ALOC: Attribute Level of Confidence for a User-Centric Attribute Assurance

Salameh Abu Rmeileh, Esther Palomar, Hanifa Shah

ALOC: Attribute Level of Confidence for a User-Centric Attribute Assurance

Salameh Abu Rmeileh, Esther Palomar and Hanifa Shah

Birmingham City University, Birmingham, UK,
{salameh.aburmeileh,esther.palomar,hanifa.shah}@bcu.ac.uk

Abstract. The proliferation of online services leads to an increasing number of different digital identities that each user has. In order to enforce access control policies, service providers need assurance that the information associated with users' identities, either received by the user or a federation partner, are correct and trustworthy. Current identity assurance frameworks assess the trustworthiness of identity providers but do not define trust on the user attribute level of granularity. In this paper, we consider the user attribute as a dynamic structure that extends the foundation of attribute authenticity and trustworthiness by introducing the attribute level of confidence (ALOC). Basically, the ALOC encompasses additional information on attributes' lifecycle and issuing mechanisms. We present the architecture design and demonstrate its components. This paper concludes discussing future research directions.

Keywords: attribute assurance, attribute-based identity, trust, level of confidence.

1 Introduction

A significant amount of our daily activities have been replaced by their digital counterparts (online services) such as banking, social networking, and shopping, to name a few. As their real life former selves, some online service providers (SPs) must be able to identify and authenticate their consumers in order to make informed access control decisions while others can be more flexible depending on the nature of the service. However, as [19] state, "modern businesses see data as a gold mine." This reliance on data incite inappropriate practices. For online service providers the certainty of proving a subjects digital identity is limited by the strength and/or some level of trust on one or more authentication attributes. On the other hand, consumers need to trust SPs to handle their personal information properly [16].

However, todays online activities have evolved beyond what the username/ password format can effectively protect. Users then create multiple

digital identities, one for each service provider that requires an acceptable level of security when granting access to the service [13]. These identities may partly overlap, but can also be mutually inconsistent, e.g. shaving few years off their age or few pounds off their weight on online social networks, or a minor pretending to be a certain age in order to gain eligibility for certain entitlements. From users' perspective, this generates a huge burden for users to manage their identities, remember associated credentials, and keep their information up-to-date [4] while from a business perspective, SPs will suffer from critical deficiencies in their access control decisions in case of false or inconsistent information been provided by the users.

A digital identity is the information about individual characteristics, distributed in the digital world, by which an entity, a thing or person, in the real world can be recognised or known [22]. Technically, a digital identity comprises a limited set of attributes that holds the information about the entity's characteristics. These attributes are attested by some party (either the entity itself or a third party). Some attributes are for identification purposes and some others are not. While an attribute may not uniquely identify an entity, the aggregation of them could potentially cause entities to be uniquely identified within a scope. Identity is also *dynamic*. Assertions of someone's age, passport, email, phone number, friendships, convictions and beliefs change over time.

This leads to identity management (IDM) becoming one of the most pervasive parts of IT systems [20]. IDM comprises the whole processes and all underlying technologies for the creation (provision), usage, update and revocation (de-provision) of a digital identity once it is not needed anymore [3]. Thus, IDM systems are about controlling and using digital identities, enabling businesses to select and share user information [10]. Trying to overcome the limitations of traditional IDM models, Federated identity management emerged as a way of sharing identity information across several trust domains [1]. However, inherent issue in these open models is heavy reliance on online identity providers.

Identity assurance frameworks then appeared to assess the trustworthiness of identity providers [2]. As a result, identity providers obtain a level of assurance (LoA) that reflects the degree of confidence in the assertions they make. On the other hand, current identity assurance approaches do not consider the definition of trust at the level of attribute and mostly consider identity as a whole lacking distinction amongst different qualities of trust, and the ability to cope with changes of trust level

over time, e.g. attributes are gathered during the registration phase and often fixed.

In this paper, we propose a privacy-enhanced user-centric attribute assurance model to ensure that identity attributes are authentic and accurately associated with the user while enabling the user to have control over his attributes. We assign trust levels to individual attributes not only registration and authentication process and we extend the attribute structure to hold the aggregation of its assertions by introducing attribute level of confidence (ALOC). Basically, ALOC is a data structure, within the attribute itself, which comprises a set of elements reflecting the correctness, authenticity, timeliness and integrity of the attribute value. Being a fundamental component of the attribute structure, ALOC defines trust on the same granular level as the attribute information. In particular, ALOC utilises the attribute's usage history, combined with its life cycle[1] [12], to build its reputation and quantify its trust level. The user identity is then a unique dynamic structure compiling the attributes that can be selectively disclosed to the SP and according to a certain policy.

The rest of the paper is organised as follows. Section 2 overviews the related work. In Section 3 we present the new user attribute data structure. The architecture and our proposal integration into potential real scenarios and platform are described in Section 4. Finally, Section 5 concludes with the immediate future work.

2 Related Work

This work builds on and contributes to the fields of identity management, attribute assurance, attribute aggregation, trust, and privacy. Thus, related work has been investigated in these main different areas: trust and reputation systems, level of assurance and credibility of claims.

Several approaches have been proposed regarding trust levels for users' attributes. In [6, 8, 7], Chadwick et al. build on NIST's concept of assurance levels and propose to have separate metrics for the Registration LoA and the Authentication LoA instead of NIST's compound metric which is dependent upon both. The Registration LoA reflects the strength of the registration method the identity provider (IdP) used e.g. registering online is much weaker than registering in person while the Authentication LoA reflects the strength of the authentication method the IdP used e.g. username/password is weaker than public key certificates and private

[1] Attribute lifecycle consists of four phases: creation, usage, update, and revocation. These phases are inspired by the IDM life cycle.

keys. Prior to any authentication taking place, a user needs to register with a service, and provide various credentials to prove his identity. After successful registration the SP creates a profile for the user and may offer different authentication mechanisms for the user to access, such as username/password with Kerberos, username/password with SSL, etc. Thus, Chadwick et al. argue that no Authentication LoA can be higher than the Registration LoA, since it is the latter that originally authenticated who the user is. In [17], Mohan et al. propose the AttributeTrust framework for evaluating trust in aggregated attributes which are provided by trusted attribute authorities. Similar to our approach, in AttributeTrust, service providers are asked to provide attribute authorities with feedback after each successfully completed transactions. However, Mohan et al. do not outline differences between trust in attribute authority and the attribute itself. Compared to this, in our work, we assign trust levels to individual attributes not only registration and authentication process and we extend the attribute structure to hold the aggregation of its assertions. Additionally, the proposed approaches by Chadwick et al. and Mohan et al. assess the trustworthiness of IdPs not the individual attributes.

Several approaches to enhance the quality of attributes, user-centricity and privacy within national eIDs exist. In [14, 15], Laube and Hauser propose a service, as an extension to the SuisseID [9] infrastructure, to handle and provide, to some extent, assurance of personal attributes with no official authority certifying or owning. Similar to our approach, the MyIdP service reuses data the user has already used as part of a transaction with a web application in order to assess the quality and trustworthiness of the data. In particular, the assessment is based on the freshness of information, quality of the attribute issuer and the recurrence of information. In [21], Slamanig et al, propose an identification and authentication model to be applied for eIDs which allows for selective disclosure to better protect citizen's privacy. Both approaches rely on the existence of IdPs.

In [23], Thomas and Meinel propose a model to consider trust on a claim basis. Their approach is to extend the notion of claim in claim-based identity management by a *credibility* value enabling service providers to specify the expected trust quality for attributes and the required user attributes. However, the *credibility* value is based on two factors: (i) whether the issuer is trusted or not and (ii) whether the claim is verified or not limiting the attributes' trustworthiness to only three possibilities, namely trusted, untrusted, and a third vague possibility where claim is verified by an untrusted issuer. In our approach we go beyond the *credibility* value by considering other attribute properties resulting in a more compliant and

scalable level of confidence for user attributes and provide more choices for a SP to express its policy demands and for users to protect their privacy. In another study [24], Thomas and Meinel present an attribute assurance framework for federated identity management based on a verification context class. However, their attribute assurance framework only offers the possibility to express which attribute has been verified by a federated IdP using a particular verification method in addition to which attributes are required. By contrast, our approach offers users and SPs a wider spectrum to express their policy demands.

The aforementioned models do not address the attribute dynamic nature nor its lifecycle which in reality affects attribute's correctness and timeliness. Drawbacks of these existing solutions motivated us to design an improved, more fine granulated yet more applicable user-centric attribute assurance model.

3 Our Model

In this section, we propose a new user-centric attribute assurance model which is applicable for the public cloud in terms of data assurance and privacy. Our model enables the users to have full control over their information while providing SPs with assured and well qualified attributes.

3.1 Roles

The following entities are involved in the architecture:

User A set of attributes will allow system users to be authenticated and authorised. For authentication reasons, the user can reveal a subset of these attributes that are endorsed by different types of endorsers and mechanisms. the user can ask for endorsements as well as endorsing other users' attributes.

Endorser This represents a trusted certification authority, a TTP, a SP, a user, or combination of all. The endorser issues an endorsement to the user for a particular set of attributes using a specific endorsing mechanism.

Service Provider The SP offers different resources or services which require user identity information. Also, a SP can act as an endorser.

3.2 Requirements

The model fulfills the following requirements:

Assurance The model must provide qualified but not necessarily certified attributes utilising various mechanisms to quantify attributes' trust levels, i.e. timeliness, correctness, and reputation as well as certification and verification by either TTPs and also fully distributed mechanisms such as crowdsourcing.

User-centricity The user always remains in full control over his attributes and benefits from the selective disclosure property when the SP supports it. However, the user cannot alter the $ALOC$ part of the attributes structure. Nonetheless, the user can view all information about his attributes such as which attributes have been endorsed by whom.

Selective Disclosure In addition to disclose only required attributes' values to SP, the user must have the option to provide the SP with information about the attributes quality and authenticity.

Privacy The privacy of user attributes must be preserved in the presence of an honest but curious cloud storage, i.e., the cloud storage must not learn anything about the user's attributes, policies or transactions.

Dynamic & Self-Contained User attributes must be dynamic and self-contained by holding all necessary information required to assess their trustworthiness without the involvement of any third party during a given transaction. The model must not involve IdPs. The user acts as its own attribute provider and only interact with third parties that endorse their attributes.

3.3 Extending Identity Attributes

Digital identities are *dynamic*, so are the online services they interact with. Similarly, attribute must be *dynamic* too. Traditionally, an attribute is defined as an ordered pair of label and value, i.e. attribute $x = \langle l, v \rangle$, e.g. $\langle Name, John \rangle, \langle Address, London \rangle$. The current static structure of attribute does not provide any information about the authenticity, integrity, or correctness of the value the attribute holds. Our approach is

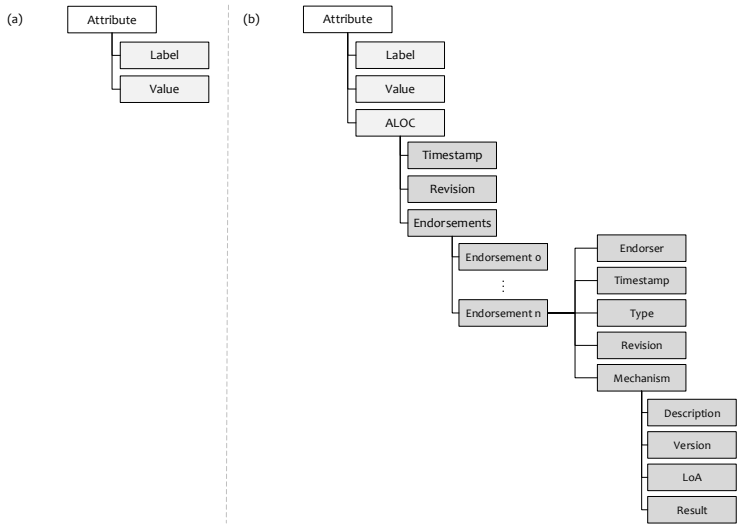

Fig. 1. Current (a) vs Proposed (b) Attribute Structure

not only to provide attribute assurance and trust but to also make attribute self-contained and more dynamic. We extend the current structure with third element, called *Attribute Level of Confidence (ALOC)*. This element holds all required information to express attribute trustworthiness as we believe this information must be part of the attribute itself and nowhere else. Therefore, we define a dynamic attribute as follows:

Dynamic Attribute is a data structure that holds some property or piece of data about an entity as well as the required information to verify the trustworthiness of this property or data. Technically, it is tuple of label, value and $ALOC$, i.e. attribute $x = \langle l, v, aloc \rangle$ where $ALOC$ is a data structure that comprises a set of elements, illustrated in Figure 1.

The following section explains $ALOC$ in detail.

3.4 Attribute Level of Confidence

Basically, $ALOC$ is a data structure which comprises a set of elements that are required to assess an attribute's trust level. These elements reflect the attribute completeness, timeliness, reputation, and authenticity. The following is a detailed explanation of $ALOC$ elements.

Timestamp This element represents the attribute's timeliness. It is important to measures the extent to which the age of an attribute is appropriate for the value it holds. We quantify timeliness as the time elapsed from the last update/revision of the attribute's value. In other words, It holds the date and time of the last revision.

Revision Represents the number of times the attribute's value has been changed, e.g. an attribute with a revision value of 7 shows that the attribute's value has been updated 7 times.

Endorsements This element holds a list of assertions by different entities, e.g. users, authorities, SPs. Each **endorsement** is a data structure comprising a number of elements as follows:

- **Endorser** Represents the entity that issued the endorsement.
- **Timestamp** The issuance date and time of the endorsement.
- **Type** This element represents the type of the endorsement. We have identified 4 types of endorsements, namely: *Creation*, *Registration*, *Authentication* and *Authorisation* endorsement, see Section 3.5.
- **Revision** Shows the attribute's revision the endorsement has been issued for.
- **Mechanism** Contains information about the certification mechanism used by the endorser.
  - **Description** Defines the type of certification mechanism, e.g. X.509 certificate, username/password token, SAML assertions, digital signature, etc.
  - **Version** The version number of the mechanism.
  - **LoA** Level of assurance of the mechanism indicates the strength of the mechanism, e.g. a X.509 certificate will have a higher LoA than a username/password token.
  - **Result** This element holds the result of the endorsing mechanism, e.g. if the mechanism is a digital signature then result is the signature itself.

Issuance and acceptance of endorsements depend on the *ALOC* policies of both the endorser and the attribute owner.

3.5 Endorsements Types and Mechanisms

We define four types of endorsements based on the phases of the attribute life cycle. Firstly, *Creation* endorsement is attached to the creation phase

of the attribute. Secondly, we break down attribute's usage phase into 3 endorsements: *Registration*, *Authentication*, and *Authorisation*. We will further elaborate on these phases' processes in Section 4.2. For example, attribute 'Academic Qualification' gets a *Creation* endorsement issued by the corresponding academic institution upon the user request; when the user registers with a SP he gets a *Registration* endorsement. A single authority may not be able to certify every attribute and not all attributes have certifying authorities. Thus, the user is able to get endorsements not only from trusted third parties and SPs but also from referral-based trust models such as crowdsourcing.

A wide range of mechanisms are used for adding endorsements namely digital signature, reputation systems, voting schemes, or referral-based mechanisms. The result of the mechanism, i.e signature, score, value, is stored in the *Result* element of the endorsement.

SPs also define own classifications and trust of endorsers, endorsement mechanisms and mechanisms' results expressed using *ALOC* policies.

4 Architecture

System architecture comprises 4 components: User-centric *ALOC* Agent, Service Provider, Secure Storage, and the User. Figure 2 depicts the proposed architecture. The designed architecture adopts Kim Cameron's Laws of Identity[5] that are widely accepted as a guideline for the design of identity systems.

4.1 Components

Components of the architecture are explained in detail.

Attribute Token Is an encrypted token containing a set of attributes' elements to be sent to a SP by the user. The attributes' elements contained in the token are based on the user *ALOC* policies.

ALOC Policy Represents the storage for the user to express rules and semantics of the attribute information to reveal, to whom, and under what conditions. The SP can also establish its own policies to express the conditions under which a service can be accessed. See Section 4.3.

Endorsement Token For every endorsing process, an endorsement token is created by the issuer containing the elements depicted in 1.

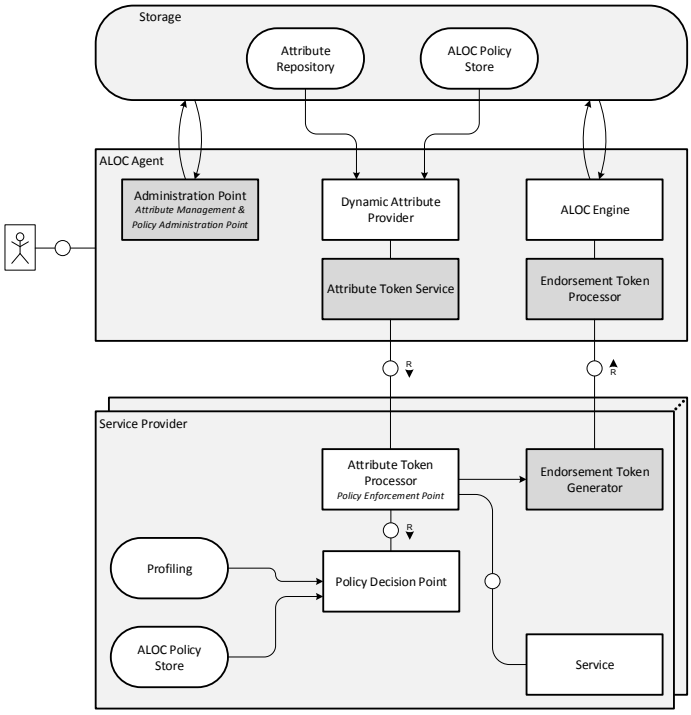

Fig. 2. Proposed Architecture for ALOC-based User-centric Attribute Assurance

ALOC Agent ALOC Agent is a software application that implements the Attribute Level of Confidence model presented in Section 3.4 and comprises several components:

- **Administration Point** This component enables the user to manage his attributes (view, add, update, revoke), define own *ALOC* policies, and request endorsements from other entities.
- **Dynamic Attribute Provider** This component builds a set of dynamic attributes based on the requested attributes and on dependence of the user's *ALOC* policies.
- **Attribute Token Service** This component is responsible to provide the Dynamic Attribute Provider with the requested attributes by the SP and to provide the SP with an encrypted token which contains what the Dynamic Attribute Provider returned.
- **Endorsement Token Processor** This component parses and validates endorsement tokens received from a SP and pass it on to the *ALOC* Engine.

- **ALOC Engine** This component is responsible for updating the user's attributes *ALOC* element after receiving information from the Endorsement Token Processor according to the user's *ALOC* policies.
- **Storage** In order to preserve the users' privacy without violating the 8th Law of Location Independence [11] (which states that IDMs should not rely on any persistent data stored locally at the user's machine), the users' dynamic attributes and ALOC policies are stored encrypted on the public cloud. The cloud storage comprises two persistent storage components. The information stored in both components can only be accessed through an ALOC Agent. The data is encrypted by the ALOC Agent before being sent for storage; none of the data is revealed to the storage provider.

Service Provider The SP consists of several components:

- **Attribute Token Processor/PEP** This component is responsible for validating attribute tokens. In order to decide whether the information in the token is accepted or not, a request is sent to the Policy Decision Point component. If the information is accepted the Attribute Token processor provides the Endorsement Token Generator component with required information to issue an endorsement token and send it back to the user's *ALOC* Agent.
- **Policy Decision Point** The Policy Decision Point compares the token information it receives to the *ALOC* policies of the service provider.
- **Endorsement Token Generator** This component issues encrypted endorsement tokens back to the user's *ALOC* Agent based on requests from the Attribute Token Processor.

4.2 Processes

Creation and Management of Attributes A user access the ALOC Agent (Mobile app, web app, browser extension) using a passphrase, a smartcard, or a biometric credential. The user creates and manages (view, update, delete) his attributes, i.e. attributes values and ALOC policies through the Administration Point within the *ALOC* Agent.

The user is able to set label and value for an attribute. Also, the user can send a request to a TTP, SP, another user, or collection of them asking for a *Creation* endorsement depending on the attribute type

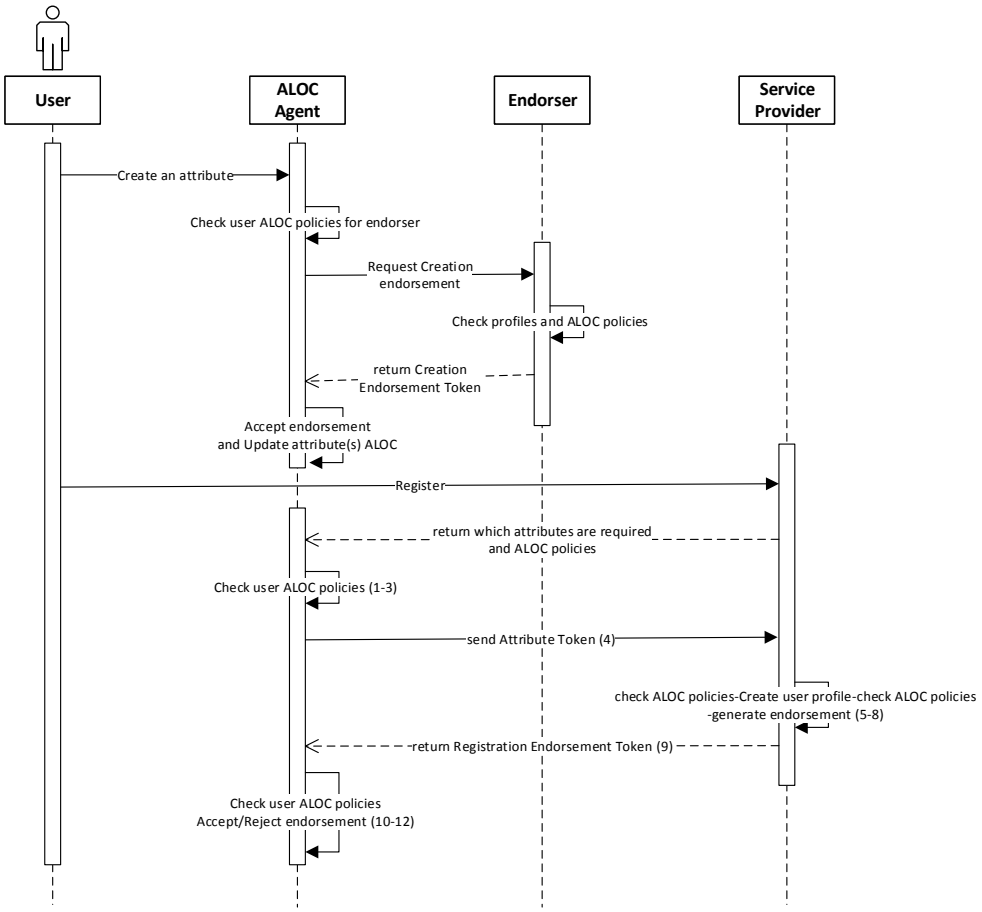

Fig. 3. Attribute Creation and Registration Endorsing Processes

and the user's ALOC policies. Figure 3 shows the attribute creation and registration endorsing processes.

Endorsing Process The following steps describe the endorsing process for the *Registration* endorsement. See also Figure 3 .

When the user register with a SP:

1. The Attribute Token Service asks the Dynamic Attribute Provider for a set of the requested attributes by the SP.
2. The Dynamic Attribute Provider checks the user *ALOC* policies and accordingly builds the set of requested attributes from the Attribute Repository.

3. The Dynamic Attribute Provider returns the set of dynamic attributes to the Attribute Token Service to issue a token.
4. The Attribute Token Service creates a token, in the requested format, which contains the set of dynamic attributes and sends it to the SP.
5. Upon receipt, the Attribute Token Processor parses the token, checks its validity then passes the information within the Token to the Policy Decision Point.
6. The Policy Decision Point checks the SP *ALOC* policies against the received information.
7. The SP applies the required verification mechanism.
8. Upon success of steps 6 and 7, the Attribute Token Processor requests the Endorsement Token Generator to issue a *Registration* endorsement token for the accepted attributes after providing it with the required information, i.e. the accepted attributes and the endorsing mechanism used.
9. The Endorsement Token Generator issues a token, which contains this information, signs it and then sends it back to the user's *ALOC* Agent.
10. The Endorsement Token Processor receives it, parses it, verifies and checks its validity, and then passes it on to the *ALOC* Engine.
11. The *ALOC* Engine checks the user's *ALOC* policies and accordingly decides whether to accept the endorsement or not.
12. If the endorsement is accepted, the *ALOC* Engine computes the attributes *ALOC* elements, i.e. weight and endorsements, and applies it the Attribute Repository.

The aforementioned steps also apply for the authentication and authorisation processes. However, the endorsement type changes to *Authentication* or *Authorisation*. Additionally, in the authentication process step (8) the Attribute Token Processor checks if the received dynamic attributes have a *Registration* endorsement by the SP, otherwise it applies a required verification mechanism. Lastly, to securely and reliably support these processes, we assume that whenever we speak of public parameters or public keys, they are available in an authentic fashion, e.g., via a PKI. Furthermore, the channels between all parties provide confidentiality, as well as authenticity, e.g., via the use of TLS.

4.3 ALOC Policies

ALOC policies are a crucial part of the ALOC trust model as they enhance the decision making process at SPs, and the selective and/or minimal disclosure of users' attributes. There are two types of *ALOC* policies

namely SP *ALOC* policies and user *ALOC* policies. Whereas the latter allows the users to control what to reveal to SPs the former allow SPs to define the trust requirements for user attributes.

User ALOC Policies Users can create their own policies to control information disclosure and endorsement acceptance. For instance, a user may create a policy that constraints what can be revealed out of the attribute data structure, e.g. a policy for particular when interacting with particular SPs to only disclose the attribute's label, timestamp, revision, and endorsements. Though the value of the attribute is not revealed, the SP will have some degree of assurance based on the *ALOC* elements. Additionally, the user can create policies for endorsement acceptance, e.g. a user policy for attribute *DateOfBirth* forces to be endorsed for *Creation* by *An Interior Ministry* or attribute *Fullname* can be endorsed for all endorsement types by any endorser.

SP ALOC Policies SPs can create policies for particular attribute information to be verified. In other words, a SP may create a policy that expresses what attribute elements required to compute the level of trust (confidence) of an attribute. For example, the SP expresses its requirements for 2 attributes (Fullname and DateOfBirth) to access a particular resource. For the Fullname attribute the SP enforces exactly one policy, in this case (A) which expresses that the attribute value is required, and it is willing to endorse the attribute if requirements are met. For DateOfBirth attribute the SP enforces 2 policies, A and B, and requires both policies to be met. In policy A the SP expresses that providing the value of the attribute is optional leaving the decision to the user. However, in policy B the SP requires the attribute weight, revision, and an endorsement from a particular endorser by a certain endorsing mechanism whether the user provides the attribute value or not.

In case of a conflict between a user policy and a SP policy then it is up to the user to make a decision to change his policy or not.

5 Conclusion and Future Work

In this paper, we dealt with attribute assurance by extending the structure of digital attributes and defining trust within the attribute itself without the involvement of a third party during a given transaction. We consider the user attribute as a dynamic data structure that extends the

foundation of user attribute authenticity and trustworthiness by introducing the attribute level of confidence (ALOC). ALOC enables a multi-level selective disclosure where the user can reveal particular attributes' elements that can be to the SP and according to a certain policy. We also proposed a user-centric attribute assurance architecture based on ALOC. An ALOC agent can be locally-installed software running on a user's device, or its functionality can be distributed between a local and cloud-based entities to reach a higher level of security and accomplish the 8th law of identity, although the problem of protecting stored data on cloud is out of our work's scope. We also showed that our architecture does not require major changes to SPs, however, requires the addition of certain components.

Our implementation consists of developing a user side and a SP side applications. For the SP side, we are developing a simple online store web service that requires user registration and authentication. For the user side, we are developing a browser add-on where the user manages his attributes, attribute endorsements and policies. In our design the user's registration, authentication and policy negotiation will be managed by the add-on. Thus, the user does not need to create a username, a password or provide attributes or ALOC elements manually, e.g. when the user clicks the registration button in the service the service and the add-on will establish communication, exchange policies, and then make a decision either the user will be registered or not based on both policies and provided ALOC elements. *ALOC* policies will adopt XACML policy language standard [18]. Currently, both parts of the implementation are under development and will be evaluated against the requirements mentioned in Section 3.2.

Although our architecture satisfies the major properties of a user-centric IDM system, there are still some properties to be improved. It is our immediate future work to finish the implementation of both, the SP and user sides, to evaluate the architecture usability, security, and test the ALOC policy negotiation between users and SPs as well as to making the endorsement mechanism strong against collusion and Sybil attacks.

References

1. Alpár, G., Hoepman, J., Siljee, J.: The identity crisis. security, privacy and usability issues in identity management. arXiv preprint arXiv:1101.0427 pp. 1–15 (2011),
2. Baldwin, a., Baldwin, A., Mont, M., Mont, M., Shiu, S., Shiu, S.: On identity assurance in the presence of federated identity management systems. In: Proc. of the 2007 ACM workshop on Digital identity management. pp. 1–19. No. 1, ACM Press, New York, New York, USA (2007)

3. Bertino, E., Takahashi, K.: Identity Management: Concepts, Technologies, and Systems (2011),
4. Bertino, E., Martino, L., Paci, F., Squicciarini, A.: Security for Web Services and Service-Oriented Architectures. Springer-Verlag, Berlin, 1 edn. (2010)
5. Cameron, K.: The laws of identity. Microsoft Corp (May) (2005),
6. Chadwick, D.W., Inman, G.: Attribute Aggregation in federated identity management. Computer 42, 33–40 (2009)
7. Chadwick, D.W., Inman, G.: The Trusted Attribute Aggregation Service (TAAS) - Providing an attribute aggregation layer for federated identity management. In: Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES. pp. 285–290 (2013)
8. Chadwick, D.W., Inman, G., Klingenstein, N.: A conceptual model for attribute aggregation. Future Generation Computer Systems 26(7), 1043–1052 (2010)
9. ECH: eCH-0113: Spezifikation SuisseID. Tech. rep., eCH (2012)
10. Hansen, M., Berlich, P., Camenisch, J., Clauß, S., Pfitzmann, A., Waidner, M.: Privacy-enhancing identity management. Information Security Technical Report 9(1), 35–44 (2004),
11. Hoepman, J.h., Joosten, R., Siljee, J.: Comparing Identity Management Frameworks in a Business Context
12. Jensen, J.: Identity Management Lifecycle - Exemplifying the Need for Holistic Identity. Information and Communication Technology pp. 343–352 (2013)
13. Jøsang, A., Fabre, J., Hay, B.: Trust requirements in identity management. Proceedings of the 2005 ... pp. 99–108 (2005),
14. Laube, A., Hauser, S.: myIdP-The Personal Attribute Hub. In: The Fifth International Conferences on Advanced Service Computing. pp. 1–5 (2013)
15. Laube, A., Hauser, S.: myIdP-The Personal Attribute Hub: Prototype and Quality of Claims. International Journal On Advances in Intelligent Systems 7(1), 1–10 (2014)
16. Lopez, J., Oppliger, R., Pernul, G.: Authentication and authorization infrastructures (AAIs): a comparative survey. Computers & Security 23(7), 578–590 (oct 2004),
17. Mohan, A., Blough, D.M.: AttributeTrust - A framework for evaluating trust in aggregated attributes via a reputation system. In: Proceedings - 6th Annual Conference on Privacy, Security and Trust, PST 2008. pp. 201–212 (2008)
18. OASIS: eXtensible Access Control Markup Language (XACML) Version 3.0 (2013),
19. Q Ethan McCallum, K.G.: Business Models for the Data Economy. O'Reilly (2013)
20. Sharman, R.: Digital Identity and Access Management: Technologies and Frameworks: Technologies and Frameworks. IGI Global (2011)
21. Slamanig, D., Stranacher, K., Zwattendorfer, B.: User-Centric Identity as a Service-Architecture for eIDs with Selective Attribute Disclosure. ACM Symposium on Access Control Models and Technologies (SACMAT) (2014)
22. Suriadi, S.: Strengthening and Formally Verifying Privacy in Identity Management Systems. Ph.D. thesis, Queensland University of Technology (2010)
23. Thomas, I., Meinel, C.: Enhancing Claim-Based Identity Management by Adding a Credibility Level to the Notion of Claims. 2009 IEEE International Conference on Services Computing pp. 243–250 (2009)
24. Thomas, I., Meinel, C.: An Attribute Assurance Framework to Define and Match Trust in Identity Attributes. 2011 IEEE International Conference on Web Services pp. 580–587 (jul 2011)