User-Controlled Automated Identity Delegation

Thorsten Hoellrigl, Holger Kühner, Jochen Dinger, Hannes Hartenstein
Steinbuch Centre for Computing (SCC) and Institute of Telematics
Karlsruhe Institute of Technology (KIT), Germany
Email: {Hoellrigl|Holger.Kuehner|Dinger|Hartenstein}@kit.edu

***Abstract*—The growing number of IT services in distributed systems increases the need to allow users to keep track of which personal data is retained by which service. User-centric federated identity management (FIM) tackles this goal by enabling users to approve each data dissemination between the providers of identity-related information, so-called identity providers (IdPs), and the consumers of this information, the service providers. To prevent a single IdP from gaining a comprehensive set of user information, user-centric FIM motivates the use of multiple IdPs even though this distribution of responsibilities might result in information redundancy and therefore raises consistency issues.**

User-centric FIM systems do not cope with information consistency sufficiently, mainly because these systems require that each dissemination of user attributes is *manually* approved by the user. We propose an approach, named *User-Controlled Automated Identity Delegation*, that allows a controlled data dissemination based on an *automated* user approval by introducing an additional party called *Identity Delegate*. The Identity Delegate is designed in consideration of the following central ideas: (i) *user centricity* – all data dissemination is still under user control, (ii) *privacy* – the delegate cannot read or gather personal data, (iii) *efficiency* – the effort to integrate and operate the delegate within an existing FIM system is kept low. We cover the experience made with an implementation based on Windows CardSpace.

I. Introduction

In distributed IT systems, users face the challenge to keep track of which information they provided to which service. This challenge strongly affects usability and privacy. The user-centric federated identity management (FIM) strives to improve *usability* and *privacy* by "empowering human beings to control their identities" [1]. A major characteristic of this "user empowerment" is that a user is enabled to freely choose at which identity provider (IdP) her identity-related information – namely attributes – is managed. Due to the following reasons, each user typically uses multiple IdPs: (i) service providers (SPs) and users do not trust in one central IdP, (ii) user attributes do typically have different trusted sources of authority, (iii) many SPs do not trust in information provided by self-hosted IdPs.

A consequence of multiple IdPs is that user attributes might be stored redundantly at the IdPs. In addition, user-centric FIM systems cannot fully avoid that SPs also store attributes locally to be able to deliver a service. This information replication results in multiple identities per user and typically heterogeneous information schemas and requires that changes to information are disseminated to all replicas to avoid inconsistencies. For instance, changes to the status of a user identity, like a deactivation are of interest for all IdPs and SPs that do also store attributes of that user.

To analyze if user-centric FIM systems do cope with the consistency issue, the way attributes are exchanged has to be considered. Current user-centric FIM systems only offer to exchange attributes between an IdP and SPs. An attribute exchange between different IdPs is not considered. All communication between an IdP and SPs has to pass the user client as this component enables the user to approve the exchange. A direct attribute flow from IdP to SP is avoided. Thus, inconsistencies between an IdP and SP can only be resolved during service usage as an SP only gets attributes when the user is actually using the service and *manually* approves the data exchange. If the user is offline, data exchange is not possible, which may cause problems in some scenarios, e.g., in long-lived services such as a magazine subscription [2]. To achieve an automation while preserving user centricity, in particular, user approval, our contributions are:

- We introduce an approach called *User-Controlled Automated Identity Delegation (UCAID)* that is based on an additional party called *Identity Delegate* and enables users to automate the dissemination of identity-related information restricted by user-defined policies.
- We cover the experience made with an implementation based on Windows CardSpace [3].

The paper is structured as follows: in Section II, we state fundamental requirements to cope with information consistency. Related work is discussed in Section III. In Section IV, we describe the concept of our approach and provide implementation details. Section V concludes the paper.

II. Fundamental Requirements

Fundamental requirements to cope with information consistency in user-centric FIM are:

User-controlled and automated attribute dissemination – in order to avoid inconsistencies, facilities are required that allow a controlled and automated dissemination of user attributes at any time. To preserve user centricity, disclosure of identity information should be controlled by the user.

Federating identities – establishing consistency between independent IdPs and SPs requires to support the linking of different user identities via so-called *identity federation*.

Coping with information heterogeneities – heterogeneous information schemas influence consistency in this respect that changes to information also have to be disseminated to information that might have heterogeneous representation but is semantically related [4]. Thus, mechanisms should be provided that allow to overcome these heterogeneities.

III. Related Work

In the following, we explain and analyze the user-centric concept in more detail using CardSpace for demonstration purpose. The use of UCAID is not limited to CardSpace and could also be used within Higgins [5] or OpenID [6].

Windows CardSpace [3] is Microsoft's implementation of the user client part of a user-centric FIM system. CardSpace represents a reference to an identity on an IdP as *information card*. When the user initiates a login at an SP supporting CardSpace, a client-side component named *identity selector* presents all information cards that match the SP policy to the user. The user accepts data dissemination by choosing an information card and by authenticating to the associated IdP. The attributes are disseminated within a security token, which is typically protected by cryptographic mechanisms.

Windows CardSpace is part of the Windows Identity Foundation (WIF) [3]. The prototypical implementation of UCAID leverages WIF and, in particular, the delegation mechanism of WIF, which works as follows: WIF allows a user to delegate a subset of her permissions to an SP by requesting a user token, called bootstrap token, at an IdP. Therefore, the IdP has to be configured to correlate the set of permissions that shall be delegated with the issued bootstrap token. The SP uses the bootstrap token when communicating with the IdP in its role as delegate. Based on the identity of the SP and the bootstrap token, which therefore represents a "certificate of authority", the IdP is able to determine the delegated permissions. These are packed in a so-called delegation token and can be used by the delegate to access other SPs on behalf of the user.

CardSpace does not allow for an *automated attribute dissemination* as it requires the user client to be involved in any data dissemination. *Information heterogeneities* are considered by Windows CardSpace 2.0 or by WIF respectively. *Federating identities* is not supported by CardSpace out-of-the-box, but could be enabled via a *linking service* introduced in [7]. The linking service was designed with attribute aggregation in mind, which refers to the process of unifying attributes for a single user from multiple IdPs [8]. As attribute aggregation requires to link identities, the linking service also enables identity federation. We do not leverage a linking service as major modifications to the identity selector would be required.

User-Managed Access (UMA) [9] enhances OAuth [10], a protocol that enables users to share information located at one service with another service in a controlled manner. UMA introduces a so-called *authorization manager* (AM) that allows to disseminate information in absence of the user. The AM decides if a data transfer is granted or denied restricted by user-defined policies. Using user-defined policies provides a reasonable approach for the automation of the user approval. However, in UMA the service provider and the consumer directly exchange attributes. This potentially causes privacy issues as it allows providers to gather information about a user. Furthermore, IdPs would have to be adapted to be able to interact with the AM as they are not able to interact with the AM out-of-the-box, which increases integration effort.

IV. User-Controlled Automated Identity Delegation

A. Description of the Approach

In this section, we present an approach called *User-Controlled Automated Identity Delegation (UCAID)* that provides the basis to cope with consistency in user-centric FIM systems. The approach introduces an additional party called *Identity Delegate* that acts on behalf of the user when services want to retrieve attributes, in particular, when the user is offline. Thereby, the delegate ensures two major aspects of *user centricity*: first, the delegate acts on behalf of the user in consideration of the user approval process by applying user-defined policies when a service wants to retrieve information, and second, the delegate acts in place of the user client in the attribute flow, i.e., instead of passing through the user client, the attribute flow passes through the delegate. UCAID does not require manual user interaction for the approval, instead, user approval is achieved by enabling the user to define policies, based on which the delegate is able to *automate* user approval.

The user trusts in the delegate to retrieve and disseminate attributes according to her policies. However, according to the principle of least privilege, we assume that the delegate is assigned only restricted permissions, so the required amount of trust in the delegate is limited. In particular, the concept assumes that IdPs are trusted to be configured to issue only attributes to the delegate that have already been encrypted and signed for requesting SPs. As FIM in general requires users and SPs to trust in IdPs, this is not increasing required trust. Therefore, the delegate acts as "identity relay" [8] that only gathers and forwards attributes and does not store attributes locally. Hence, an attacker taking control of the delegate is not able to read, store or alter user attributes.

Using the delegate involves different possibly recurring steps. In the *preparation step*, the delegate, designed as a service provided by a third party, has to be prepared by the operator to be usable for the user. The next step, named *registration step*, allows the user to state the IdPs from which the delegate should retrieve attributes. In the *authorization step* (see Fig. 1), the user specifies the policies, i.e., the user authorizes an SP to retrieve attributes from the delegate. Essential for achieving consistency is the last step, named *update step* (see Fig. 2), where the SP requests attributes from the delegate without requiring any manual user interaction.

In the following, we describe the approach in consideration of the different steps in more detail.

Preparation step – in this step, IdPs and SPs have to be introduced to the Identity Delegate, mainly to cope with information heterogeneities. For this purpose, the delegate operator and the providers have to establish mappings between their attribute schemas. Therefore, they identify semantically related attributes and define functions which transform these attributes into each other [4]. UCAID applies an intermediary schema that is only known to the delegate, thus, transformations are required that map semantically equivalent attributes between provider schemas and the intermediary schema.

Registration step – in this step, the user initially creates an identity on the delegate. After login, the user links this identity to her identities on all IdPs that shall be used to retrieve attributes from. This is done by consecutively authenticating against these IdPs using the mechanism of the underlying user-centric FIM system. By using mechanisms of the underlying system, a consistent user experience is ensured. When the user authenticates against an IdP, the Identity Delegate requires the IdP to include linking information for the user. This information is stored locally at the delegate and serves as reference to the IdP identity in later steps. The delegate confirms the registration by issuing the user her delegation identity. In case of CardSpace, the delegate issues an information card.

Identity linking impacts privacy if the linking information allows maliciously collaborating providers to correlate user identities. This enables them to aggregate information about a user, e.g., services delivered to the user could be tracked across multiple providers. A malicious correlation of user identities by linking information can be prevented if the linking information is only usable in the relationship of the two providers that federated identities, but has no meaning to a third provider. This requires a creation of an opaque, generated identifier whenever the user federates her identity on the delegate to an identity on a provider. When the user links her identity on an IdP with her identity on the delegate, the IdP is responsible for the creation of linking information and should be configured to consider privacy issues.

Thereafter, the user states which attribute should be retrieved from which IdP, i.e., she has to state the authoritative source for semantically related attributes (see [4]). Stating the authoritative source for semantically related attributes is required as identity-related information of a user is stored redundantly over different IdPs. For example, the *lastname* of a user is typically stored by multiple IdPs. Thus, it is required to configure the delegate which IdP is authoritative for this information. We like to note that letting the user state the authoritative source might raise level of assurance [11] issues because a user could state an authoritative source with a low assurance level potentially causing security issues with IdPs with higher assurance levels. Anyhow, as SPs and IdPs do know the issuer of an attribute, each SP or IdP is able to accept or decline received attributes.

Authorization step – in this step, the user is enabled to state which SP is allowed to retrieve which attributes from the Identity Delegate. We call these statements *dissemination policies*. A dissemination policy at least describes for each attribute which SP may consume this attribute. Finer-grained policies could be expressed by different policy description languages like PREP [12]. The expressiveness of the policies therefore depends on the policy language used.

The Identity Delegate enables the user to specify and administrate dissemination policies, and the delegate evaluates and enforces these policies whenever an SP requests attributes. Although it would be possible that IdPs store, evaluate or enforce dissemination policies, this would require to enhance existing IdPs with new functionality, thereby increasing the effort to integrate this approach into existing systems.

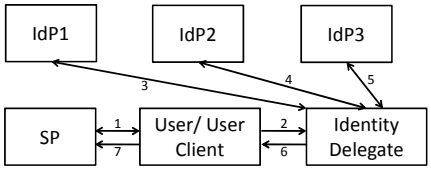

Fig. 1. Initial authorization step, shown by way of example in a scenario with three IdPs

From the user's perspective, a basic dissemination policy, i.e. a policy stating if an SP is allowed to retrieve requested attributes or not, only requires to log in at the SP and to acknowledge the authorization by using the mechanisms of the underlying user-centric FIM system. The authorization step has to be repeated for each SP that should be authorized. The information flow required for the authorization is depicted in Fig. 1. First, the user connects to an SP that is interested in retrieving user attributes from the delegate (1). The SP has to express this interest to the user client. If the user grants SP authorization, she authenticates against the Identity Delegate (2), which creates a dissemination policy for the SP. Thereafter, it retrieves the attributes from the authoritative IdPs (3–5). The attributes are forwarded to the user client (6), which finally sends them to the SP (7).

Furthermore, the authorization step serves to link the user identity on the SP with the identity on the delegate. When the user authorizes the SP (step (2) in Fig. 1), the Identity Delegate generates an opaque user identifier for the SP and sends this identifier along with the attributes to the user client (6). The client forwards the information to the SP (7). When the SP requests user attributes from the delegate in the update step, it includes the identifier in the request, and the delegate resolves the identifier to the respective opaque IdP identifier.

Update step – in this step, the SP periodically retrieves attributes via the Identity Delegate to achieve consistency in a relaxed manner, following a consistency model introduced in [4]. We chose this pull approach over a push approach as current user-centric FIM systems do not consider IdPs to push user attributes. Thus, facilitating a push approach requires to enhance existing IdPs with functionality, e.g., to manage which recipients to notify in case of an attribute change, and therefore increases integration effort.

The update step is shown by way of example in Fig. 2. It involves the SP requesting attributes from the delegate (1). The delegate authenticates the SP and evaluates the policy of the user. If the SP is authorized, the delegate requests the authoritative IdPs for the required attributes (2–4) and forwards the attributes to the SP (5).

As the delegate is required to forward user attributes without change, it cannot perform transformations, although it maintains the necessary transformation rules. To cope with this issue, the delegate sends relevant rules within the token to the SP. Thus, the SP can perform transformations itself without being forced to manage schema mappings.

As the attributes requested by an SP may be located at

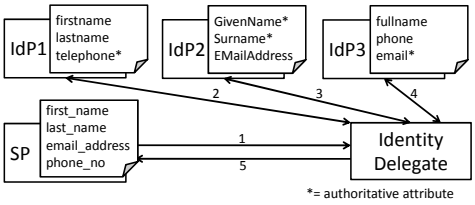

Fig. 2. Update of user attributes, shown by way of example in a scenario with three IdPs

different authoritative IdPs, the delegate supports attribute aggregation. In summary, the overall process is: (i) an SP requests attributes, (ii) the delegate finds the authoritative IdP and retrieves the attribute from this IdP, (iii) afterwards the delegate determines required transformation rules, (iv) finally, the delegate aggregates the attributes into one response.

B. Implementation Details

The prototypical implementation is based on Windows CardSpace 2.0 and Active Directory Federation Services (ADFS) 2.0 [3]. Unmodified instances of ADFS 2.0 were used as IdPs. An Active Directory (AD) was deployed as identity store. No changes were made to the identity selector. The delegate was developed from scratch. SPs, IdPs and the delegate authenticate using client and server certificates.

The delegate is implemented as ASP.NET web application with an additional WS-Trust Security Token Service (STS) and information card issuance functionality. It is initially configured with a list of trusted IdPs, a list of known SPs that do also trust the delegate, and transformation rules to map the correspondent schemas. The user registers with the Identity Delegate and leverages the web interface to federate identities on IdPs, to state authoritative IdPs for attributes and to administer dissemination policies. After finishing configuration, the user is issued an information card which can be used to authorize SPs or IdPs to retrieve attributes from the delegate.

The authorization step is started by the user initiating a CardSpace login at the SP. To indicate the interest in retrieving identity-related information from the Identity Delegate, an SP states a custom claim type named *LongLivedIdentifier*, which is included in the information card of the delegate, as required claim. In the prototype, the SP declares the delegate as trusted attribute source, so the identity selector allows the user to select the card of the delegate. By selecting this card, the user authorizes the SP. After retrieving the requested attributes from the respective IdPs, the delegate aggregates the encrypted tokens and adds a self-issued token including an opaque identifier in the *LongLivedIdentifier* claim. This identifier allows the SP to refer to a user when retrieving attributes from the delegate.

In order to enable the delegate to request claims for a user without knowing the credentials of the user on the IdP, we use the delegation abilities of WIF. Whenever the user federates an IdP with the delegate, the delegate saves the user token as bootstrap token. This token only contains a reference to the user identity on the IdP, but no further user attributes. When the delegate requests attributes from this IdP, it authenticates against the IdP and sends the bootstrap token. Based on the bootstrap token the IdP determines the delegated rights; if the rights are sufficient, the IdP sends the requested attributes in a delegate token encrypted for the respective SP or IdP.

First performance tests, e.g., to evaluate the overhead generated by the prototype, have been conducted. The results indicate that the approach is usable in real-world scenarios as attribute requests are answered in an acceptable time with modest network load.

V. Summary and Outlook

In summary, we argued that current user-centric FIM systems do not cope with information consistency sufficiently because of the way user centricity is enforced in current systems, in particular, the required manual user consent during the data dissemination process. The necessity of manual user consent may even thwart user control as users may tend to approve each dissemination without examination. We stated fundamental requirements to an adequate approach and examined CardSpace w.r.t. these requirements; CardSpace does not fulfill all requirements.

We presented an approach called *User-Controlled Automated Identity Delegation* that copes with information consistency in user-centric FIM systems by introducing an additional component that enables users to automate the dissemination of attributes based on user-defined policies. The approach allows to integrate providers with low integration and operation effort while preserving user centricity and considering privacy.

First efforts to evaluate our approach have been made. However, further evaluation efforts in consideration of the performance of the prototype are required to gain a better indication of the usability of the approach. In addition, operational aspects have to be analyzed in more detail.

References

[1] E. Maler and D. Reed, "The Venn of Identity," *IEEE Security and Privacy*, vol. 6, no. 2, pp. 16–23, 2008.

[2] T. Hoellrigl, J. Dinger, and H. Hartenstein, "FedWare: Middleware Services to Cope with Information Consistency in Federated Identity Management," in *ARES 2010: Proc. of the Fifth International Conf. on Availability, Reliability and Security*, 2010, pp. 228–235.

[3] Website of the Windows Identity Foundation. [Online]. Available: http://msdn.microsoft.com/en-us/security/aa570351.aspx

[4] T. Hoellrigl, J. Dinger, and H. Hartenstein, "A Consistency Model for Identity Information in Distributed Systems," in *Proc. of the 34th IEEE Computer Software and Applications Conf.* IEEE, 2010, pp. 252–261.

[5] Higgins. 2010. [Online]. Available: http://www.eclipse.org/higgins/

[6] OpenID Foundation. 2010. [Online]. Available: http://openid.net/

[7] D. W. Chadwick and G. Inman, "Attribute Aggregation in Federated Identity Management," *Computer*, vol. 42, no. 5, pp. 33–40, 2009.

[8] N. Klingenstein, "Attribute Aggregation and Federated Identity," in *SAINT '07: Proc. of the 2007 Intl. Symposium on Applications and the Internet Workshops*. IEEE, 2007, p. 26.

[9] M. Machulak, E. Maler, D. Catalano, and A. van Moorsel, "User-Managed Access to Web Resources," University of Newcastle upon Tyne: Computing Science, Tech. Rep., 2010.

[10] OAuth Core 1.0. 2010. [Online]. Available: http://oauth.net/core/1.0a

[11] P. Madsen and H. Itoh, "Challenges to Supporting Federated Assurance," *Computer*, vol. 42, no. 5, pp. 42–49, 2009.

[12] G.-J. Ahn and M. Ko, "User-centric Privacy Management for Federated Identity Management," in *COLCOM '07: Proc. of the 2007 Intl. Conf. on Collaborative Computing*. IEEE, 2007, pp. 187–195.