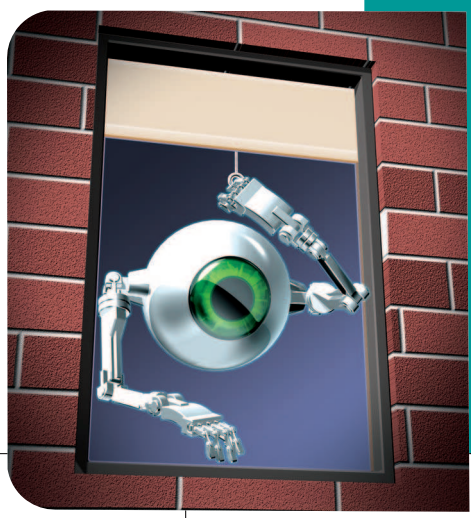
**CERIAS Tech Report 2007-15**

# INTEGRATING FEDERATED DIGITAL IDENTITY MANAGEMENT AND TRUST NEGOTIATION-- ISSUES AND SOLUTIONS

by Abhilasha Bhargav-Spantzel and Anna Squicciarini and Elisa Bertino

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

# Trust Negotiation in Identity Management



Most organizations require the verification of personal information before providing services, and the privacy of such information is of growing concern. The authors show how federated identity management systems can better protect users' information when integrated with trust negotiation.

Abhilasha Bhargav-Spantzel, Anna C. Squicciarini, and Elisa Bertino
*Purdue University*

In today's increasingly competitive business environment, more and more leading organizations are building Web-based infrastructures to gain the strategic advantages of collaborative networking. However, to facilitate collaboration and fully exploit such infrastructures, organizations must identify each user in the collaborative network as well as the resources each user is authorized to access. User identification and access control must be carried out so as to maximize user convenience and privacy without increasing organizations' operational costs. A *federation* can serve as the basic context for determining suitable solutions to this issue. A federation is a set of organizations that establish trust relationships with respect to the identity information—the *federated identity information*—that is considered valid. A federated identity management system (IdM) provides a group of organizations that collaborate with mechanisms for managing and gaining access to user identity information and other resources across organizational boundaries.

IdM systems involve at least two types of entities: *identity providers* and *service providers*. An IdP manages user authentication and user-identity-relevant information. An SP offers services to users who satisfy the policy requirements associated with these services. It specifies and enforces the access control policies for the resources it offers. An organization in a federation can act as both an IdP and an SP. In most IdM systems (see the "Initiatives and systems" sidebar), IdPs authenticate users using single-sign-on technology. With SSO, users can log on with the same username and password for seamless access to federated services within one or multiple organizations. *Federated identity* includes not only users' login names, but also user properties, or *user identity attributes* (user attributes, for short). Thus, authorizations specified for a given resource are no longer expressed in terms of user login IDs but in terms of requirements and conditions against user properties.

One challenge with current IdM systems is distributing the IdPs' functionality among IdPs and SPs (in this article, we don't differentiate between service providers and IdPs in a federation). We need a secure and privacy-preserving mechanism for retrieving the user attributes from different SPs. The IdM system must provide only the user information that is needed to satisfy the requesting SPs' access control policies. In this regard, users have differentiated privacy preferences for various types of personal information.[1] For example, users might agree to share demographic information but not credit card or health information. Such requirements call for a flexible and selective approach to sharing user attributes in federations. A system could achieve selective release of identity by supporting multiple federated digital identities. For example, a user could have a business identity and a personal identity, and their corresponding profiles would have associated privacy preferences. Such an approach, however, contradicts the main aim of federated identity solutions—that is, minimizing the management of multiple profiles by the user.

One way to achieve such flexibility and fine-grained access is to enhance IdM technology with automated trust-negotiation (ATN) techniques.[2] Trust negotiation is an emerging access control approach that aims to establish trust between negotiating parties online through bilateral credential disclosure. Such a negotiation aims to establish a trust level sufficient to release sensitive resources, which can be either data or services.

## Initiatives and systems

Liberty Alliance and WS-Federation are two emerging standards for identity federation in the corporate world. Because these projects are similar, we only describe the former.

Liberty Alliance (www.projectliberty.org) is based on Security Assertion Markup Language (SAML) and provides open standards for single sign-on with decentralized authentication. SSO lets users sign on once at a Liberty-enabled site and remain signed on when navigating to other Liberty-enabled sites. This group of Liberty-enabled sites belongs to a circle of trust—that is, a federation of SPs and IdPs based on the Liberty architecture. The IdP is a Liberty-enabled entity that creates, maintains, and manages user identity information and provides SPs with this information. Similarly, the federated attribute management and trust-negotiation (FAMTN) framework builds on an SSO and provides a flexible decentralized trust management system for registered users.

According to the Liberty Alliance framework, a federation might include multiple IdPs, which could also be SPs. Basically, in a given Liberty circle of trust, multiple IdPs can share a user's information. These IdPs establish trust relationships and access policies a priori while forming the circle of trust. The Liberty protocols don't dictate the underlying semantics and related protocols. Truly decentralized identity management requires a more automatic methodology for federating user information among IdPs. The FAMTN framework doesn't distinguish SPs from IdPs. Each SP in the federation can act as an IdP. SPs exchange information through automatic trust negotiation (ATN), according to an on-demand dynamic protocol.

The Shibboleth (http://shibboleth.internet2.edu) initiative originated in academia and is similar to the Liberty Alliance in that it aims to facilitate resource sharing between research and academic institutions. It extends the federated identity information concept to federated user attributes. When a user at an institution tries to use a resource at another, Shibboleth sends attributes about the user to the remote institution, rather than making the user log in to that institution. The receiver can check whether the attributes satisfy the SP's policy. The Shibboleth IdP accounts for all user attributes and user privacy preferences when giving information to other SPs. The FAMTN approach differs from Shibboleth in that it doesn't rely on a central IdP for all user attributes. Rather, user attributes are distributed among the federation SPs, each of which can act as an IdP. The ability to negotiate with different SPs adds flexibility to how users can define different privacy preferences with respect to federation members. Shibboleth requires trust agreements to define the population, retention, and use of attributes, thus making it difficult for external users (who aren't affiliated with the federation) to carry on ad hoc negotiations for the various services offered. In other words, unlike our framework, Shibboleth isn't open to external users.

Researchers have developed several systems and prototypes for trust negotiations in Web-based applications. *TrustBuilder*,[1] one of the most significant proposals, provides a set of negotiation protocols that define the message ordering and the type of information the messages will contain, as well as strategies for controlling the messages' exact content. It defines various strategies to let strangers establish trust by exchanging digital

In this article, we discuss how to integrate federated IdM with trust-negotiation techniques. More specifically, we discuss how to implement trust negotiation between SPs in a federation, and between users and SPs. This is, to the best of our knowledge, the first attempt to integrate a federated IdM system with a trust-negotiation system. A key aspect of the resulting framework—*federated attribute management and trust negotiation* (FAMTN)—is that a user doesn't have to provide a federated attribute (that is, attributes the user is willing to share in a federation) more than once to a given federation. Internal users of FAMTN systems can perform negotiations by exploiting their SSO ID without having to repeat identity verification. Further, a FAMTN system supports temporary SSO, so external users can perform negotiations with the federation using the federated framework to reduce the amount of identity information they need to provide.

## Comparison of IdM and ATN systems

The trust-negotiation paradigm has several similarities to federated IdM. Both aim to better handle users' sensitive information; however, trust negotiation ultimately aims to handle introductions between strangers, whereas IdM systems are typically for closed environments.

ATN systems and IdM systems also differ in several important ways, as Table 1 shows. Importantly, we based our analysis on the IdM and ATN models as they were originally designed. Researchers have proposed variations to both approaches in the past few years, which make the evaluation results slightly different.

### Open versus closed environment

ATN techniques,[3] developed for use in open systems, provide protocols for introducing strangers to each other. They might be useful for the initial trust-establishment process between users and IdPs or to automatically manage introductions between different federation groups.

### Credential and identity attribute management

In a typical ATN system, the user is the IdP. ATN is a user-centric system in which a client stores credentials and provides them on behalf of a user through negotia-

credentials and using access control policies that specify the combinations of credentials a stranger must disclose to gain access to each local service or credential. Marianne Winslett and her colleagues[2] developed *Unipro*, a unified scheme to model resource protection, including policies. It represents one of the most significant proposals in the negotiation research area, and most significantly influenced our work. However, Unipro doesn't support privacy policies, nor does it define an ad hoc policy language.

Kent Seamons and his colleagues[3] explored the issue of supporting sensitive policies, obtained by introducing hierarchies in policy definitions. They also addressed privacy issues in trust negotiation.[4] However, their approach doesn't provide a comprehensive solution to such problems because it only deals with protecting sensitive policies, achieved by dynamically modifying policies during a negotiation.

William Winsborough and Ninghui Li[5] introduced a role-based trust-management language that they use to map entities to roles based on the properties described in their credentials. They also developed an algorithm to locate and retrieve credentials that aren't locally available. This *credential chain discovery* is an important aspect of trust negotiation because assuming the credentials to be locally stored is too strong an assumption for decentralized collaborative environments.

We based our framework on Trust-$\chi$,[6] a trust-negotiation system for peer-to-peer environments. Trust-c is complemented by an ad hoc XML based language, $\chi$-TNL, for encoding negotiation policies, digital credentials, and security-related information. A main difference between Trust-$\chi$ and our work is that FAMTN's negotiation process is much more articulated and can involve third parties in addition to the two parties initiating the negotiation. Thus, FAMTN is characterized by multiparty negotiations, as opposed to Trust-$\chi$'s two-party negotiations.

Having been widely studied in theory, ATN systems are now ready for use in real applications. TrustBuilder is an example of an actual system for support of trust negotiations. Current Web services only provide basic negotiation capabilities. The full potential of trust negotiations will be achieved when the practical limitations related with public-key infrastructures are overcome.

**References**


1. T. Yu, M. Winslett, and K.E. Seamons, "Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies for Automated Trust Negotiation," *ACM Trans. Information and System Security*, vol. 6, no. 1, 2003, ACM Press, pp. 1–42.
2. T. Yu and M. Winslett, "A Unified Scheme for Resource Protection in Automated Trust Negotiation," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, 2003, pp. 110–123.
3. K.E. Seamons, M. Winslett, and T. Yu, "Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation," *Proc. Network and Distributed System Security Symp.*, Internet Soc., 2001.
4. K.E. Seamons, M. Winslett, and T. Yu, "Protecting Privacy During On Line Trust Negotiation, *Proc. 2nd Workshop Privacy Enhancing Technologies*, Springer Berlin/Heidelberg, 2002, pp. 129–143.
5. W.H. Winsborough and N. Li, "Protecting Sensitive Attributes in Automated Trust Negotiation," *Proc. ACM Workshop Privacy in the Electronic Soc.*, ACM Press, 2002, pp. 41–51.
6. E. Bertino, E. Ferrari, and A.C. Squicciarini, "Trust-c: A Peer-to-Peer Framework for Trust Establishment," *IEEE Trans. Knowledge and Data Eng.*, vol. 16, no. 7, 2004, pp. 827– 842.


**Table 1. Automated trust negotiation (ATN) versus identity management (IdM) systems.**

| CRITERIA | ATN SYSTEMS | IDM SYSTEMS |
|---|---|---|
| Environment | Open | Closed |
| Credential management | User centric | Polycentric |
| Attributes used | Certified attributes or credentials | Certified and uncertified attributes |
| Attribute encoding | X.509 certificates, XML certificates | Username, Security Assertion Markup Language (SAML) assertions, X.509 certificates, Kerberos tickets |
| Architecture | Peer-to-peer | Client-server |
| Policies | Privacy policies, access control policies | Privacy policies, authorization policies |
| Policy language | XML-based trust-negotiation language (X-TNL), register transfer, Protune, and so on | Extensible Access Control Markup Language (XACML) |
| Trust model | Pairwise trust (some brokered trust) | Pairwise trust, brokered trust, community trust |
| Unique identification | Optional | Single sign-on required |
| Credential discovery | Credential chain management protocols | Discovery service protocols |

tion. Although recent work has looked at storing user credentials with SPs using anonymous credentials, most ATN systems assume that users directly manage their own credentials. In IdM systems, on the other hand, SPs save user profiles for future use in the federation according to the user's privacy preferences.

ATNs typically negotiate certified attributes or credentials. IdM systems mainly use uncertified attributes, al-

though they can also support certified attributes. IdM systems usually rely on Security Assertion Markup Language (SAML) assertions for encoding attributes, whereas in ATN systems, attributes are encoded in credentials, which are the digital equivalent of physical certificates, represented according to the X.509 certificate format.

### Architecture

An ATN system is typically used in peer-to-peer (P2P) systems, so clients and servers have the same basic architecture. Any entity serving as provider in a trust negotiation can act as a client in a different negotiation. In IdM frameworks, IdPs, SPs, and clients all have different architectural components depending on that entity's functionality. The P2P nature of ATN systems simplifies the integration of an ATN's architectural components with the existing IdM systems.

### Policies

Both IdM and ATN systems aim to satisfy user privacy preferences for their personal data and to ensure that access control policies are stated and enforced. So, both offer privacy and access control policies. However, in ATN systems, access control policies play a key role in the trust-negotiation processes, whereas they're only a marginal aspect in IdM systems. As such, ATN policies can be more complex and provide alternative ways of satisfying the requirements for access to a given resource or expressing different usage conditions. This ensures soundness for any transaction, meaning that if user preferences and the SP's requirements are compatible, the transaction will certainly succeed. Soundness isn't guaranteed in current IdM systems because they lack formal negotiation procedures and a corresponding expressive policy language. However, IdM systems provide mechanisms for policy exchange that additional negotiation modules could use to provide ATN functions.

### User identity

Both ATN and IdM systems require users to be identified. Such a requirement is particularly relevant in IdM systems, which aim to uniquely identify users within federations. Users in an IdM mostly need an SSO to interact with any SP in the federation and to ensure that their attributes are linked to them. By contrast, identity is usually a secondary aspect in ATN systems because authentication is based mainly on user properties rather than on the sole identity. However, real case scenarios show that authentication is often a first-class requirement in specific negotiations. Further, IdM systems rely on SSO to identify users, so there's no need to certify user identities in other ways. ATN systems obtain identities using credential combinations, although they might use SSO in specific contexts. In ATN systems, there's no need to link multiple negotiations to the same identity because identification is (if required) executed on the fly, while the negotiation process is taking place.

### Trust model

A typical IdM system has three types of trust models:[4]

- a *pairwise* model for two entities that have direct business agreements with each other;
- a *brokered* trust model for two entities that don't have a direct agreement with each other, but have agreements with one or more intermediaries so as to enable construction of a business trust path between the two entities; and
- a *community* trust model for several entities that have a common business agreements within the community or federation.

Although all three trust models can use ATN systems, the brokered trust model integrated with ATN provides a unique feature to existing IdM systems.

### Other similarities

Both ATN and IdM also require credential discovery, although they use different methods. Using a discovery service, IdMs collaborate to make assertions about a user from a local IdP to a remote IdP. Similarly, ATN systems use credential discovery to retrieve remote credentials not available at the negotiating parties.

Another related aspect is delegation. Although delegation isn't a main issue in trust negotiations, both IdM and ATN systems achieve delegation through ad hoc protocols and credentials enabling entities to negotiate on behalf of third parties. In IdM systems, we can use the brokered trust model to delegate the responsibility for attribute assertion to another IdP that the user trusts more.

## Integrating IdM and trust negotiations

FAMTN combines the advantages of the IdM and ATN approaches, providing a truly distributed approach to managing user identities and attributes with negotiation capabilities.

A FAMTN federation essentially involves two type of entities: FAMTN service providers (FSP) and users. In the FAMTN framework, we don't distinguish between SPs and IdPs: each SP in the federation can act as an IdP. SPs exchange information through ATN, according to an on-demand dynamic protocol. FSPs support identity and attribute provisioning, as we detail later.

Our approach supports negotiations between an FSP and the user, and between two FSPs in the same federation. The protocol for negotiations between FSPs and users depends on the interacting user's type. The distinction is based on the user's membership in the federation. A user who's affiliated with an organization within the

federation is a *member user* of the federation. The federation is more likely to have information about a member user even if the member hasn't accessed any of its services. This also depends on the member organization's policy, which defines which of its affiliated user attributes are federated. An SSO user identification identifies the member in the federation.

On the contrary, *external users* must provide all required attributes at their first negotiation. The first negotiation between an external user and an FSP includes identity provisioning, because the provider issues a temporary user ID to be used within the federation. The use of time-limited SSO ID for nonmembers ensures identity linkability. (We can reasonably assume that the federation policy defines the time interval.) Of course, users might have multiple identities but choose to adopt one for requesting access to service. We don't elaborate on this issue because it goes beyond our article's scope. By interacting further with the federation, the amount of information about users that is disclosed to the federation increases. This information can be linked to the user (who becomes as *repeated external user*) and thus reused in the subsequent negotiations. As a result, the system executes more efficient negotiations with fewer attributes required from the user.

Figure 1 shows an example. User $U$ requests service from service provider $SP_1$. $SP_1$ requires user attributes ($a$, $b$) to satisfy its service policy. $U$ provides ($a$, $b$) and gets the service. Suppose that $U$, at the end of this successful negotiation, opts for sharing attribute ($a$) within the federation, and suppose that $U$ then requires a service from another provider $SP_2$ in the same federation. Suppose that the attribute requirements there are ($a$, $c$). In this case, however, $U$ only has to provide the attribute c to receive the service.

At the end of a successful negotiation, users receive one of two types of ticket:

- a *trust ticket* provides information about the previous services and FSPs the user has accessed; and
- a *session ticket* provides recent history information to help speed up negotiations, as we detail later.

The second type of negotiation occurs between two FSPs. This negotiation type is useful when a user successfully negotiates a service from one FSP and automatically becomes eligible to receive service from another FSP. As such, when the user asks for a service, the FSP providing it can directly negotiate user-related attributes with the FSP holding such attributes from previous negotiations. Also, negotiations among FSPs might be required for verifying external user identities. Because we don't rely on a single IdP, an IdP might not be aware of the last registered users. When the FSP receives a request from a locally unknown user ID, it can directly interact with the SP that issued the claimed user ID to double check its validity (for simplicity, we assume the user ID contains FSP information to easily identify the issuer).

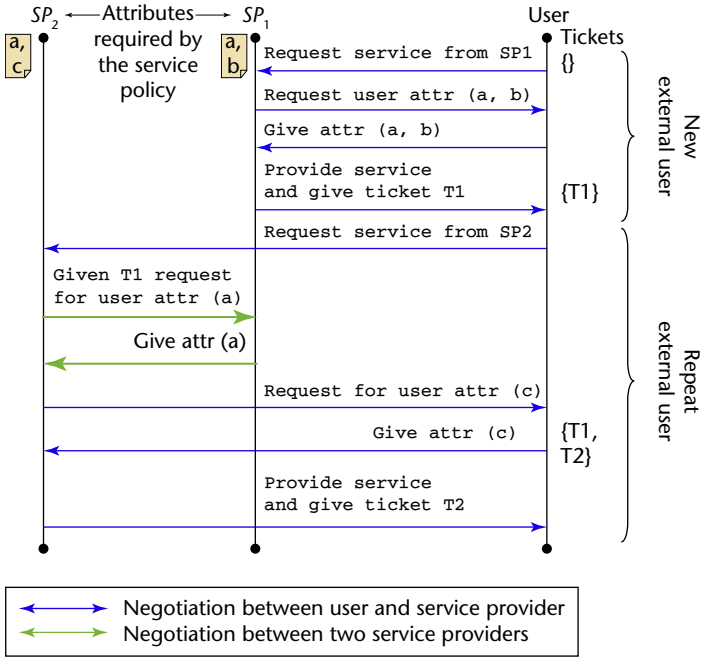



Figure 1. External user negotiating with two service providers (SPs) of a federation. A user who has already provided attributes to any SP in the federation might not need to provide them again when another SP in the federation requires them.

## Architecture of service providers in FAMTN

A FAMTN framework consists of an FSP containing the necessary components required to execute two functions: trust negotiation among users and FSPs and federation of user attributes.

Figure 2 shows the FSP architecture. An FSP's components derive from FAMTN's two underlying frameworks: ATN and federated IdM. Each FSP can perform the functionality of an IdP and an SP.

The FSP's main components are:

- the *Web services* component, which enables secure communication within the federation and with the users; and
- the *user negotiation* component, which contains the modules executing the negotiation, depending on whether the user is a member or nonmember (this component is directly related to the trust ticket management layer).

Other parts of the FSP include the trust ticket management layer which manages the trust tickets and the session tickets required for the negotiation. The *policy management and enforcement* components store the authentication and

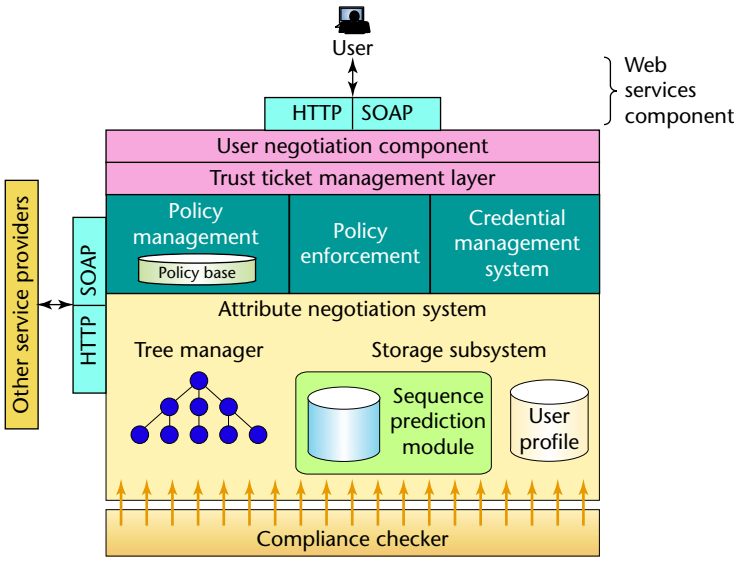



Figure 2. The federated attribute management and trust negotiation (FAMTN) service provider architecture.

access control policies in the policy base and enforce them, respectively. The *credential management* system manages and validates certificates and user tickets by verifying the FSPs' signatures. It's also responsible for revocation when required. The *attribute negotiation* system consists of the main components required for negotiation:

- the *tree manager*, which stores the negotiation's state;
- the *storage subsystem* containing the *sequence prediction module*, which caches and manages previously used trust sequences and user profile information; and
- the *compliance checker*, which tests policy satisfaction and determines request replies during a negotiation.

### An example use case

Figure 3 shows an example scenario of the Liberty Web services framework (WSF)[5] with additional FSP components. (See the "Initiatives and systems" sidebar for more on Liberty Alliance, which provides open standards for SSO with decentralized authentication.) In this scenario, the following steps take place:

1. A user, say Joe, accesses $SP_1$ using SSO.
2. Using redirection and IdM system protocols, an IdP transmits a SAML assertion authenticating Joe to $SP_1$.
3. $SP_1$ requires a certificate from Joe to verify his address for delivery and that he is older than 21.
4. Joe doesn't trust $SP_1$ so won't reveal his certified credential to it. He therefore negotiates with the IdP and reveals his credential to it instead.
5. $SP_1$ negotiates with the IdP, which finally sends a SAML assertion stating whether Joe satisfies $SP_1$'s age criteria. So, Joe doesn't have to reveal the actual credential to $SP_1$, ensuring that the credential is stored only with a trusted party.
6. Joe also registers his address with $SP_1$ for delivery but imposes as a condition that his address should be released only to a member of the federation and only when the address is required for a purchased product delivery and the member is certified by the Better Business Bureau (BBB).
7. Joe subsequently accesses $SP_2$ to order a pizza. Because of SSO he gets seamless access.
8. $SP_2$ asks Joe for his address. Joe tells $SP_2$ to get his profile from other sites in the federation. (In this case, it's actually an agent operating at the client on behalf of Joe that suggests request redirections. We use Joe to simplify the example's presentation.) Using the discovery service, $SP_2$ contacts $SP_1$, who negotiates with $SP_2$ to verify that the conditions for Joe's attribute release are met. If the negotiation succeeds, $SP_2$ receives the required information and can make the appropriate delivery.

This example demonstrates how we can implement additional privacy and flexible policies with ATN. Also, not all FSP components are required in a typical IdM system. FSP can leverage modules belonging to the Liberty Alliance Framework or other IdM systems, such as the *discovery service* (DS) and *personal profile* (PP) policy and credential management systems. The ATN-specific parts (the solid color components) in Figure 3 are the subset of FSP components used for ATN in the Liberty WSF framework.

## Negotiations in a FAMTN federation

Session tickets and trust tickets are the main building blocks in our trust negotiation protocols. Both ticket types are temporal with a fixed lifetime. We assume loosely synchronized clocks in the federation. We use the SSO ID as the user ID in the tickets.

A session ticket ensures that if the negotiation ends successfully and the same user requests the same FSP for the same service in a subsequent session, the system can grant the service immediately without having to unnecessarily repeat the trust-establishment process. A session ticket therefore contains the fields $Signed_{FSP}<\tau(s_{req}), u, T, R>$, where $\tau(s_{req})$ denotes the service requested, $u$ is the user ID, and $T$ is the ticket timestamp. Here, $R$ denotes the negotiation's result and can be a simple statement or a structured object. The use of structured objects is particularly interesting for tracing intermediate results of negotiations of aggregated services.

The FSP signs a session ticket and gives a receipt of the trust establishment. Because session tickets are encrypted

with the FSPs private key, they are tamperproof and verifiable. The time-out mechanism depends on the type of user attributes required for the service, and the service's security level.

The trust ticket determines the list of services external users have accessed. Assuming that all the FSPs are members of the same federation, any member provider can verify another member provider's signature. Such a ticket has the following form:

$$Signed_{SP_{last}} \left\langle list\left\{\tau(s), FSP, T\right\}u, T-I\right\rangle$$

Every 3-tuple in the list contains the service type, the corresponding FSP, and the timeout. The variable $u$ corresponds to the temporary user identification, and $T-I$ is the ID's expiration date. The ticket is signed by the last FSP with which the user had a successful transaction. At the end of a successful transaction, the FSP takes the current user trust ticket, removes all timed-out entries, appends its information, signs it, and sends it to the user.

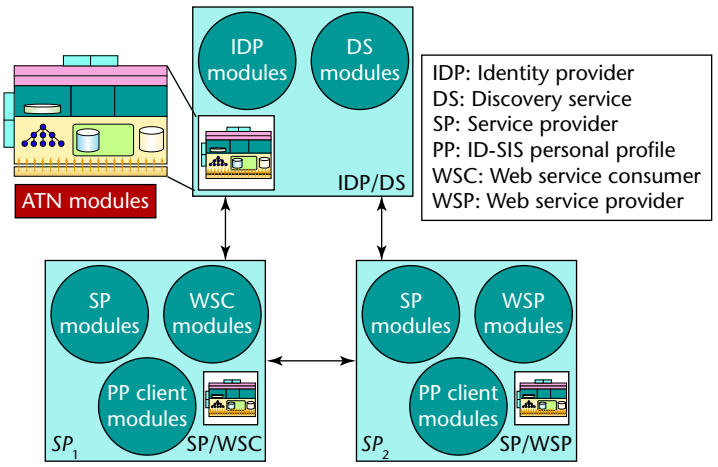



Figure 3. Liberty Web services framework and federated service provider with three Web sites and system modules. The arrows indicate the possible communication of the various module sets.

### Implementing trust tickets through cookies

Many IdM systems use cookies to make user information available to servers. State information is stored at the client, which sends the cookie to the server the next time the user accesses that server. Like session and trust tickets, cookies can be valid only for the session during which they were issued or can persist beyond the session's end. A persistent cookie is typically written to a file on the browser's hard drive if its lifetime hasn't elapsed when the browser is shut down and therefore can be used for a longer period of time. In a truly distributed federation that has more than one IdP, an FSP needs a mechanism to determine which IdP has the user information. In Liberty, this problem is known as the *introduction problem*. Currently, Liberty Alliance protocols rely on cookies for redirecting IdPs.

Cookies offer several advantages. Implementing them is efficient, because you don't need new software to use them, and you can use them independently of any authentication mechanism. They also provide dynamic state information, which is helpful for preventing several security threats. One such threat is an *impersonation* attack, which arises when a user has successfully logged onto one FSP, but the other FSPs in the federation don't re-authenticate the user. Thus if the authentication is no longer valid, because of attacks or other failure, the FSP has no straightforward way to detect it. Cookies help the FSP check whether the authentication ticket is associated with the user identity as well as whether the IdP session is valid for that user. Alternatives to using cookies for the introduction problem are based on interactions with the user either actively or on the use of statically hand-configured list of possible user IdPs. Such approaches inhibit the seamless SSO process and are less efficient.

Cookies, however, have some security problems:[6]

- They're usually in clear text. Headers are generally unprotected and even encrypted cookies are vulnerable to replay attacks.
- Because cookies are stored on local machines, anyone using the machine can easily read them.
- You need to control where cookies are sent, because you wouldn't want to send the user cookie to an untrusted service provider. For example, several current spyware applications exploit user cookies, so we need to better control cookies' destinations.

As a consequence, cookies shouldn't store personal identifiers or sensitive information. In real applications, however, a cookie typically stores the SSO user ID or other tracking record, which might leak information about the user. Better storage and usage protocols and mechanisms can address most of these security vulnerabilities. We propose implementing trust tickets in IdM systems using cookies to exploit cookies' advantages while preventing the vulnerabilities we've just described. Indeed, the timeouts and signed information given by the session and trust tickets contain reliable and dynamic state information. To further increase cookie security, federations should use mechanisms enabling selective download of cookies. Browsers typically give users limited choice about how to handle cookies. Control is coarse-grained: the browser will download no cookies or must accept all cookies. Letting a user choose cookies from a Web site that uses a single domain versus multiple domains can cause problems in federations, which are typically multiple-domain environments. Building server filters is currently complicated and not feasible for average

```
Require: userID, userAuthenticationInfo
Ensure: IsRegistered(userID)
1: userRequest ← getRequest(userID)
2: if userRequest ∉ ServicesFSP then
3:   return Abort-Negotiation
4: end if
5: *Comment: For Members*
6: if isValidMember(userID) = true then
7:   sessionTicket ← getSessionTicket(userID)
8:   if sessionTicket ≠ NULL ^
     sessionTicket.time < timeout then
9:     return OK
10:  end if
11:  M_FSP = getMemberFSP(userID)
12:  remAttrList1 ← NEGOTIATE_FSP (Curr_FSP, M_FSP
13:            userID, userRequest)
14:  if remAttrList1 ≠ NULL then
15:    remAttrList2 ← NEGOTIATE_User(Curr_FSP,
16:              userID,CurrPolicy_FSP)
17:  else
18:    send(SessionTicket) → userID
19:    return OK
20:  end if
21:  if remAttrList2 ≠ NULL then
22:    return Abort-Negotiation
23:  else
24:    send(SessionTicket) → userID
25:    return OK
26:  end if
27: end if
28: *Comment: For Non-Members*
29: FSPlist ← getTrustTicket(userID)
30: while FSPlist ≠ EmptyList do
31:   Mi = rmHeadOfList(FSPlist)
32:   remAttrList3 ← NEGOTIATE_FSP (CurrFSP, M_i
33:            userID, userRequest)
34:   if remAttrList3 = NULL then
35:     send(TrustTicket) → userID
36:     return OK
37:   end if
38: end while
39: if remAttrList3 ≠ NULL then
40:   remAttrList4 ← NEGOTIATE_User(Curr_FSP,
41:            userID,CurrPolicy_FSP)
42: end if
43: if remAttrList4 ≠ NULL then
44:   return Abort-Negotiation
45: else
46:   send(TrustTicket) → userID
47:   return OK
48: end if
```

Figure 4. Algorithm for negotiating trust in FAMTN.

users. Like privacy preferences, a user should be able to set cookie preferences, specifying more fine-grained conditions. For example,

- Accept only signed cookies from a given federation FSP.
- Accept cookies from BBB-certified members by negotiating servers' attributes.
- Send cookies that don't contain personally identifying information.
- Send cookies to FSPs that aren't in a conflict-of-interest class for the FSP that set the cookie.

We need a policy language to express these preferences that can be integrated with cookies' storage and usage mechanisms.

### Negotiation in identity federated systems

The trust-establishment negotiation process depends on the type of user and the history of the user's interactions with the federation members. Algorithm 1 (Figure 4) shows the complete negotiation process developed for FAMTN. It includes all user cases, assuming one federation is in place. Multiple federations with nonempty intersection are outside this article's scope.

Four types of users cases give the basis of the design and analysis of the user–FSP negotiation process. Intuitively, a recent user should obtain service access faster than a new user. The short-termed session tickets help achieve this. Similarly, a repeat user, who has already received services from different FSPs in the federation, should get service access faster than a new external user. This is because the new external user directly negotiates all the required attributes with the FSP, whereas for a repeat user, the FSP can retrieve some of the attributes from FSPs the user has visited before. Information about the previously visited FSPs is in the list of trust tickets, which are retrieved iteratively until user attribute requirements are satisfied. At each iteration, the FSP requiring the user attributes to satisfy its service disclosure policy negotiates with the FSP indicated in the trust ticket. If the retrieved attributes don't suffice, the FSP negotiates directly with the user. Finally, a member user, being internal to the federation and thus more trusted, should have advantages in the negotiation process over a new external (nonmember) user. Indeed, the FSP retrieves the member user at-

## Existing federations

Federated identity can deliver several compelling benefits to organizations. Federation makes it possible for local identities and their associated data to stay in place, yet be linked together through higher-level mechanisms. The following are examples of existing federations.

The SWITCHaai Federation (www.switch.ch/aai/documents.html) is a group of organizations (universities, hospitals, and libraries, for example) that have agreed to cooperate on interorganizational authentication and authorization. They operate a Shibboleth-based authentication and authorization infrastructure (see http://shibboleth.internet2.edu).

By using Shibboleth authentication and authorization technology, InCommon (www.incommonfederation.org) facilitates sharing of protected resources, enabling collaboration between InCommon participants that protects privacy. Access decisions to protected resources are based on user attributes contributed by the user's home institution. InCommon became operational on 5 April 2005.

The HAKA Federation in Finland (www.csc.fi/suomi/funet/middleware) entered its production phase in late 2004. The Federation, established in 2003 and based on Shibboleth, currently includes two (of 20) universities and one (of 29) polytechnics as identity providers, and four service providers, including the National Library Portal (Nelli). In Finland, libraries in higher education traditionally cooperate in licensing electronic journals.

The Liberty Alliance Identity Federation Framework (ID-FF) allows single sign-on and account linking between partners with established trust relationships. The Identity Web Services Framework (ID-WSF) lets groups of trusted partners link to other groups and gives users control over how their information is shared. Finally, the Identity Services Interface Specifications (ID-SIS) will build a set of interoperable services on top of the ID-WSF.

tributes directly from the organizations in the federation within which users are affiliated. This provides an efficient mechanism for retrieving users attributes because it avoids iterated negotiations among all the FSPs a user has interacted with. Here we assume that the affiliated organization stores and possibly certifies all of the member users' attributes. Member users can also use the session tickets like the external users.

Before we can fully integrate federated IdM systems and trust-negotiation, several issues must be addressed, including questions regarding policies—that is, policy compliance and subsumption of policies. The language to define the policies should use vocabulary well understood not only by users and organization, but by the whole set of organizations. This might not be a realistic assumption, and we need to look into alternatives. Policy languages supporting the specification of credential sharing within a federation don't exist and will be useful for better privacy control in a federation. Another important problem is the representation of attributes. This is essential for efficient lookup if several users are using the system. The attribute's meaning and its underlying logic can also help users infer implications between conditional attributes. □

### References


1. D.L. Baumer, J.B. Earp, and P.S. Evers, "Tit for Tat in Cyberspace: Consumer and Website Responses to Anarchy in the Market for Personal Information," *North Carolina J. Law and Technology*, vol. 4, no. 2, 2003, pp. 217–274.
2. H. Skogsrud et al., "Trust-serv: A Lightweight Trust Negotiation Service," *Proc. 30th Int'l Conf. Very Large Data Bases*, Morgan Kaufmann, 2004, pp. 1329–1332.
3. A. Hess et al., "Content-Triggered Trust Negotiation," *ACM Trans. Information Systems Security*, vol. 7, no. 3, 2004, pp. 428–456.
4. Liberty Alliance Project, *Liberty Trust Model Guidelines*, www.projectliberty.org/liberty/content/download/1232/8000/file/liberty-trust-models-guidelines-v1.0.pdf.
5. Liberty Alliance Project, *Liberty Alliance ID-WSF 2.0 Specifications*, www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications.
6. V. Samar, "Single Sign-on Using Cookies for Web Applications," *Proc. 8th Workshop Enabling Technologies on Infrastructure for Collaborative Enterprises* (WETICE), IEEE CS Press, 1999, pp. 158–163.



***Elisa Bertino*** *is professor of computer science and electrical and computer engineering and research director at the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University. Her main research interests include security, database systems, object technology, multimedia systems, Web-based information systems. Bertino has a PhD in computer science from University of Pisa. She is a co-editor in chief of the VLDB Journal and member of the editorial board of several journals, including IEEE Internet Computing, IEEE Security & Privacy, and ACM Transactions on Information and Systems Security. She is a Fellow of the IEEE and the ACM. Contact her at bertino@cs.purdue.edu.*

***Anna Cinzia Squicciarini*** *is a postdoctoral research associate at Purdue University. Her main research interests range from trust negotiations, privacy models and mechanisms for privilege and contract management in virtual organizations to grid systems and federated identity management. Squicciarini has a PhD in computer science from the University of Milan. Contact her at squiccia@cs.purdue.edu.*

***Abhilasha Bhargav-Spantzel*** *is a computer science PhD candidate at CERIAS, Purdue University. Her main research interests include identity management, identity theft protection, cryptography, biometrics, and policy languages. Bhargav-Spantzel has a bachelor's degree in computer science and mathematics from Purdue University. Contact her at bhargav@cs.purdue.edu.*