship only before the interactions take place, which means the *Circle of Trust* (CoT) should be pre-configured. For the traditional CoT, the *Trust Anchor List* (TAL) contains the digital information about the trustworthiness, such as certificates and public keys. Protocol messages whose digital signatures are valid within the TAL are accepted to interact with other entities in the particular CoT. This means that if the certificate is not in the TAL, the entity will not be trustworthy. We could consider that TAL is static because the trust does not evolve any other unknown reasons. Thus, this pre-configured TAL model is obviously not suitable for the dynamic open environment.

In order to reduce repeated login activities, users could login to any service once to gain access to other services within the static CoT by SSO mechanism. However, sometimes, the target service that users demand to access is not in the particular CoT associated with the SSO mechanism. Dynamic trust circle enables any untrusted SP to negotiate with the IdP within the circle. Any untrusted party could join and quit the trust circle automatically after trustworthiness is identified. In an effort to develop a more convenient approach for users to gain access to various services, we present a new trust model called *Dynamic Trust Model* (DTM). Unlike existing CoT approach, DTM allows any untrusted SP to join the particular CoT automatically if that SP is satisfied with policies pre-configured in the administrator of the CoT. This approach to identity management allows users to access any demanded SP by login once if the negotiation between that SP and user's trusted IdP is successful. In this paper, we make the following contributions:

- A dynamic trust policy language is developed, which

extends the *RT* family of policy languages [1] to support trust negotiation. The formal syntax for this language is presented in *Backus Naur Form* (BNF) based on the concept of role membership. A detailed example is also provided to demonstrate the flexibility and usability of this policy language.

- The Dynamic Trust Model is systematically developed to allow untrusted SP to join the existing CoT by negotiation. The protocol is provided to describe the process of trustworthiness establishment.
- The process and algorithm of the negotiation methods are identified, which extends the *Web Service Trust* (WS-Trust) language, to support communication between different entities.

The rest of this paper is organized as follows. We present the overview of the system model in Section 2. Then, we develop a dynamic trust policy language, and provide the syntax language in Section 3. After that, DTM, the protocol, and algorithms are presented in section 4, followed by the discussion of the related works in Section 5. Finally, we conclude our paper and outline our future work in Section 6.

II. SYSTEM MODEL

In this section, we present a user-requested trust negotiation model in order to enable a simple user to access an untrusted SP by SSO.

Figure 1. System Model

As shown in Figure 1, this negotiation is presented between the trusted IdP and an untrusted SP. Generally, in a CoT, there are many trusted principles such as service providers, identity providers, delegators, and users. For the purpose of simplicity, in our model, we assume that there is only one IdP in the CoT.

Unfortunately, the traditional CoT, as we known, is static because all the trust relationships are pre-configured in TAL. In this kind of trust circle, users could login to any services which are already trusted by their identities in SSO use case.

However, they can not access their demanded services by the same identities if such services are not in the CoT. Therefore, the trust negotiation between untrusted SP and the particular trust circle is needed. In our model, the negotiation between the CoT and the untrusted SP is established to enable such untrusted SP to join the circle temporarily.

III. POLICY LANGUAGE

In this section, we present the dynamic trust policy language, which extends the *Attribute-based Trust Negotiation Language* (ATNL) to support dynamic trust management for SSO. ATNL, a formal language for specifying credentials and policies, is based on *RT* family of trust management language [2]. *RT* language introduces the role membership to define the semantics for many access policies. There are two basic levels in *RT* languages: RT_0 and RT_1 . RT_0 simply identifies the name of the role and relationships between principles and roles. There are four types of role definitions: simple member, simple containment, linking containment, and intersection containment. RT_1 extends RT_0 by adding additional parameters to the definitions of the roles.

A. Dynamic Trust Policy Language and Syntax

Dynamic trust policy language consists of four main parts to describe the policy which is pre-set in a particular principal. The syntax of the policy language is represented in BNF (*Backus Naur Form*). We present a policy-setting to explain all the negotiated information. The details and syntaxes of each part are illustrated in the following text.

1. Policy-setting: the policy of a particular principal consists: policy-issuer, policy-stmt, credential, and trust index number.

Syntax 1: policy-setting

```
<list of X>::=<X>
<set of X>::=<X>
```

```
<policy-setting>::=<set of policy-issuer><set of
policy-stmt><set of credential><set of trust index
number>
```

2. Policy-issuer: policy-issuer takes the form:

```
<prin>.<p-issuer-type>
```

The element *p-issuer-type* contains three different attributes: client, service provider, and identity provider.

Syntax 2: policy-issuer

```
<policy-issuer>::=<prin>."<p-issuer-type>
<p-issuer-type>::=("client"|"SP"|"IdP")
```

3. Policy-stmt: policy-stmt takes the form:

```
<policy-head>←<policy-body>
:::necessary/un-necessary
```

Policy-head and policy-body follow the *RT* structure. A policy-stmt can be either necessary or un-necessary.

Syntax 3: policy-stmt

```

<policy-stmt> ::= <policy-head> " ← " <policy-body> "::" ("necessary" | "un-necessary")
<policy-head> ::= <role>
<policy-body> ::= <prin> "." <role-term> | <p-linking> | <p-intersection>

<role> ::= <prin> "." <role-term>
<role-term> ::= <role-name> | <role-name> "(" <list of field> ")"
<field> ::= <field-name> "=" (<var> | <constant>)
<p-linking> ::= <role-term> | <role-term> "." <p-linking>
<p-intersection> ::= (<prin> "." <role-term> ) | (<prin> "." <role-term> "∩" <policy-body>)

```

Figure 2. Syntax for policy-stmt

These two types will affect the trust negotiation processes. The necessary policy-stmt means that if any entities demand to establish the trust relationship with this entity, they must provide the credentials to match all this kind of statement. The un-necessary policy-stmt means that if any entities demand to establish the trust relationship with this entity, depending on the trust index number, they do not have to provide all the credentials to match this kind of statement.

4. Credential: a credential can be either a simple member credential or a simple containment credential. These two kinds of definitions are the basic types presented in *RT* language. A simple member credential takes the form: $K_A.R \leftarrow K_D$, which means principle K_A considers principal K_D to be a member of role $K_A.R$. A simple containment credential takes the form: $K_A.R \leftarrow K_B.R_1$ where principal K_A defines the role $K_A.R$ to contain all members of $K_B.R_1$ which is defined by K_B .

Syntax 4: credential

```

<credential> ::= <role> " ← " (<prin> | <role>)

```

5. Trust index number: trust index is the number N , $0 \leq N \leq 1$, which presents the acceptable proportion of provided credentials. This number describes the proportion of unnecessary policies with which the negotiator should be satisfied. For example, if the trust index number is 0.8, 80 percent of "un-necessary" policies should be satisfied by negotiator's credentials to make sure the negotiation process is successful. This number presents the lowest acceptable value of a particular trust level which is decided by the IdP or SP, which means the bigger N is, the higher trust level is. In addition, this number could make the negotiation more flexible, the negotiator could partly disclosure the credentials to satisfy the policies in stead of matching the policies very strictly.

B. An Example

We present an example to explain how the dynamic trust policy language is deployed in particular principals. The two

negotiators are MyUniversity, an identity provider of the trust domain, and an untrusted service provider GoodLive. A principal Alex signed in one of the services in the trust domain which is authorized by MyUniversity. At this time, Alex is willing to access another service GoodLive that is not in this trust domain. The GoodLive service is an accommodation service which is not in MyUniversity's trust circle.

From GoodLive's perspective, Goodlive offers a permission to anyone who satisfies its policy. To satisfy its policy, the requestor should be certified as an undergraduate informatics student by any level 5 university (authorized by government), and the requestor should be more than 18 years old (The age should be shown on driver license or passport). Furthermore, GoodLive holds many additional credentials. For example, Goodlive is a Realestate service member and a good service provider certified by Realestate.

From MyUniversity's perspective, MyUniversity only conditionally allows some service providers to join its trust circle. The service provider should be a Realestate member and a good service provider certified by Realestate as well. MyUniversity also holds Alex's credentials when Alex registered in MyUniversity.

The policy settings in MyUniversity and GoodLive are shown in Figure 3 and 4.

IV. DYNAMIC TRUST MODEL AND PROTOCOL

A. DTM

As depicted in Figure 5, the Trust Engines are presented on both sides of Identity Provider and Untrusted Service Provider. All information for the negotiation is transmitted between Trust Engines.

Generally, for the traditional CoT, the TAL is pre-configured in every entity before interactions take place. TAL contains the digital information about the trustworthiness, such as certificates and public keys. Protocol messages whose digital signatures are valid within the TAL are accepted to interact with other entities in the particular CoT. This means if the certificate is not in the TAL, the entity will not be trustworthy. We could consider that TAL is static because the trust does not evolve with time, experiences,

Figure 3. MyUniversity's Policy.

Figure 4. GoodLive's policy.

and any other unknown reasons. Thus, this pre-configured TAL model is obviously not suitable for the dynamic open environment.

Instead of the static TAL, we present an advanced *Dynamic Trust List* (DTL) which is based on the negotiation between entities. Any untrusted entities could join the DTL if they are considered to be trustworthy by the identity provider of the CoT. DTL could contain more information about the entities. The DTL will be updated automatically under special events, such as the completion of a successful negotiation. The trust engine, a logic block, processes the trust information between external entities and internal parties. This logic component is also responsible for the management of DTL. The trustworthiness between entities is established based on policies. Thus the trust engine makes the decision whether the external entities satisfy the policy that is preset in internal entity. If the credentials, which are held in the external entity, satisfy that policy, the external entity will be added into the DTL of the internal party; otherwise, it will be refused to join the CoT.

Figure 5. Dynamic Trust Model

B. Protocol

As soon as the trust is established between the internal identity provider and the external service provider, the SP is considered as the trust entity and joins the DTL which is configured in internal IdP. The details of the trust negotiation protocol, as shown in Figure 6, are as follows.

Figure 6. Protocol for trust negotiation

- Step 1: A user logs in to the service by her/his identity and password which are authorized by the internal identity provider.
- Step 2: The internal identity provider passes the user's credentials on to the user. Generally, at this time, the user could single sign on other trusted services in the CoT by using that authorized credentials.
- Step 3: The user uses the credentials authorized by the internal identity provider to make a request to access the demanded service provider which is the external untrusted service provider.
- Step 4: The external untrusted SP sends a trust negotiation request to the internal IdP. The policy content and other trust negotiation information are contained in the request message.
- Step 5: The internal IdP responds to the external SP with a message containing the parameters, credentials, and further negotiation requests including the IdP's policy statement.
- Step 6: The external SP makes the decision whether the IdP could be trusted, and responds to IdP with its own credentials.
- Step 7: The internal IdP makes the decision whether the external SP could be trusted and notifies the user about the decision. If the trust relationship is established between them, the external SP will join the DTL and allow that particular user to use its service. Otherwise, that SP will refuse the user to access.

In order to implement the trust negotiation between the internal IdP and external SP, we present a function SATISFY(k, K) to be sent in the security token messages. SATISFY(k, K) represents whether two entities k and K mutually satisfy the respective policies. The security token message with SATISFY function consists of four main parts: target, policy issuer, policy statement, and value.

- Target: the entity who is the target entity to be satisfied

with the policy of requestor.

- Policy Issuer: the entity who sent SATISFY request message.
- Policy Statement: the content of the policy.
- Value: 00 or 01 or 10. If entity k is satisfied with policy of K , the value is 01, otherwise is 00. If entity K is satisfied with policy of k , the value is 10, otherwise is 00.

C. Process and Algorithm

To implement the protocol above, we choose *Web Service Trust* (WS-Trust) language to send and receive messages between different entities. Two algorithms are presented to describe how to calculate the value of SATISFY(k, K) in the Trust Engine. The processes of interaction between the external SP and the internal IdP are described in Figure 7.

Figure 7. Process of dynamic trust model

The process 1 in Figure 7, the external SP sends a trust negotiation request to the internal IdP. The format of the sending message is presented in WS-Trust language as below:

```
<wst:RequestSATISFY(k,K)>
  SATISFY(k,K).target
  SATISFY(k, K).policy-issuer
  SATISFY(k, K).policy-stmt
  SATISFY(k, K).value
< /wst:RequestSATISFY(k,K)>
```

The general mechanisms of WS-Trust that are defined for requesting and returning security tokens are extensible. A request is initiated with a `<wst:RequestSATISFY(k,K)>` that identifies the details of the request; and it could contain

initial negotiation or challenge information. Every request needs a response message. A response is returned with a `<wst:RequestSATISFY(k,K) Response>` that contains additional negotiation or challenge information. For example, the process 2 in Figure 7, a response message has to be returned to the external SP. Additional negotiation information could be contained in element `<challenge>` as described WS-Trust specification.

The Trust Engine will manage the value of SATISFY (k,K) to establish the trust relationship between k and K. As the target is different, value(k) and value(K) are received from different entities. If the value (k) + value (K) = 11, the external SP will be added into the DTL of the internal IdP.

The algorithms for calculating the value of SATISFY (k, K) in the Trust Engine are described as follows.

Algorithm 1: for calculating the value of SATISFY(k,K)

Input: Target(k), Policy Issuer(policy-issuer), Policy Statement(policy-stmt)

- 1) Start.
- 2) Search credentials C_n from target to match the policy-stmt. If matched credentials are found, output value (k/K) = 01, go to Algorithm 2.
- 3) If matched credentials are not found, output value (k/K) = 00, go to process 4.

Algorithm 2: for decision of negotiation

Input: value (k), value (K)

- 1) If value (k) or value (K) is empty, go to process 3.
- 2) If value (k) + value (K) = 11, go to process 5.

V. RELATED WORK

The concept of CoT is recognized as part of the federated identity vision by the *Liberty Alliance Federation Framework* (ID-FF) [3]. They have developed specifications and guidelines for organizations to establish a legally binding CoT. Similar federated identity management approaches which can be accomplished by means of formal Internet standards are presented, such as *OASIS's Security Assertion Markup Language* (SAML) [4], *Shibboleth* [5], *OpenID* [6], or *WS-Federation* [7]. SAML, which is highly flexible, defines an XML based framework to allow the exchange of security assertions between entities. *Shibboleth* is normally used in educational environments; and the trust between different domains is implemented by agreement of common rules and contracts based on *PKI* infrastructure. *OpenID* is a decentralized framework for user-centric digital identity; thus, the specification could only cover a narrow range of SSO use cases. *WS-Federation*, is an identity federation specification which allows different domains to negotiate

information on identities or attributes. These frameworks only rely on the static CoT, even some of them are flexible.

A related area of work is automated trust negotiation which was introduced by Winsborough et al. [8]. The trust negotiation policy language is based on *RT* [9], which is a family of Role-based Trust-management language. J. Li and N. Li [10] have developed an Attribute-based Trust Negotiation Language (ATNL) for specifying credentials and policies. ATNL allows one to specify policies that govern the disclosure of partial information about a sensitive attribute. These works in the area of automated trust negotiation are considerable contributive. However, they did not concern about the dynamic trust in federated identity management.

Adam J. Lee and T. Yu [11] introduced a dynamic trust model which is a first step to consist vertical trust and horizontal trust. Composite Trust Model (CTM), a flexible policy language, was developed to allow arbitrary composition of horizontal and vertical trust metrics. Cross-domain resource sharing fundamentally depends on digital credentials for access control. Extensive research has been done in the areas of trust management [12], [13] and distributed proofs [14], [15]. F. Almenarez and P. Arias [16] also introduced reputation system into a dynamic trust model. They extended SAML standard in order to facilitate the creation of federation relationships in a secure dynamic way between prior unknown parties. These works paid more attention on the reputations. However, to introduce reputation into federated identity management is a contestable issue.

VI. CONCLUSION AND FUTURE WORK

The current federation frameworks which attempt to achieve identity federation have drawbacks and can not be deployed in dynamic open environments. Underlying trust models are too limited to establish trustworthiness between entities, especially when it comes to interaction with previously unknown parties. The participants of a static Circle of Trust excessively rely on pre-configured Trust Anchor List, which means the trust relationship is established mostly in business level. In this paper, we took the first step towards developing a dynamic trust management model for federated identity management. We developed a dynamic trust policy language, which is based on ATNL and *RT* language. The Dynamic Trust Model was proposed to allow untrusted SP to join the existing CoT automatically by negotiation. We showed the SATISFT function to implement the negotiation via sending and receiving messages based on Web Service Trust language.

In the future, we plan to develop a complex dynamic trust model based upon DTM. In current time, DTM is the first step, we only considered one identity provider in particular CoT. However, the real world situation is much more complicated with more than one identity provider in a CoT, and the untrusted service providers sometimes are authorized by other identity providers or delegators.

Furthermore, we will address issues related to security and privacy in dynamic open environment.

ACKNOWLEDGEMENT

This research is partially supported by a URC Near Miss Grant of the University of Wollongong.

REFERENCES

- [1] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35-86, Feb. 2003.
- [2] N. Li and J. C. Mitchell. Datalog with constraints: A foundation for trust management languages. In *Proceedings of the Fifth International Symposium on Practical Aspects of Declarative languages (PADL 2003)*, pages 58-73. Springer, Jan. 2003.
- [3] Thomas Wason (Ed.) Liberty ID-FF architecture overview v1.2 *LibertyAlliance Specification*, 2004.
- [4] R. Philpott S. Cantor, J. Kemp and E. Maler. Assertions and Protocols for the Security Assertion Markup Language (SAML) V2.0. *OASIS Security Services Technical Committee Standard*, 2005.
- [5] Internet2. Shibboleth architecture. Available in <http://shibboleth.internet2.edu>.
- [6] OpenID. Openid authentication 2.0. Available in <http://www.openid.net>.
- [7] Web services federation language version 1.1, December 2006.
- [8] W. H. Winsborough, K. E. Seamons, and V.E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition*, volume 1, pages 88-102, IEEE Press, Jan. 2000.
- [9] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114-130. IEEE Computer Society Press, May 2002.
- [10] J. Li, N. Li, and W.H. Winsborough. Automated trust negotiation using cryptographic credentials. In *ACM Conference on Computer and Communications Security, Virginia*, pages 46-57, US, Nov. 2003.
- [11] A. J. Lee and T. Yu. Towards a dynamic and composite model of trust. In *ACM Symposium on Access Control Models and Technologies*, pages 217-226, Stresa, Italy, June, 2009.
- [12] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *IEEE Symposium on Security and Privacy*, Oakland, pages 164-173, CA, May 1996.
- [13] N. Li and J. C. Mitchell. RT: A role-based trust-management framework. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition*, pages 201-212, Apr. 2003.
- [14] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 81-95, May 2005.
- [15] T. Jim. SD3: A trust management system with certified evaluation. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 106-115, May 2001.
- [16] F. Almenarez, P. Arias, A. Marin, and D. Diaz. Towards dynamic trust establishment for identity federation. In *Conference of the Euro-American Association on Telematics and Information Systems*, Prague, CZ, June, 2009.