Accepted Manuscript

Privacy-preserving attribute aggregation in eID federations

Walter Priesnitz Filho, Carlos Ribeiro, Thomas Zefferer

PII: S0167-739X(17)32796-6
DOI: https://doi.org/10.1016/j.future.2018.09.025
Reference: FUTURE 4460

To appear in: *Future Generation Computer Systems*

Received date : 21 December 2017
Revised date : 21 July 2018
Accepted date : 9 September 2018

Please cite this article as: W. Priesnitz Filho, et al., Privacy-preserving attribute aggregation in eID federations, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.09.025

Privacy-preserving Attribute Aggregation in eID Federations

Walter Priesnitz Filho[a,*], Carlos Ribeiro[a,1], Thomas Zefferer[b,1]

[a] *Instituto Superior Técnico, Universidade de Lisboa*
Av. Rovisco Pais, 1, 1049-001 Lisboa, Portugal
[b] *Secure Information Technology Center Austria*
Inffeldgasse 16a, 8010 Graz, Austria

Abstract

Personalized electronic services, e.g. from the e-government domain, need to reliably identify and authenticate users. During user-authentication processes, the electronic identity of the respective user is determined and required additional attributes, e.g. name and date of birth, linked to this identity are collected. This attribute-collection process can become complex, especially if required attributes are distributed over various attribute providers that are organized in a federated identity-management system. In many cases, these identity management systems rely on different ontologies and make use of different languages. Hence, identity federations, such as the one currently established across the European Union, require effective solutions to collect user attributes from different heterogeneous sources and aggregate them to a holistic user facet. At the same time, these solutions need to comply with minimum disclosure rules to preserve users' privacy. In this article, we propose and introduce a solution for privacy-preserving attribute aggregation. Our solution combines attributes from different domains using ontology alignment and makes use of locality sensitive hashing functions to preserve users' privacy. Evaluation results obtained from conducted experiments demonstrate our solution's advantages for both, service providers and users. While service providers can be provided with a larger set of attributes, users remain in full control of their data and can decide on which of their attributes shall be revealed.

Keywords: Electronic identity, Identity federation, Attribute aggregation, Interoperability, Ontologies, Privacy

1. Introduction

Governments and public administrations face the challenge to continuously improve their e-government infrastructures in order to cope with fast-changing requirements and to provide citizens useful electronic

*Corresponding author
Present address: Universidade Federal de Santa Maria - Av. Roraima, 1000 Prédio 5 97105-900 Santa Maria - RS, Brasil

Email addresses: walter.filho@tecnico.ulisboa.pt (Walter Priesnitz Filho), carlos.ribeiro@tecnico.ulisboa.pt (Carlos Ribeiro), thomas.zefferer@a-sit.at (Thomas Zefferer)

ACCEPTED MANUSCRIPT

services. During recent years, interoperability between e-government solutions has been on the agenda of many public-sector organisations [1]. In particular, achieving interoperability between different national electronic identity (eID) solutions has been a topic of growing interest, as electronic identification and authentication are crucial building blocks of transactional e-government services.

The European Union (EU) and its Member States (MS) are a prime example of this. For many years, EU MSs have developed and rolled out country-specific eID solutions independently from each other. As a result, citizens from, for example MS A have been unable to use their eIDs to authenticate at e-government services provided in MS B, undermining the idea of a converging European society and a digital single market. To solve these issues, the EU has been committing efforts to the study of heterogeneity in existing European eID systems and the legal implications that need to be addressed when these systems aim to become interoperable. An example of the efforts committed to achieve interoperability between European e-government and eID solutions are the EU-funded Large Scale Pilots (LSP) eCodex[1], epSOS[2], PEPPOL[3], SPOCS[4], STORK, and STORK 2.0[5]. Their goal is to bring interoperability to different public-sector domains such as justice, health care, and procurement. With regard to eID, the LSPs STORK and STORK 2.0 are especially worth mentioning, as they have yielded a first interoperability solution for national eID systems by developing an identity federation (IF) framework.

In general, an IF can be regarded as an association of multiple identity systems (ISs). An IF defines a set of common attributes, information-exchange policies and sharing services, allowing for cooperation and transactions between IF members, i.e. between different identity systems [2]. An IS, in turn, typically contains, at least, a user, an Identity Provider (IdP), and a Service Provider (SP) acting as Relying Party (RP) [3, 4]. The IdP establishes, maintains, and secures the electronic identity linked with a subject (i.e. the user), and may also confirm the identity of that subject. From a technical perspective, the confirmed identity of a subject comprises at least a unique identifier and a set of additional attributes such as first name, family name, or date of birth. The RP makes transaction decisions based upon receipt, validation, and acceptance of a subjects confirmed identity within the Identity System (IS). This way, SPs assuming the role of RPs can control access to their services and resources. In addition, an IS can also comprise one or more Attribute Providers (APs). An AP stores additional attributes for users. These attributes optionally enrich the user's confirmed electronic identity. If required, SPs can request attributes for identified users from APs being part of the same IS.

The goal of an IF is to achieve interoperability between different ISs. An IF guarantees that IdPs, SPs/RPs, and APs from different ISs can interact with each other based on a defined attribute set. While

[1]https://www.e-codex.eu/
[2]https://www.epsos.eu/
[3]https://peppol.eu/
[4]https://www.eu-spocs.eu/
[5]https://www.eid-stork2.eu/

Table 1: Ontologies from the Attribute Providers - Example 1

$Ontology_{AP_A}$	$Ontology_{AP_B}$
Address	Birthday
BloodType	Blood
DateOfBirth	E-mail
Email	GivenName
Nationality	Occupation
FamilyName	PassportNumber
GivenName	PhysicalAddress
Identification	SSN
MaritalStatus	Sex
Occupation	Surname
Phone	Telephone
Sex	
Title	

this works fine in theory, problems arise in practice, when e.g. attributes required by SPs exceed the set of common attributes defined by the IF. This is especially problematic in scenarios, where attributes from different (federated) IS need to be merged. For instance, there might be SPs inside a federation demanding user facets comprised of attributes managed by multiple APs originating from more than one IS. If returned attributes are not part of an agreed attribute set, the SP is unable to assign attributes to the correct user. Specifically, returned attributes could be seen as several facets, one from each IdP or AP, belonging to different users, rather than a single facet of one user containing all the attributes required by the SP.

Such scenarios make necessary an effective attribute-merging process. Achieving such a process requires finding intersections between attribute sets provided by different APs. This can only be achieved if there is a common vocabulary to match attributes from these various APs. This is best illustrated by means of an example: Consider a user attempting to access a service provided by an SP and being asked by the SP to provide the attribute set $C = \{$*Address, Birthday, Name, Social Security Number (SSN)*$\}$. The user's name is *João*, his birthday is *1987-05-21*, his Address is *Av. Example, 3*, and his SSN is *496-32-6450*. The user has attributes stored in his own country (e.g. Portugal) and also in a foreign country, e.g. Austria. Consider A, for required attributes provided by AP_A (Austria), and B, for required attributes provided by AP_B (Portugal), as shown in Table 1. Summarizing, the SP requires the Attribute Set C to grant access to the user, but the Set C belongs to more than one AP ($C = A \cup B$).

If the intersection $I = A \cap B$ uniquely identifies the user, then it would be possible to deduce, by transitivity, that all attributes in C belong to the same user. If I is the empty set, or if it is not sufficient to uniquely identify the user, there is no way to unambiguously confirm the user's identity. Taking, for instance, the subset *Name* and *Birthday* from C, it is not possible to guarantee that there is only one *João* with the birthday *1987-05-21*. However, it may be possible to define $A^* \supset A$ and $B^* \supset B$ such that $I^* = A^* \cap B^*$ identifies the user unambiguously. A^* and B^* are supersets of A and B, respectively, containing attributes available at the APs but not asked by the SP (e.g. Surname).

The problem with this approach is that the Set $C^* = A^* \cup B^*$, (*Address, Birthday, Name, SSN* and ***Surname***), exceeds what the SP actually requires. Providing the SP with the additional attribute ***Surname*** would hence violate the minimum-disclosure rule and compromise the user's privacy. Thus, even though sufficient attributes would be available to unambiguously identify the current user, these attributes must not be used.

Our proposal uses the supersets (e.g. Surname in the above example) to merge attribute sets, but hides the exact values of the additional attributes used to preserve privacy. At first glance, Zero Knowledge Proof of Knowledge (ZKPK) or Homomorphic Encryption appear to be appropriate approaches to reach this goal. However, as attributes are based on different vocabularies in the given use case, these techniques cannot be applied directly. We show that Locality Sensitive Hashing (LSH) functions adequately address this issue, as demonstrated in [5]. LSH functions are ideal to preserve privacy while still enabling comparisons. However, they cannot directly solve the problem of intersecting attribute sets to find the common universal identifier facet. This requires comparing attributes from several APs, which may be arduous due to source heterogeneity [6].

Ontologies appear to be a promising approach to tackling this issue, as they foster sharing and reuse of knowledge [7]. An ontology is a specification to use a certain terminology so that it is consistent with the theory defined in that ontology. The problem is that when dealing with diverse ISs within an identity federation, it is unlikely that they all employ the same ontology to describe their information [8]. Furthermore, it is also unlikely that they even use the same language in their ontologies. This especially applies to real-world use cases such as pan-European identity federations as targeted by the EU.

Intersecting attribute sets from different APs relying on different ontologies and languages thus raises the demand for an appropriate ontology-mapping solution. In general, ontology mapping deals with the need to reconcile ontologies that cover similar domains of knowledge but use different nomenclatures [8]. Priesnitz et al. [9] have assessed and ranked different ontology alignment solutions according to their effectiveness for the given scenario. Based on this and other previous works, we propose a solution for privacy-preserving attribute aggregation in identity federations.

1.1. Open Issues

Previous work has revealed that there are solutions available to promote ontology alignment [9]. Furthermore, there are also solutions available to assess the similarity between attributes based on blinded attribute values. However, so far no work has combined and employed these building blocks in a similar context, i.e. to aggregate user attributes in identity federations. Thus, there are still some open issues regarding ontology aligning in the presence of heterogeneity and attribute merging in an IF context. In this section, we describe those identified.

1.1.1. Interoperability

Ontologies constitute a valuable knowledge-sharing resource. Still, there are some open issues regarding their potential in identity federations. Using ontologies to represent knowledge in this context is an important direction to achieve a consistent path towards the reliable exchange of user data. That way, extending their usage in such a context could help in promoting semantic interoperability among involved entities of an IF. Even if IF entities already use ontologies to represent knowledge, it is improbable that they use the same ontology and the same language. This also applies to new entities joining these federations. It is not expectable that they all use the same language to develop their ontologies. Adjusting the different languages in the parties' ontologies to a common one and aligning them is hence a considerable improvement to achieve semantic interoperability. Our solution proposed in this article addresses this issue.

1.1.2. Alignment Quality

In identity federations, entities usually interact with each other multiple times when exchanging user data. Accordingly, these several interactions can provide different levels of accuracy regarding the user data they exchange due to inevitable mistakes such as misspellings and abbreviations. Thus, assessing the accuracy of exchanged data and feeding the alignment relations with those metrics each time two entities are interacting, can improve the confidence level. Our solution addresses this issue. As far as we are aware, no other work uses this kind of feature to improve the confidence level of exchanged and aligned data.

1.1.3. Alignment Alternatives

An alignment may result in no correspondence between key attributes from two APs despite having the same user authenticated on both APs. In this case, it is not possible to establish a relation between those APs, even if they are part of an identity federation and share data about the same group of users. Using the relationships already stored can help in finding an AP chain, linking the two APs and providing the user attributes required by an SP. The solution proposed in this article addresses this issue by establishing chains of APs.

1.2. Contribution

In this article, we propose a novel privacy-preserving approach to aggregate attributes within an identity federation. The proposed solution relies on LSH functions and ontology-alignment approaches. Based on these fundamental technologies, our proposal comprises the following features:

- an aligning history function (HF), which uses previous confidence level assessment values to increase the reliability of the current CL;
- a third party attribute provider (AP_T) approach that allows the establishment of a chain of APs improving the confidence on the user identity, since he/she has key identifiers, among the involved APs;
- a multi-language strategy to handle several languages in identity federations' ontology definitions; and
- an attribute-blinding method based on Locality Sensitive Hashing (LSH) functions to preserve users' privacy during attribute-aggregation processes.

The combination of all these features yields a comprehensive solution for privacy-preserving attribute aggregation. This way, this work contributes to improved user identification and authentication processes in identity federations.

1.3. Structure

This paper is structured as follows: Section 2 and Section 3 provide relevant background information and survey related work. From this survey, open issues are identified that are not covered by existing solutions. Our proposal to address these issues is introduced in Section 4. A concrete implementation of the proposed solutions is presented in Section 5. In Section 6, we evaluate our solution by means of several experiments. Finally, conclusions are drawn in Section 7.

2. Background

Identity Systems (ISs) manage information used to identify a user in a given environment (e.g. eID systems in e-government portals). Interoperability among ISs deals with exchanging user attributes allowing the user to use a specific service outside his/her system (e.g. public services). When one IS sends these attributes to another IS, it is mandatory to keep user data private, disclosing just what is necessary for the execution of the respective action. To promote interoperability among ISs by means of an identity federation, there must be a common base of concepts to be used by all IS that wish to interact.

Ontologies are used to represent knowledge, but it is improbable that federated ISs use the very same ontology to represent the same knowledge [8]. Therefore, to allow ISs to communicate using a common knowledge base requires a mechanism that analyzes all possible knowledge representations in involved ISs and merges them into a unified one to be used in this communication process. The solution proposed in this article accomplishes this task. In the following, fundamental concepts used by the proposed solution are briefly sketched.

2.1. Interoperability

The absence of machine-readable descriptions impacts the quality and the efficiency of electronic services. This, in turn, increases administrative burdens and makes the provision of services more expensive. Public Service (PS) descriptions delivered through e-Government portals are usually unstructured and not machine-readable [10], which makes it hard for them to become interoperable.

Data interoperability, in an e-government context can be defined as the capability of all interacting participants to access, reuse, and understand data in both human-to-machine and machine-to-machine formats [11]. Different representations, languages, purposes and syntaxes must be reconciled to reach a common understanding of the datas meaning and to achieve data interoperability. Interoperability is the ability of organisations to interact towards mutually agreed common goals. They interact sharing information and knowledge, through the business processes they support, exchanging data between their respective Information and Communications Technology (ICT) systems [12].

There are four distinct types of data interoperability [11]: technical, syntactic, organisational, and semantic. This work focuses on semantic interoperability. Semantic interoperability means that datasets have a common understanding of terminology: the same term means the same or these datasets apply that term in the same way.

Semantic Web uses ontologies to define knowledge to address the interoperability issue [13]. Semantic Web aims to extend current interfaces in a standardized machine-readable format, adding annotations for knowledge description to reach interoperability. Semantic interoperability (SI) depends on the services interfaces description and how the services clients share the meaning of the information [14].

2.2. Identity Provider Proxy

An Identity Provider Proxy (IdPP) centralizes integration of federated eID tokens by carrying out the authentication for the SP [15]. The SP does not take any action regarding any integration of eID tokens. The IdPP (being a data controller or data processor) handles the data protection aspects.

For example, in the STORK project application context, there is one proxy service per Member State that handles its eIDs and SPs [15]. The Pan-European Proxy Service (PEPS) comprises two components: S-PEPS and C-PEPS. The S-PEPS is located in the country of the SP and handles the authentication process, redirecting the authentication requests of foreign citizens to their C-PEPS. The C-PEPS, which is located in the citizens country, carries out the authentication of its citizens. The C-PEPS asserts successful authentications and sends them back to the S-PEPS, which asserts them to the SP.

2.3. Ontologies

Scenarios, where each member of an identity federation has its own knowledge representation, require a standard mechanism to represent this knowledge in order to support interoperability. Ontologies are used to provide such a standard resource when formalizing knowledge.

An ontology is an agreement for describing a common model to be shared among administrative and non-administrative parties. This agreement permits information exchange in a human-readable and understandable manner [16].

Ontologies can improve SI by adjusting various terms to make them useful in several applications. Ontologies also provide structured vocabularies describing a formal specification of shared concepts in a given domain, contributing to solving semantic heterogeneity. Despite being a useful resource to promote semantic interoperability, ontologies matching and merging constitute the main challenge [14] of interoperability and data integration.

The potential of using ontologies for identity federation has already been recognized before [17]. However, these works rely on adopting a common ontology definition for person entity attributes. This approach is not directly applicable in our context scenario, as we consider different countries and hence different attribute definitions.

2.4. Ontology Alignment

Even when systems use ontologies for knowledge description, the number of parties involved in the identity federations (e.g. Stork, eIDAS[6]) is usually large. As a consequence, several different ontologies can be used to do this representation. Ontology alignment finds equivalences between entities semantically correlated in ontologies. These equivalences can promote interoperability through query answering, or data translation [18]. Ontology alignment is used to obtain a common knowledge representation among entities (e.g.: users / citizens attributes definitions on APs). When two ontologies are aligned, the entities involved can start to use a common vocabulary to communicate with each other. It is required to align these different knowledge representations when different ISs , e.g. from different MSs, communicate with each other to find a way to provide shared knowledge among them.

More formally expressed, an alignment A_0 comprises a set of correspondences between entities of a given pair of ontologies O_1 and O_2. Moreover, some other parameters [18] can extend the definition of alignment, namely:

1. an input alignment A, to be extended;
2. the matching parameters, such as weights, or thresholds; and

[6]https://www.eid.as/home/

3. external resources, such as common knowledge and domain-specific thesauri.

The resulting alignment A_0 can be of various cardinalities: one-to-one (1:1), one-to-many (1:m), many-to-one (n:1) or many-to-many (n:m).

A correspondence C between two ontologies O_1 and O_2 consists of a source concept $c_s \in O_1$, a target concept $c_s \in O_2$, a relation type, and an optional confidence level between 0 and 1, expressing the computed likelihood of the correspondence [19].

A correspondence [18] is a 4-tuple:

$$\langle id, e_1, e_2, r \rangle \tag{1}$$

where:

- id is an identifier for the given correspondence;
- e_1 and e_2 are entities, e.g. classes and properties of the first and the second ontology, respectively; and
- r is a relation, e.g., equivalence ($=$), more general ($\sqsupseteq$), disjointness ($\perp$), holding between e_1 and e_2.

Correspondences have some associated metadata, e.g.: confidence (on a [0.0, 1.0] scale), where 1 represents the maximum probability that the relation holds.

The Ontology Alignment Evaluation Initiative[7] (OAEI), which promotes annual evaluations of matching systems, proposes the usage of three metrics to assess the confidence level taken by these matching systems [18], namely:

- Precision: measures correctness;

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- Recall: measures the completeness;

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- F-Measure: aggregates both previous metrics.

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

Where:

- TP stands for True Positive values;
- FP are the False Positive values;
- FN stands for False Negative values;

[7] http://oaei.ontologymatching.org/

2.5. Privacy

Privacy [20] is a fundamental human right, laid down in the United Nations Universal Declaration of Human Rights, the European Convention on Human Rights, and national constitutions. Since it began, the main focus of privacy has been personal information, especially with regard to defending individuals from government surveillance.

Data protection is the administration of personal information and the European Union frequently uses this definition in elaborating privacy-related laws and regulations [20], [21].

Privacy terminology includes [20] terms such as data controller, data processor and data subject. Their meaning is as follows:

- Data Controller: An entity which determines the purposes for which and the way in which any item of personal information is processed.
- Data Processor: An entity which processes personal information on behalf of and upon the instructions of the Data Controller.
- Data Subject: An identified, or identifiable individual to whom the personal information is related directly or indirectly.

The Organisation for Economic Cooperation and Development (OECD) Privacy Framework defines some basic principles with regard to fair information practices [22], namely:

- Collection Limitation Principle: There should be limits to the acquisition of personal data. Personal data should be obtained according to the law and by fair means.
- Data Quality Principle: Personal data should be related to the purposes for which they are to be employed, should be correct and kept up-to-date.
- Purpose Specification Principle: The purposes for collecting personal data should be specified beforehand and the succeeding use limited to those purposes.
- Use Limitation Principle: Personal data should not be revealed, made accessible or employed for purposes other than those defined in accordance with Purpose Specification Principle except:
  1. with permission of the data subject; or
  2. by the law's authority.
- Security Safeguards Principle: Personal data should be protected by security safeguards against risks such as loss, unauthorized access, destruction, use, changes or disclosure of data.

- Openness Principle: There should be a comprehensive policy of openness about evolving, practices, and policies on personal data.
- Individual Participation Principle: Individuals should have the right:
    1. to obtain from a data controller a confirmation of whether or not it has data relating to them;
    2. to be communicated about data relating to them;
    3. to be given reasons if (a) and (b) are denied, and to be able to challenge such denial;
    4. to challenge data relating to them and, if successful, to have the data deleted, corrected, supplemented or improved.
- Accountability Principle: A data controller should be responsible for complying with rules which give effect to the principles declared above.

Transferring user data among several MS's identity systems implies being careful about how to send this data to each MS in order to not reveal it (Use Limitation Principle). User privacy and the minimum-disclosure rule must be respected (Security Safeguards Principle). Some solutions to address the Use Limitation Principle feature:

- Aggregated Zero-Knowledge Proofs of Knowledge (AgZKPK) [23];
- Oblivious Commitment Based Envelope (OCBE), [24];
- Locality-Sensitive Hashing (LSH) [9].

3. Related Work

Several solutions exist to achieve ontology alignment in practice (e.g. AlignAPI, PROMPT, and XMAP) and to perform queries in blinded text values (e.g. MinHash, Nilsimsa, and TLSH). These are key features in the proposed solution since they provide the building blocks to make the systems interoperable and do not disclose any information except that requested from the user.

3.1. Ontology-Alignment

Since understanding the concepts adopted by each party is crucial to aggregate the attributes they store, merely applying ontologies to model these concepts is not enough to promote data interoperability. A robust asset to align the different ontologies is as important as a good definition of the knowledge representation applied.

Two or more ontologies are aligned to enable interested entities to employ a common terminology to communicate with each other. The following subsections briefly describe three of the most commonly used ontology alignment solutions.

3.1.1. AlignAPI

The Alignment API (AlignAPI[8]) can be applied for the development, integration, and composition of matchers [25]. Its reference implementation aims to promote the development of tools for manipulating alignments and calling matchers [26].

3.1.2. PROMPT

PROMPT[9] is an algorithm and a tool for merging and aligning ontologies [27]. It demands direct interaction with the user. The tool takes two ontologies as input [28] and guides the user through the process of creating a merged/aligned ontology.

3.1.3. XMAP

The XMAP[10] is a high-precision ontology matching system that can perform matching on large ontologies [29]. It uses the UMLS[11] and WordNet[12] to compute a synonymy degree between two concepts in several ontologies, using their context. The XMAP relies on the Microsoft Translate API [13] to operate with ontologies in multiple languages.

3.2. Locality-Sensitive Hashing Functions

LSH Functions ensure that the collision probability is higher for closer objects (similar values) than for those that are far (different attribute values) [30, 31]. Locality-sensitive hashing functions perform a similarity query on an LSH index in two steps [32]:

1. Selecting candidate objects for a given query q using LSH functions; and
2. Ranking these objects according to their distances to q.

Performing similarity queries on blinded data would also be possible using homomorphic encryption, but LSH is less complex [33], which improves the performance of the matching verification process. In the following subsections, we succinctly describe existing implementations of LSH functions.

3.2.1. MinHash

MinHash evaluates the similarity of any two sets demanding only a constant number of comparisons [34]. MinHash performs the evaluation by extracting a representation $h_k(S)$ of a set S using deterministic sampling. This representation $h_k(S)$ has a constant size k, independent from $|S|$. The computation of $h_k(S)$ incurs a complexity linear in set sizes.

---

[8] http://alignapi.gforge.inria.fr/
[9] http://protegewiki.stanford.edu/wiki/PROMPT
[10] http://www.labged.net/index.php?rubrique=mapage38
[11] https://www.nlm.nih.gov/research/umls/
[12] https://wordnet.princeton.edu/
[13] https://www.microsoft.com/en-us/translator/translatorapi.aspx

3.2.2. Nilsimsa

Nilsimsa [35] is a Locality-Sensitive Hashing function that receives an arbitrary input and outputs an n-bit digest. It adopts n buckets to count the trigrams that appear in the input and converts the counts to an n-bit digest. The similarity evaluation between two inputs is conducted comparing the corresponding position of the two Nilsimsa digests and counting the number of corresponding bits. The algorithm counts the number of corresponding bits of the two Nilsimsa digests in the same position to recognize the similarity between two inputs [36]. The higher the number of corresponding bits, the more similar the two documents.

3.2.3. Trend Locality Sensitive Hashing - TLSH

This method computes a TLSH value from given input data. The TLSH value is obtained by summing up the distance[14] between the digest headers and the digest bodies. The resulting distance score ranges from 0 to 1000+. Digests with a $distance \leq 100$ are considered to be similar. Digests with a $distance > 100$ are considered not similar. The assessment of the TLSH digest of the byte string follows these steps [37, 38]:

1. Process the byte string using a sliding window to populate an array of bucket counts;
2. Calculate the quartile points, q_1, q_2, and q_3;
3. Define the digest header values as a function of:
    (a) the length of the file;
    (b) the quartile points calculated in step (2); and
    (c) a checksum.
4. Define the digest body by processing the bucket array;
5. Produce the output digest by concatenating the digest header from step (3) and the digest body from step (4).

3.3. Interoperability for Electronic Identities

Achieving interoperability between electronic-identity systems has been a topic of scientific interest for years. Large research projects such as STORK and STORK 2.0 have not only yielded a specification and implementation of an interoperability framework that ensures interoperability between European national identity systems, but have also produced various publications that address the topic from a scientific perspective [15, 39, 40].

In addition to these works, other authors have approached the topic of eID interoperability as well. In [41], the authors present a review on identity management frameworks. They assess existing solutions

[14]The assessment of the distance occurs in a process similar to the Hamming distance.

and emphasize the relevance of privacy and anonymity (e.g. to protect users against linkability), as well as location independence (e.g. allowing users to provide their attributes independent of the respective attribute provider's location). Our work addresses these relevant aspects by not storing any user attribute and by allowing the user to specify the preferred AP.

Another interesting work related to our proposal has been introduced by Esposito [42]. The focus of this work lies on interoperable, dynamic and privacy-preserving access control solutions for cloud data storage. The author proposes an ontological approach matching the different ontologies describing the diverse access-control models. The usage of pseudonyms avoids the exposition of the users' personal information, preserving their privacy. Our work follows a slightly different approach. It handles the diversity of attribute specifications with the help of ontologies and preserves the users' privacy by blinding attribute values using the Nilsimsa LSH function.

4. Proposed Solution

Surveying related work reveals that there is currently no satisfactory solution that enables privacy-preserving attribute aggregation in federated identity systems. In this section, we propose a solution to bridge this gap.

The key element of our proposed solution is a component called User Identification Strengthening [6] (UsIdS). This component becomes part of federated identity management systems as shown in Figure 1 and extends these systems with the following features:

- Ontology Mapping with privacy preservation;
- Language translation;
- History-based confidence level improvement; and
- Third Party AP chain construction.

Figure 1 shows the role of the UsIdS in more detail. The UsIdS extends the Identity Provider Proxy (IdPP) (e.g. STORK C-PEPS [5]) of each Identity System (IS) belonging to the identity federation. If an IS does not have an IdPP, the UsIdS can also run on the IS's AP. The UsIdS acts as a Data Processor (as described in 2.5) processing user attributes without storing them, and without requiring any other data from the users than that processed by the IdPP.

During a user-authentication process, an iterative process involving the user is executed. In this process, required attributes are aggregated and delivered to the SP. Figure 1 presents an overview of the UsIdS workflow. A typical authentication process comprises the following steps. The user accesses an SP using its IS Proxy (IS-IdPP) as IdP (Fig. 1, Step 1). The SP redirects to the user's IS-IdPP to obtain the user's

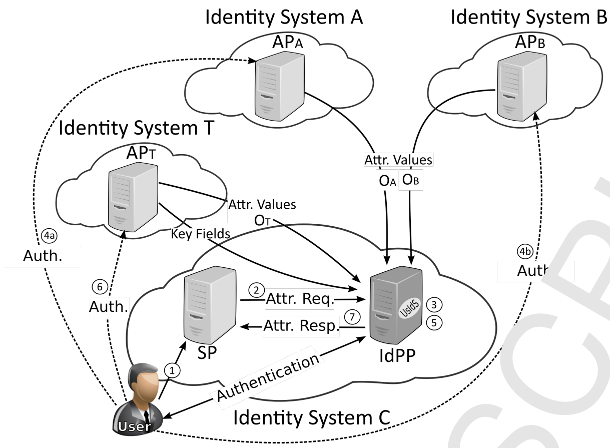

Figure 1: UsIdS Architecture

credentials (Fig. 1, Step 2). Then, the IS-IdPP performs a search for the requested credentials (*Req*) in the stored ontologies (e.g. O_A and O_B) it has for the requesting user (Fig. 1, Step 3). Next, the UsIdS sends a request to get the attributes in *Req*. The user authenticates at both APs (Fig. 1, Steps 4a and 4b) that reply with the requested attribute values in plain text (e.g. Pt_A or Pt_B) if they are in *Req* and/or blinded (e.g. LSH_A or LSH_B) otherwise.

If it is not possible to merge the sets of attributes obtained from different APs, even with the additional blinded attributes, the UsIds uses its ontologies database to search for Third Party AP (AP_T) candidates to establish an association between AP_A and AP_B (Fig. 1, Step 5). The user authenticates at that AP_T (Fig. 1, Step 6), which then returns to the UsIdS the attributes that link the attribute sets from AP_A and AP_B. After obtaining the attributes, and finding a way to merge them, the produced facet is provided to the SP (Fig. 1, Step 7). The communication process encompasses different messages. Figure 2 describes the details of the communication between the UsIdS and each AP, omitting the authentication messages for the sake of simplicity.

Our proposed solution comprises two distinct phases. The goal of the first phase is to find a common identifier between the participating attribute providers. This common identifier is nothing more than the user's set of attributes, shared by both attribute providers, which identify the user uniquely and may, consequently, be used to link both user facets.

The strategy to find such an identifier has two alternative paths. Each path is tried in sequence from the most simple to the most complex one until one succeeds.

1. **First Phase** - Find common key identifier(s):
   [Alternative Paths]

   (a) Between AP_A and AP_B;

(b) Between some Third-Party AP (AP_T) and each AP_A and AP_B.

The second phase is the one that satisfies the SP's request. It starts by requesting the actual attribute values. Then it checks if the attributes belong to the same user performing the requests on both attribute providers. It then evaluates an Aggregation Confidence Index (ACI) and returns the requested facet and its ACI[15]. This phase has three sequentially executed steps to provide the answer to the SP's request.

2. **Second Phase -** Satisfy request

   [Sequential Steps]

   (a) Request the attribute values from the APs;

   (b) Verify the user's unicity in all the APs and evaluate the ACIs of the requested attributes;

   (c) Return requested attribute values, and its ACIs;

If there are no common key identifiers or no attribute match, an error message is sent to the SP. Each phase is described in detail in the following subsections (Subsections 4.1 and 4.2).

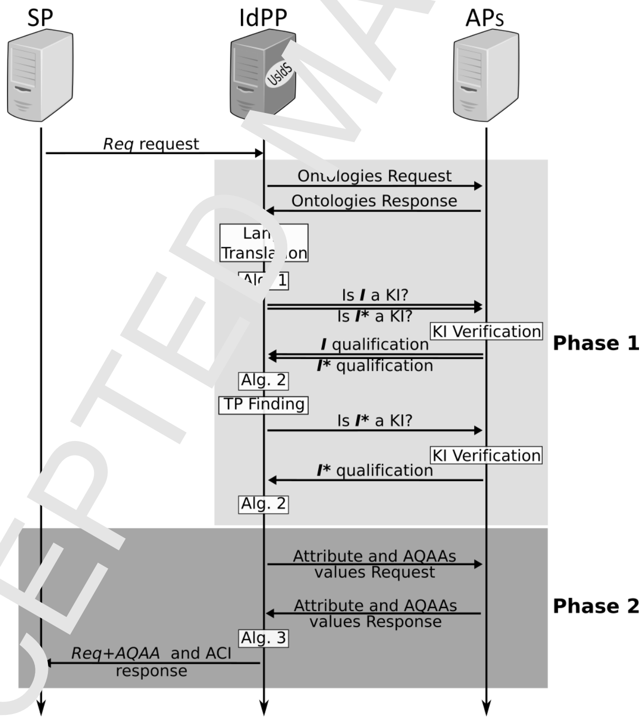

Figure 2: Time line communication evolution

[15] The ACI is a metric that indicates the confidence level of the aggregation process of each attribute pair, see Algorithm 3 for more details.

4.1. First Phase

The first phase of the protocol is the most complex. Moreover, it is the one where the process of aligning ontologies is necessary given that it is not expected that every AP will use the same ontology. The process of translating the language of the involved ontologies also takes place in this phase. The goal of this phase is finding common key identifiers between APs.

Both paths in the first phase share a common procedure (Algorithms 1 and 2), with small differences. The difference between the first path and second path is that the second path runs the same procedure twice, one between a third party AP_T and AP_A and another one between AP_T and AP_B. This yields two key identifiers that can be transitively coupled as though they were only one.

The UsIdS runs both steps with the help of the user that interacts with the APs providing, or not, their consent on the attribute exchange. In the following, the user's role in the communication process between the UsIdS and APs is omitted, but it is assumed that there is always an authenticated user performing the communication with the APs.

4.1.1. Multi Language Alignment

The UsIdS starts by requesting ontologies from the attribute provider and the service provider. It then verifies the language of the provided ontologies. If they do not have the idiom of the ontology in the UsIdS, the UsIdS proceeds by creating a new translated version of them.

Creating a new version helps in verifying when the original ontologies change by just comparing them (the stored ones) with fresh ones provided on each interaction with the attribute providers/service providers.

These new translated versions of the ontologies are used to perform the alignment with the UsIdS ontology and, eventually, with the other attribute providers involved in the process.

4.1.2. Ontology Aligning

The UsIdS proceeds by aligning the ontologies by one of the methods described in Section 2.4. The result is an Ontological Relation (OR), a Confidence Level (CL) for each aligned attribute pair, which is used for calculating the ACI for the complete set of attributes. It also returns $Comp_A$ and $Comp_B$ with those attributes that O_A and O_B do not share.

Sometimes the aligning process results in attributes paired with more than one attribute on the other ontology (e.g. $(O_A.Address, O_B.Title, 0.63)$ and $(O_A.Address, O_B.Address, 1.0)$, taking a threshold $t = 0.6$). This multiplicity of attribute pairing in ORs depends on the threshold level chosen in the alignment process. To address this multiplicity pairing issue, a procedure runs on all attribute pairs having a $CL < 1$ and deleting those having the same attribute in another association with $CL = 1$. The UsIdS keeps the OR to communicate with both APs using their terminologies.

Algorithm 1 Ontology Aligning Process

Data: The URI of the Ontologies from both APs (O_A, O_B).
A threshold level to each attribute pair aligned (t).
Result: The Ontological Relation (OR) between O_A and O_B;
The Confidence Level of each attribute pair aligned (CL);
The complement of O_A related to O_B ($Comp_A$); and
The complement of O_B related to O_A ($Comp_B$).
function ONTOLOGYALIGNING(O_A, O_B, t)
  $EN\_O_B = translate(O_B)$
  $OR = Align(O_A, EN\_O_B, t)$
  $Comp_A = O_A \setminus O_B$
  $Comp_B = O_B \setminus O_A$
  $OR = removeDuplicated(OR)$
  saveToDB(OR) ▷ A copy of O_A and O_B is stored on local file system.
  return $OR, CL, Comp_A, Comp_B$

4.1.3. Common Key Identifier

The resulting OR is used to request attributes from each AP_x (where x stands for A or B). The OR is a common subset of attributes in that AP_x, and each AP_x should verify if it is sufficient to identify the user uniquely. Note that the OR may not be sufficient to uniquely identify every user in AP_x, but it may be adequate to identify the requesting user uniquely at a specific time. For instance, taking the attributes *Given_Name* and *Nationality* may not be sufficient to identify a user in any student database. However, performing a search in a specific database with a specific name and nationality may be enough to return just one record.

Assuming that $AP(r, a)$ stands for the list of attribute values, where r stands for the requesting user, and a for the list of attribute names. And that OR_{AP} stands for a list of attribute names in the ontology relation OR for that specific AP. An AP classifies the OR as a Key Identifier (KI) for the requesting user r if there isn't another user in the AP with the same set of values for the attribute in the OR, formally:

$$\forall u \in AP : u \neq r \implies AP(u, OR_{AP}) \neq AP(r, OR_{AP}) \quad (5)$$

The algorithm takes as input the full ontological relation OR, from Algorithm 1, the user identification on that AP UID, and the attribute name to be checked $attrName$. Then the value of the attribute in the OR for that specific user is assigned to a local variable $attrValue$. $attrValue$ is then submitted in a query to verify the number of users with that attribute value for that attribute name. If the number of users is one, then that attribute is a key identifier for that user on that AP. It is important to notice that performing this verification in this way does not disclose any information, besides true or false, about that attribute set of the user. If the attribute set is a KI for both APs, then it is called a CKI, and phase 1 stops.

Algorithm 2 Key Identifier Verification

```
Data: An aligned ontology to test (OR), the User identifier (UID)
      on that AP, and the Attribute Name (attrName) to be
      checked;
Result: True if the attrName is a KI for the user.
function isKI(OR, UID, attrName)
    attrValue = getValues(OR, UID)
    users = getUsers(OR, attrName, attrValue)
    if (users == 1) then
        return True
    else
        return False
```

4.1.4. Third Party AP

When it is not possible to identify the user uniquely, a third-party AP (AP_T) strategy is applied. This AP_T is no more than an element acting as a link between two other APs. The process starts by finding an AP_T for which there is a OR_{TA} and a OR_{TB} classified as CKI for the same user.

The strategy for finding the AP_T with the necessary characteristics is by sequentially testing every AP known by the UsIdS. For larger lists, the search can be sped up by using heuristics like the number of times that the OR between two APs was classified as CKI[16], or the length of the OR (longer ORs have more probability of being a CKI than others).

The protocol to check that either OR_{TA} or OR_{TB} is a CKI is similar to the one described earlier. They differ because AP_T must verify that the ORs are checked for the same user, which is easier if they are checked at the same time, in the same request.

4.2. Second Phase

The second phase of the protocol retrieves the attribute values from both APs, checks that the shared attributes have the same value in those APs, and return the result to the SP. In this phase, the algorithm run by the UsIdS faces two challenges. The first challenge is to compare attributes from different ontologies that are not directly comparable. The second challenge is to match attributes without knowing some of their values. In fact, some attributes are used only to match the facets from different attribute providers. Since those attributes are not in *Req*, the user should not be asked to reveal them.

The strategy to handle the first challenge is to use similarity functions $\mathcal{S}$, which combine Hamming Distances with other heuristics to calculate a distance between the values in both APs. The strategy to handle the second challenge is to use Locality Sensitive Hashing (LSH) functions, like Nilsimsa [43], which allows comparison with the signatures using the same kind of similarity functions that are used to match clear values.

[16] An OR can be classified as CKI for one user and fail to be classified as a CKI for another user.

The Nilsima function was chosen for the current prototype because it presented the best results in a series of performance tests conducted in [9]. The Nisilma function is transparently applied by each AP whenever the UsIdS requests an attribute value that is not in *Req*. The Algorithm 3 uses these two strategies. It returns a composed facet with all the attributes requested by the SP, or an error if the attributes returned by both APs do not match.

Algorithm 3 Get Unified User Facet

Data: The merged ontology (Ontological Relation) from both
  APs (OR);
  The ACI value obtained on previous interactions;
  The attribute set requested by the SP (Req);
  Distance threshold between values (d);
  The ontologies from AP_A (O_A) and AP_B (O_B);
  The complements $Comp_A$ and $Comp_B$.
Result: An array, $Facet$, which provides the confidence level of
  matches of each attribute in Req, and the ACI.
function FINDMATCHES(OR, Req, d, O_A, O_B)
  $SharAttr = (OR \cap O_A \cap O_B)$
  for ($attrName \in SharAttr$) **do**
    $Val_A = AP_A(UserID, attrName, Req)$
    $Val_B = AP_B(UserID, attrName, Req)$
    $cl = \mathcal{S}(Val_A, Val_B)$
    if ($cl == 0$) **then** OR.ORCI++
    if ($cl \leq d$) **and** ($attrName \in Req$) **then**
      $Facet+ = (attrName, Val_A, cl)$
    else
      return error
  for ($attrName \in Comp_A$) **do**
    if ($attrName \in Req$) **then**
      $Val_A = AP_A(UserID, attrName, Req)$
      $Facet+ = (attrName, Val_A, 0)$
  for ($attrName \in Comp_B$) **do**
    if ($attrName \in Req$) **then**
      $Val_B = AP_B(UserID, attrName, Req)$
      $Facet+ = (attrName, Val_B, 0)$
  $ACI = ACI/\mathbf{len}(Facet) * OR.ORCI$
  return $Facet$, ACI

The Get Unified User Facet Algorithm (Alg. 3) receives as parameters the OR, the previous value of the ACI for that parties, the requested attributes (Req), a distance threshold d, the ontologies from AP_A and AP_B (O_A and O_B, respectively), and the complements $Comp_A$ and $Comp_B$, from O_A and O_B respectively. The shared attributes ($SharAttr$) is the result of the intersection among OR, O_A, and O_B. A test in all attributes in $SharAttr$ is performed to verify if they belong to Req. If the attribute belongs to Req, the clear text value is requested to its AP and assigned to Val_A or Val_B, respectively. Otherwise, its LSH value is obtained from the set ($O_A \setminus Req$), or ($O_B \setminus Req$), and also assigned to Val_A or Val_B, respectively.

The similarity function $\mathcal{S}$ is applied to both values. If the returned distance is less than the threshold d and if the attribute is in Req, then the Facet receives that new attribute value and the Confidence Level (CL) on that alignment. The Facet also receives the attribute values for those attributes that belong to $Comp_A$, or $Comp_B$, but with the value 0 (zero) in the Confidence Level (CL) since it does not represent an alignment.

The Aggregation Confidence Index (ACI) is updated taking into account the CL between the attribute values of both APs and the ACI obtained in the previous alignments with the same APs (History Function - Subsection 6.2). At the end of the process, the ACI is normalized, on a scale from 0.0 to 1.0, and returned together with the composed Facet.

4.2.1. Alignment Confidence Improvement

Ontology alignment is a threshold matching process that can be tuned to provide almost no false positives. However, being too restrictive results in many false negatives, especially because attribute names are often very small words or sequences of words. A high number of false negatives defeats the alignment's purpose and prevents users from finding CKI between APs.

In the specified context, it is common that the same APs are used in several aggregation procedures. These recurrent interactions between/among the APs can provide valuable information to the previously established ontology alignments. We assume that previous alignments can be improved on subsequent alignments (cf. 4.3.5). For instance, if two attributes names from different ontologies are wrongly paired, most comparisons of user values of those attributes will be false, providing a hint about the attributes misalignment.

Every alignment generated between AP Ontologies is stored in a database for future use and improvement. Alignments do not have private data, although their confidence levels are calculated using the attribute values of previous facet aggregations. Ontology alignment confidence levels are updated on every facet merging request using the alignment, as a sub-product of the facet merging confidence level.

4.3. Facet Merging Confidence Level

The ultimate goal of the UsIdS is to provide the SP with a single set of attributes comprising two facets of the same user, together with a confidence level on the correctness of that merger. This confidence level must be very high for the process to be useful. Unfortunately, calculating it is not trivial. There are a number of variables in the calculation, for which it is only possible to provide a rough approximation. However, many of them have a relatively low impact on the calculation and may be underestimated without much loss of precision.

In this Section we provide a lower bound estimation of the confidence level for each facet merger.

Let M_n be the random variable representing the n^{th} facet merger involving two APs. Then, $P(M_n)$ denotes the probability of both facets belonging to the same user. Recall that the confidence level of the ontology alignment evolves with the number of previous facet mergers using the same APs.

Both facets are aligned in pairs $A_j = (A_j^0, A_j^1)$, representing the j^{th} attribute from Ontology 0 and Ontology 1, respectively, and the semantic accuracy of that alignment is denoted by $P^n(A_j)$.

The process of verifying that two facets belong to the same user involves comparing the attribute values of each aligned attribute pair. The attribute value pairs to compare, at each iteration n, are denoted by $V_{nj} = (V_{nj}^0, V_{nj}^1)$, and the probability of both values being the same is denoted by $P(V_{nj})$.

4.3.1. Probability Distribution Function of M_n

According to the Baye's Theorem

$$PosteriorOdds = Likelihood \times PriorOdds$$

where the *Odds* are the number of times that it is more probable that a hypothesis occurs against its opposite, before (*PriorOdds*) or after (*PosteriorOdds*) for a given event are detected. The *Likelihood* represents the number of times that a given event is more probable if the hypothesis is evaluated as true against its evaluation as false.

Let $\mathcal{O}(M_n)$ be the Posterior Odds of M_n, $O(M_n)$ the Prior Odds, and $L(V_{nj}, M_n)$ the Likelihood of V_{nj} under the hypothesis M_n, then the probability distribution function $P(M_n)$ is given by

$$P(M_n) = \frac{\mathcal{O}(M_n)}{\mathcal{O}(M_n)+1} = 1 - \frac{1}{\mathcal{O}(M_n)+1} \tag{6}$$

where

$$\mathcal{O}(M_n) = O(M_n)\prod_j L(V_{nj}, M_n) \tag{7}$$

Assuming that the probability, beforehand, of two given facets belonging to the same user is 50%, i.e. before any validation, the facets are equally probable to be from the same user or different users, then $O(M_n)$ is equal to 1. This is a conservative assumption because if the user was able to authenticate in both APs then it is more probable that it is the same user than two different users.

4.3.2. Attribute Likelihood Assessment

The likelihood that an attribute pair is equal is given by:

- The probability of the attribute pair is equal given that the hypothesis M_n is true $P(V_{nj}|M_n)$; over
- The probability that they are equal given that the hypothesis M_n is false $P(V_{nj}|\overline{M_n})$.

$$L(V,M_n) = \frac{P(V_{nj}|M_n)}{P(V_{nj}|\overline{M_n})} \tag{8}$$

where $P(V_{nj}|M_n)$ is provided by the Nilsimsa Distance algorithm over the value pair $V_{nj} = (V^0_{nj}, V^1_{nj})$, and $P(V_{nj}|\overline{M_n})$ denotes the probability of a false positive, that may happen due to two main reasons:

1. The attribute values are the same but they are from two semantically distinct attributes, which were not aligned correctly (e.g. Given Name from one user and Last Name from another). The probability of such an event is denoted by $P(V_{nj} \cap \overline{A_l}|\overline{M_n})$.

2. Two users share the same attribute (e.g. two users with the same Given Name), which is denoted by the probability $P(V_{nj} \cap A_i|\overline{M_n})$.

Thus

$$\begin{aligned} P(V_{nj}|\overline{M_n}) =& P(V_{nj} \cap \overline{A_l}|\overline{M_n}) + P(V_{nj} \cap A_i|\overline{M_n}) \\ =& P(V_{nj}|\overline{M_n} \cap \overline{A_l})(1 - P^{n-1}(A_j)) + \\ & P(V_{nj}|\overline{M_n} \cap A_i)P^{n-1}(A_J) \end{aligned} \tag{9}$$

where

- $P(V_{nj}|\overline{M_n} \cap \overline{A_i})$ denotes the probability of a false positive if the alignment is incorrect;
- $P(V_{nj}|\overline{M_n} \cap A_i)$ denotes the probability of a false positive if the alignment on that attribute is correct;
- $P^{n-1}(A_j)$ denotes the trust on the alignment after the previous assessment.

4.3.3. False Positives with Correct Alignment

The probability $P(V_{nj}|\overline{M_n} \cap A_i)$ is directly proportional to the number of repetitions of each attribute value. For instance, if there are many users with the same given name, the probability of this false positive is high.

Let $|A^i_j|$ denote the average number of users with the same attribute value A_j in AP_i, and $|AP_i|$ denote the total number of users in the repository AP_i, with $i = 0, 1$; then:

$$P(V_{nj}|\overline{M_n} \cap A_i) \approx \frac{|A^0_j|.|A^1_j| - 1}{|AP^0|.|AP^1|} \tag{10}$$

4.3.4. False Positives with Wrong Alignment

The probability $P(V_{nj}|\overline{M_n} \cap \overline{A_j})$ increases with the frequency of attribute values V_{nj}. The actual number of equal attribute values depends on many factors (the type of value, the universe of values in the AP, etc.) although it is expectable that smaller words or sentences are more prone to be repeated than longer words/sentences [44], which may provide an estimate on this probability.

Let $|V_{nj}|$ denote the length (number of characters) of the values of V_{nj}, and $\mathcal{F}(d)$ the frequency of the words/sentences with dimension d, that exists in the ontology's language, then assuming that the number of users in the APs is big, the probability may be approximated by:

$$P(V_{nj}|\overline{M_n} \cap \overline{A_i}) \approx \mathcal{F}(|V_{nj}|)\mathcal{F}(|V_{nj}|) \tag{11}$$

and, according to [44] the frequency of the words by length d, given by:

$$\mathcal{F}(d) \approx 11,74 * d^3 * 0,4^d \tag{12}$$

4.3.5. Trust on the Alignment

The alignment trust $P^{n-1}(A_j)$ changes as the number of assessments n grows. The improvement on the alignment confidence level can be seen as a sub-product of the facet merger confidence level calculation. As depicted in Equation 9 the confidence level of the facet merger depends on the alignment confidence level P^{n-1}(A_j), but the alignment confidence level also depends on the confidence level of the previous facet mergers. According to Bayes's theorem

$$P^{n-1}(A_j) = 1 - \frac{1}{\mathcal{O}(A_j) + 1} \tag{13}$$

where

$$\mathcal{O}(A_j) = O(A_j)\prod_{m}^{n-1} L(V_{mj}, A_j) \tag{14}$$

and

$$O(A_j) = \frac{P^0(A_j)}{1 - P^0(A_j)} \tag{15}$$

where $P^0(A_j)$ denotes the initial alignment probability of attribute j, after applying the Ontology alignment algorithm (e.g. AlignAPI), and $L(V_{mj}, A_j)$ denotes the likelihood of the attribute value pair being equal under the hypothesis that the alignment is correct.

4.3.6. $L(V_{mj}, A_j)$ Assessment

The Likelihood $L(V_{mj}, A_j)$ is given by:

- The probability that the attribute value pair V_{mj} is equal, if the alignment on that attribute A_j is correct ($P(V_{mj}|A_j)$), over
- The probability that the attribute value pair V_{mj} is equal, if the alignment on that attribute A_j is incorrect ($P(V_{mj}|\overline{A_j})$).

$$L(V_{mj}, A_j) = \frac{P(V_{mj}|A_j)}{P(V_{mj}|\overline{A_j})} \tag{16}$$

As before, $P(V_{mj}|A_j)$ is set by the Nilsima Distance algorithm over the pair V_{mj}, while $P(V_{mj}|\overline{A_j})$ denotes the probability of a true or false positive with an incorrect alignment:

1. The false positive is denoted by $P(V_{mj} \cap \overline{M_m}|\overline{A_j})$, occurs when two users have the same value in two different semantic attributes (e.g. the given name of one is equal to the surname of the other);

2. The true positive, denoted by $P(V_{mj} \cap M_m|\overline{A_j})$, occurs when one user shares the same value in different attributes (e.g. one user with same given name and last name).

Therefore:

$$
\begin{aligned}
P(V_{mj}|\overline{A_j}) =& P(V_{mj} \cap \overline{M_m}|\overline{A_j}) + P(V_{mj} \cap M_m|\overline{A_j}) \\
=& P(V_{mj}|\overline{M_m} \cap \overline{A_l})(1 - P(M_m))+ \\
& P(V_{mj}|\overline{A_j} \cap M_m)P(M_m)
\end{aligned}
\tag{17}
$$

where

- $P(V_{mj}|\overline{M_m} \cap \overline{A_l})$ is given by eq. 11;
- $P(V_{mj}|\overline{A_j} \cap M_m)$ denotes the probability that a single user has two attributes with the same value (e.g. same surname and given name).

using the same strategy of equation 11, $P(V_{mj}|\overline{A_j} \cap M_m)$ may be majorated by

$$
P(V_{mj}|\overline{A_j} \cap M_m) \leq \mathcal{F}(|V_{nj}|)\mathcal{F}(|V_{nj}|) - \frac{1}{((|A_j^0| - 1)(|A_j^1| - 1)} \tag{18}
$$

where $|A_j^0|$ and $|A_j^1|$ are the number of equal values of attribute j in AP^0 and AP^1. These values depend on the size of the APs and on the type of the attribute. A passport number does not repeat, but a given name or a family name can be very common. The simplest way to estimate those values is classifying the attributes into categories and assign a frequency value to each category. Passport numbers are unique, then $|A_j^0| = 1$. Short names may repeat, at most, 10% in the entire database, then $|A_j^0| \leq 0.1|AP^0|$, while long names are less prone to repetition $|A_j^0| \leq 0.01|AP^0|$.

5. Prototype Implementation

To evaluate the practicability of the proposed solution, we developed a proof-of-concept prototype implementation. The implementation comprises a Service Provider and an Attribute Provider as illustrated in Figure 3 using RESTful Web Services written in Java with the JAX-RS RESTFul API[17]. We also implemented the test ontologies representing user attributes used in our experiments. Our implementation focuses on the Service Provider and Attribute Provider intentionally. The implementation of the intermediate gateways is regarded trivial. Respective solutions are already available, e.g. STORK [40], and eIDAS.

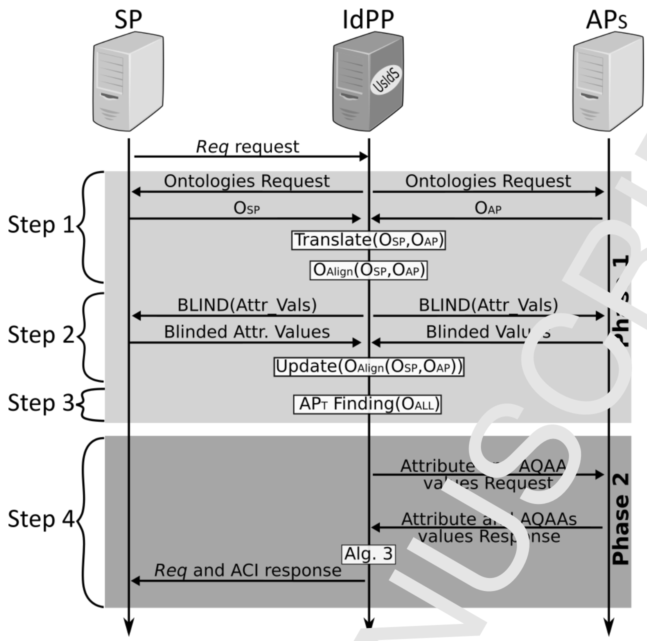

Figure 3: Communication overview.

For simplicity, the SP plays both the SP and one of the APs roles, which may in fact be a real valid scenario, as it is not uncommon for SPs to store data about their users.

The multi-language feature (Subsection 4.1.1) has been developed using the Yandex API[18], which provides the services needed to identify the language and to translate the attribute names before the alignment process begins (Fig. 3, Step 1). Next, the AlignAPI [26] performs the initial ontology alignment (Fig. 3, Step 1), and the Nilsimsa LSH function [36] carries out the blinding procedure of the attribute values (Fig. 3, Step 2). Additionally, the prototype implements a confidence level improvement algorithm (Fig. 3, Step 2 - see Subsection 4.2.1) and the third party AP strategy as described in Subsection 4.1.4 (Fig. 3, Step 3).

Figure 3 depicts the four communication steps over the two phases described in Section 4, which are detailed below.

5.1. First Phase

The First Phase, as described previously, performs the alignment between the ontologies of both involved parties (i.e. *AP* and *SP*). The approach adopted was to put the threshold of the ontology alignment step so that no false negatives occur. False positives are trimmed by extending the facets to compare with as many blinded attribute values as possible, and increasing the alignment confidence level continuously (cf. section 4.2.1).

[17]https://jersey.java.net/index.html
[18]https://tech.yandex.com/translate/

5.1.1. Multi-Language Alignment

The ontologies requested, from the SP and the AP, are evaluated using their attributes to identify their languages. If the language is not the same as that employed by UsIdS, the UsIdS translates the ontology to its own language. The UsIdS then saves a local copy of these new versions of the ontologies. The Algorithm 4 illustrates the language verification procedure.

Algorithm 4 Language Verification

```
Data: An ontology O_Lang from AP (O_AP) or SP (O_SP)
      to be checked;
Result: True if the ontology is in the UsIdS's language, or false
        and the translated version of the ontology otherwise.
function isSameLanguage(O_Lang)
    ontoLang = getLanguageFromOntology(O_Lang)
    if (ontoLang == langUsIdS) then
        return True
    else
        O_Translated = getTranslVer(O_Lang, langUsIdS)
        saveTranslVer(O_Translated)
        return False
```

5.1.2. Ontology Alignment

In the Ontology Alignment step (Fig. 3, Step 1), the service provider submits its ontology (O_{SP}) to UsIdS. The UsIdS employs the AlignAPI algorithm to align O_{SP} with the attribute provider's ontology O_{AP} generating $O_{ALIGN} = O_{AP} \cap O_{SP}$.

A Resource Description Framework (RDF) file results from this alignment. O_{ALIGN} comprises all attribute name relations and their respective confidence levels (CLs) assessed. These CLs are taken observing a threshold provided during the ontology alignment process. The threshold used is as low as the lowest confidence level on the true positive alignments identified. This strategy helps in eliminating the false negatives and allows the alignment process to be refined using the blinded attribute values approach (see Subsection 5.1.3).

Every time an interaction occurs with the UsIdS, the interacting party (e.g. AP) transmits its ontology to it. Sending its ontology is an important behaviour because the UsIdS checks for changes in the provided ontology using the one it has stored for that party. If both ontologies are the same, the UsIdS jumps to the Alignment Improvement execution discarding the remaining steps.

5.1.3. Alignment Improvement

In the Alignment Improvement (Fig. 3, Step 2), the UsIdS attempts to improve the confidence levels of the attribute-name pairs obtained previously (i.e. Ontology Alignment). The UsIdS requests from both AP and SP the blinded attribute values of the specified user for every attribute in O_{ALIGN} with confidence levels (CLs) lower than 100%. Then, the UsIdS evaluates the similarity of the blinded values received from the SP and the AP by applying the Nilsimsa Distance (ND) and updates the respective confidence levels (CL) for each attribute alignment (cf. Section 4.3.5).

5.1.4. AP_T Finding

If the provided attributes are not sufficient to establish a Common Key Identifier (cf. Section 4.1.3), i.e. it is not possible to ensure that it is the same user, the UsIdS starts a search on the database of ontological relations (ORs) for a third party attribute provider (AP_T).

To perform this search procedure, the UsIdS looks for all ORs, in which the user has attributes (i.e. AP_T candidates, AP_{TCand}). Then it proceeds by checking their ontologies trying to find a common key identifier between its ontology and the AP_{TCand}. If the UsIdS finds a key identifier in an attribute provider (AP_{TCand}) for that user and a key identifier is found with two other attribute providers, then that AP_{TCand} acts as an AP_T between the other two APs.

5.2. Second Phase

The second phase handles the actual exchange of attributes. The conducted ontology alignment process enables SP and APs to exchange attributes on their own terminology. The ontology alignment produced is employed to map the attributes needed by the service provider using O_{SP}, to the vocabulary used by the attribute provider, i.e.: O_{AP} through the UsIdS. This way, the attribute provider can perform a query on its database using its terminology. The attribute provider uses the attributes requested by the service provider to parameterize a query, which the attribute provider executes on its database. Finally, the attribute provider sends back to the service provider the set of attribute names and values requested (Fig. 3, Step 4), together with the confidence level of the facet aggregation.

6. Evaluation

We developed different test scenarios to evaluate our proposed solution. The first scenario uses two ontologies, one in English (O_E) and one in German (O_G), and checks the alignment between them considering the diversity of idioms (cf. Section 4.1.1). The second scenario uses two ontologies to evaluate the performance of the confidence level improvement algorithm (Subsection 4.2.1). Finally, the third scenario verifies how our solution handles the AP_T (cf. Section 4.1.4) strategy proposed. In the following subsections, we describe each one of these scenarios.

The SP and the AP databases were populated with 1000 users randomly generated by a random data generator[19]. Then an intersection of 26 users on both databases was artificially adjusted to allow test

[19]https://www.mockaroo.com/

execution. These 26 common users are the ones that have their attribute values changed during the test cases execution.

6.1. Multi Language Alignment

The purpose of this test scenario is to verify how the proposed solution performs on the language feature. An ontology in German (O_G) was written by a native speaker to evaluate how it would perform. This ontology contains the same ten attribute names as the ontology in English (O_E).

The original O_G was provided to our solution and aligned with O_E. After this first alignment, our multi-language feature translated O_G into a new version of the ontology but in English (i.e. EN_O_G), and only then it was aligned with O_E to assess the resulting alignment.

6.2. Confidence Level Improvement

This test scenario aims to evaluate the accuracy of the alignment process improvement procedure. We designed ontologies O_1 and O_2, with small, but logical, differences in attribute names, and assigned one to the SP and the other one to the AP. The AP and SP databases share 26 common users, which were used to conduct test runs with 26 rounds of positive matches, and measure the confidence level improvement over time in each run.

The test runs were conducted over four different types of test attribute values, dubbed TC1 to TC4, which account for different similarity scenarios among databases. In TC1, the Best Case Scenario, the CLs are 1 and never get worse. In TC2, the virtually Real World Scenario as in [5], the CLs are between 0.668 and 1. In TC3 , a Bad Scenario, the CLs are between 0.4 and 0.7, and in TC4, the Worst Scenario, the CLs are between 0.0 and 0.4. The last two test samples where generated using artificial similarity ranges between 40% and 70% ($TC3$) and 0% and 30% ($TC4$), and mimic a situation were the alignment is untrustworthy and should not be used at all.

Also in these last two test cases, $TC3$ and $TC4$, the experiments were executed using ten different user order sequences in its execution. Then the averages obtained from each result were evaluated.

Table 2: Test Cases Samples: Attribute changes sample for Ontology1. Adapted from [5].

Attribute	TC_1	TC_2
First Name	Joseph	Joseph
Surname	Boyd Junior	Boyd Jr.
Birthday	1953-03-15	15.03.1953
Profession	Associate Professor	Professor
...	...	...

6.2.1. *Third Party AP - AP_T*

The AP_T test scenario aims to verify the feasibility of the proposed Third Party AP feature, as well as its accuracy in establishing links with the involved APs. It means that the UsIdS must be able to establish a link between two attribute providers using a third one (AP) to establish that connection.

To achieve this goal, three ontologies (O_1, O_4, and O_5) were developed (see Table 3) and assigned to AP_A (O_1), AP_T (O_4), and AP_B (O_5). The service provider makes a request containing attributes from AP_A and AP_B, but those APs do not have a common key identifier. To satisfy the request, an AP_T should establish a link with AP_A and with AP_B so that those links allow the users identity to be ensured and answering the service provider with the attributes requested.

Table 3: Attribute names of the involved parties in the AP_T approach.

AP_A (O_1)	AP_T (O_4)	AP_B (O_5)
Address	Address	Birthday
BloodType	Anniversary	Blood
DateOfBirth	Blood	**E-mail**
Email	**Email**	GivenName
Nationality	FamilyName	Occupation
FamilyName	Gender	**PassportNumber**
GivenName	**Identification**	PhysicalAddress
Identification	Name	SSN
MaritalStatus	Occupation	Sex
Occupation	**PassportNumber**	Surname
Phone	**Telephone**	**Telephone**
Sex		
Title		

Notice that it is possible to establish a link between AP_A (O_1) and AP_B (O_5) by transitivity ensuring that is the same user using:

$O_1.Identification = O_4.Identification$ and

$O_4.Passport\_Number = O_5.Passport\_Number$.

6.3. *Results*

Our test scenarios provided results that support our goals. This subsection presents these results.

6.3.1. *Multi Language Alignment*

To perform both alignments with, and without, our translation feature, a threshold of 40% was defined. This value represents the worst threshold value when using the translation so that all ten attributes translated (EN_O_G) have correct alignments identified with O_E (see Subsection 5.1.3).

Table 4: Alignment between SP_B (German) and AP_A (English) without any translation.

SP_B (O_{DE})	AP_A (O_{EN})	Step 1	Step 2	Conf.
Wohnadresse	Address	55.56%	100.0%	99.9999999999975%
E-Mail	Email	100.0%	100.0%	100.0%
Familienname	FamilyName	45.45%	100.0%	97.5927343752426%
Vorname	GivenName	50.0%	100.0%	99.7250833346734%
Identifikator	Identification	59.26%	100.0%	99.9975712777413%

Table 4 shows the alignments identified without any translation. As can be observed, Step 1 was able to identify five attribute alignments with confidence levels from 45.45% (Familienname ↔ FamilyName) to 100% (E-Mail ↔ Email).

The resulting EN_O_G has the following ten attribute names(English):

- Beruf ↔ Profession
- Geburtstag ↔ Birthday
- Geschlecht ↔ Sex
- Familienname ↔ Family name
- Telefonnummer ↔ Phone number
- Vorname ↔ First name
- Blutgruppe ↔ Blood group
- Identifikator ↔ Identifier
- Reisepassnummer ↔ Passport number
- Wohnadresse ↔ Residential address
- E-Mail ↔ E-Mail

The alignment of this translated version of O_G (i.e. EN_O_G) with O_E, having a threshold of 40%, resulted in all of the ten attribute names aligned with their corresponding attributes in O_E, as can be observed in Table 5.

Using the translation feature allowed the alignment of the ten attributes of both ontologies. It represents an increment of 100% compared with the approach without any translation, which achieved five attribute alignments.

It is important to notice that the solution implemented applies an API[20] that currently supports more than 90 different languages.

[20] https://tech.yandex.com/translate/doc/dg/concepts/api-overview-docpage/

Table 5: Attribute names translated to the alignment between AP_A (English) and SP_B (German).

SP_B (O_{DE})	AP_A (O_{EN})	Step 1	Step 2	Conf.
Wohnadresse	Address	56.00%	100.00%	99.9999999999975%
Blutgruppe	BloodType	52.63%	100.00%	99.9999999999974%
Geburtstag	DateOfBirth	52.63%	100.00%	99.9906954154319%
E-Mail	Email	100.00%	100.00%	100.000000000000%
Familienname	FamilyName	100.00%	100.00%	100.000000000000%
Vorname	GivenName	44.44%	100.00%	99.7074362696315%
Identifikator	Identification	66.67%	100.00%	99.9978046835530%
Beruf	Occupation	41.38%	100.00%	99.9999999976026%
Telefonnummer	Phone	52.63%	100.00%	99.9999973807486%
Geschlecht	Sex	100.00%	100.00%	100.000000000000%

6.3.2. Confidence Level Improvement

The results obtained show the effectiveness of the CL improvement algorithm, by improving the CL to almost 100% for each attribute in Test Case 1 (TC1) and Test Case 2 (TC2) and eliminating the alignment altogether in TC3 and TC4.

Figure 4 shows the results for the best case TC1. This TC represents those interactions where the attribute values are almost exactly the same for both APs. The Figure depicts the Error Rate ϵ ($\epsilon = 1 - CL$) evolution over each of the 26 rounds for three attributes: $DateOfBirth$ (mean: 0.976, std. dev.: 0.118), $FamilyName$ (mean: 0.973, std. dev.: 0.126), and $Identifier$ (mean: 0.980, std. dev.: 0.102). We choose these attributes because they have the first step alignment value smaller than 100%: 52.63%, 47.06%, and 64.00% respectively. Notice that after 26 rounds the error rate is lower than 10^{-60}, thus achieving the goal of a very high confidence level.

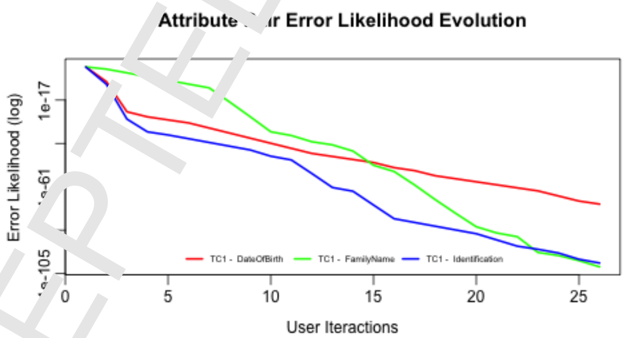

Figure 4: Error rate ϵ ($\epsilon = 1 - CL$) evolution with each user round, in the best case scenario, TC1.

Figure 5 shows the results for the worst TC in our assessments, TC4. The goal of this TC is to evaluate the performance of the CL improvement algorithm when either the initial alignment is incorrect or the two facets do not belong to the same user. Assuming that it is not probable that the initial alignment is completely incorrect the results of this test case depict the case of two users trying to mimic as one. The metrics obtained on this TC were: $DateOfBirth$ (mean: 0.027, std. dev.: 0.080), $FamilyName$ (mean: 0.024, std. dev.: 0.076), and $Identifier$ (mean: 0.036, std. dev.: 0.107).

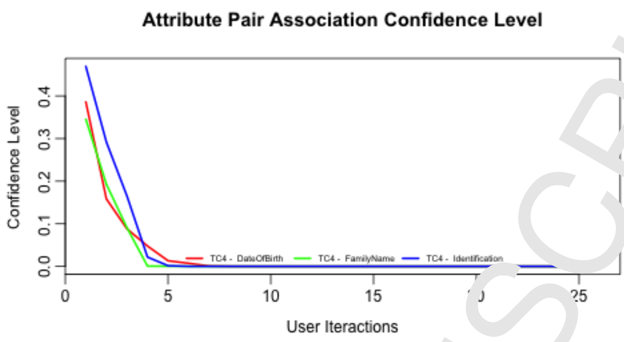

Figure 5: Confidence level evolution for 3 attributes over the 26 user evaluation rounds, in the worst TC, TC4.

Note that after a few interactions, the CL of the alignment tends towards zero, which is the overall goal of the improvement algorithm, either it reinforces the confidence level or demotes it completely. Figure 6 depicts this effect clearly by showing the evolution of the confidence level on all TCs for the $DateOfBirth$ attribute.

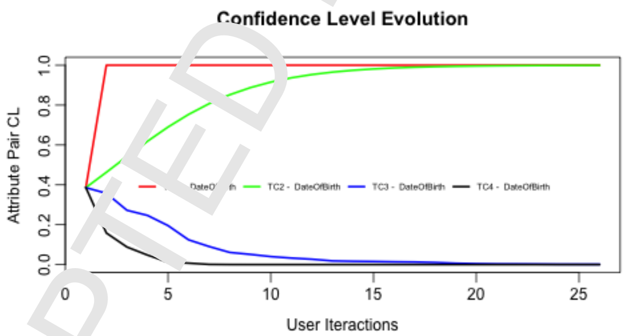

Figure 6: Confidence level evolution with user test rounds for each TC.

The overall goal of the improvement process is to improve the confidence level that the attribute set delivered to the SP being all from the same user. For test cases TC1 and TC2 the error rate $\epsilon = 1 - CL$ is very low ($\approx 10^{-25}$ for TC1 and 0.08 for TC2)(Figure 7).

As expected, the overall confidence level results for test cases TC3 and TC4, the confidence level of the resulted attribute set, are very low, telling the SP that it should not accept them, as they are probably from different users. On the other hand, the values for the certainty that the user who is trying to provide the

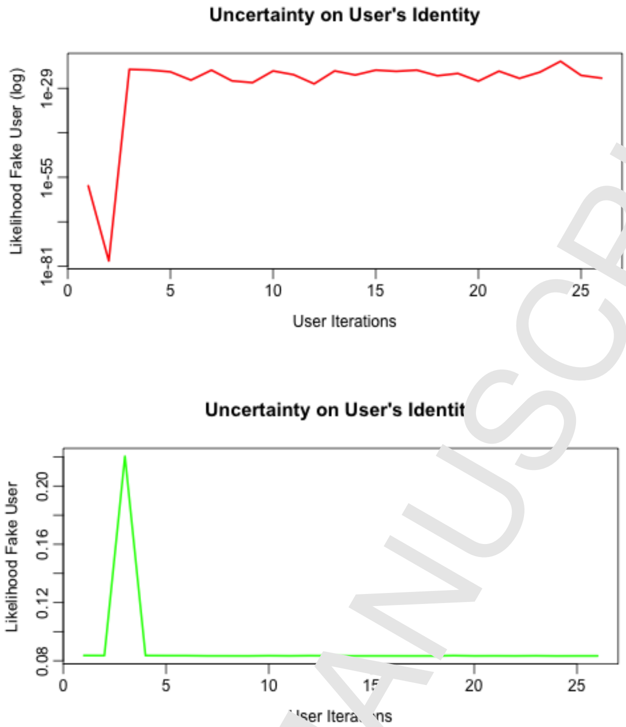

Figure 7: Overall error rate evolution with user test rounds for the best test case TC1 (top) and the real life test case TC2 (bottom).

credentials is a legitimate user drops on TC3 and TC4. If on TC1 and TC2 the likelihood of a fake user was small, on TC3 and TC4 the likelihood of an authentic user presents low values. Especially for TC4 that presents the smallest similarity values of all the TCs. Figure 8 depicts confidence level evolution with user-test rounds for TC3 and TC4. Notice that there is no clear tendency in both cases, but the confidence level values are always very low after the first user-test round.

6.3.3. Third Party AP

Our implementation was able to identify, in the already established ontological relations, APs that could act as an AP_T. It also provides the attributes it used as a key identifiers in this process.

In our experiment, as described in Subsection 6.2.1, our prototype identified the attributes *E-mail, Passport Number* and *Telephone* as Key Identifiers to establishing a link among AP_A, AP_B, and AP_T.

6.3.4. Privacy

Finally, executed tests also confirmed the proposed solution's capability to preserve users' privacy. All blinded attribute values have the same length. By using an appropriate hashing function, attributes with

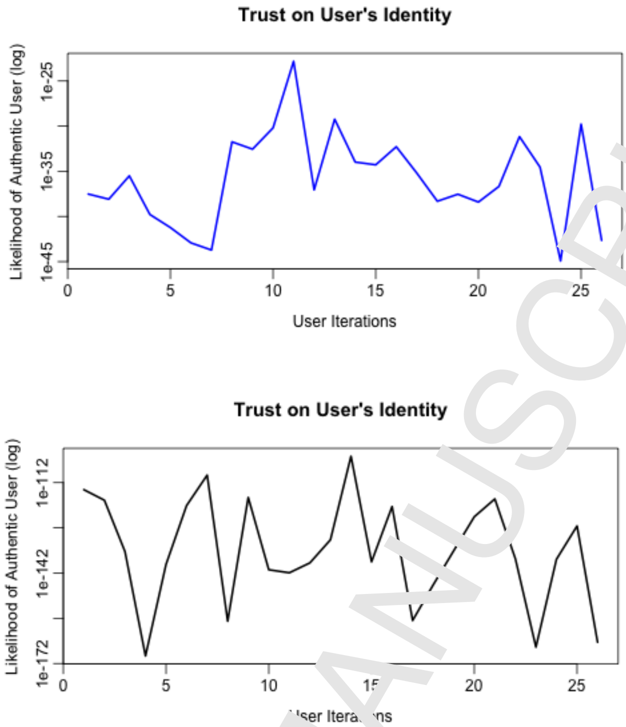

Figure 8: Overall confidence level evolution with user test rounds for the worst test cases TC3 (top) and TC4 (bottom).

e.g. four characters and more extended attributes with e.g. several hundred characters will all result in a 64 character blinded value (i.e. hash value), as shown in Table 6. Furthermore, SPs are provided with attributes in clear text if, and only if, the user consents. Additionally, the IdPP does not store any attribute value, but only attribute name pairs relations. The IdPP, also, does not see any attribute value, except the ones the user explicitly consents. Moreover, finally, when requesting attributes from an AP, the IdPP receives a pair with an attribute name and the hash value of the attribute value. Since it does not know anything about the semantics of the attributes asked, the information disclosed to the IdPP is just a string in some language (i.e. attribute name) and an LSH hash value (attribute value). All these features maximize the user's privacy during user-authentication processes.

Table 6: LSH signatures from different attribute value sizes example

$AttributeinClearText$	$BlindedValue$
Avenida Exemplo de Melo e Silva, 2371, Santa Maria, Rio Grande do Sul, Brazil	c20b9b68c490510204c101525999949cc0152907[illegible]05065[illegible]42ce83aba836a0498
Walter	100a0000140009810000000a02000200011[illegible]00008[illegible]000000000000000008008001

7. Conclusions

In this article, we have proposed a solution for the attribute aggregation problem in identity federations. The propose solution i) fits current deployed IdS scenarios, e.g. STORK, eIDAS; ii) is able to handle partially federated identity systems (i.e. scenarios where some APs require local authentication), iii) supports entities (SPs and APs) relying on different ontologies and languages, iv) preserves users' privacy while still providing results with high confidence levels.

The ability to handle several languages represents a step forward to applying this solution in cross-country scenarios. Although we have performed our experiments using English and German only, the employed API supports more than 90 languages. Albeit the accuracy of our implementation depends on the performance of the API, we believe that possible inaccuracies from the translation process can be overcome by adopting lower threshold values in the first step of the ontology alignment process. By lowering the threshold boundaries in initial ontology alignment, the solution eliminates false negatives that may occur, even due to poor translations, leaving the confidence level improvement algorithm the task of eliminating false positives.

Our solution also improves user privacy by not storing any user data on the UsIdS (Data Processor) and by avoiding the disclosure of attributes required by the matching process but not required by the SP. This is achieved by blinding attribute values using LSH functions. This kind of feature is relevant in our context, since the APs can have attribute values stored in slightly different forms such as abbreviations, and contractions. The only attribute values witnessed in clear text by the UsIdS are the ones the user authorized for disclosure to the Service Provider.

Finally, the third party Attribute Provider feature of our proposed solution promotes a greater level of possibilities in aligning attributes. Taking the diversity of Attribute Providers that each identity system can have in its ecosystem, the AP_T approach makes it possible to establish connections between APs whenever a direct link (a common key identifier) is not feasible.

Overall, our solution represents an encouraging improvement to the interoperability of electronic identities. While these solutions work nowadays on an agreed set of attributes, our solution enables an exchange of attributes between arbitrary identity systems and their entities. Additionally, it improves information quality provided to the SP in deciding to disclose, or not, a service to a user. Finally, it provides more chances to a successful identity linking process by using our AP_T approach. This way, our solution can be seen as a useful contribution to a new generation of interoperability solutions for electronic identities.

Funding

This work was partially supported by CAPES Proc. Num. 99999.009096/2013-02 and EU project Stork 2.0 CIP-ICT-PSP-2011-5-297263.

References

[1] C. E. Jimenez, A. Solanas, F. Falcone, E-government interoperability: Linking open and smart government, Computer 47 (10) (2014) 22–24. doi:http://dx.doi.org/10.1109/MC.2014.281.
URL http://dx.doi.org/10.1109/MC.2014.281

[2] S. Carmody, M. Erdos, K. Hazelton, W. Hoehn, B. Morgan, T. Scavo, D. Wasley, Incommon technical requirements and information (2005).

[3] T. W. House, National strategy for trusted identities in cyberspace: Enhancing online choice, efficiency, security, and privacy (2011).
URL https://pages.nist.gov/NISTIR-8149/nistir8149.html

[4] M. Ates, J. Fayolle, C. Gravier, J. Lardon, Complex federation architectures: Stakes, tricks & issues, in: Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology, CSTST '08, ACM, 2008, pp. 152–157. doi:10.1145/1456223.1456258.
URL http://doi.acm.org/10.1145/1456223.1456258

[5] W. Priesnitz Filho, C. Ribeiro, T. Zefferer, An ontology-based interoperability solution for electronic-identity systems, in: 2016 IEEE International Conference on Services Computing (SCC), 2016, pp. 17–24. doi:10.1109/SCC.2016.11.
URL http://dx.doi.org/10.1109/SCC.2016.11

[6] W. Priesnitz Filho, C. Ribeiro, Obtaining strong identifiers through attribute aggregation, in: SECURWARE 2014, The Eighth International Conference on Emerging Security Information, Systems and Technologies, 2014, pp. 96–100.
URL http://www.thinkmind.org/download.php?articleid=securware_2014_4_40_30155

[7] T. R. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 5 (2) (1993) 199–220. doi:10.1006/knac.1993.1008.

[8] K. Todorov, Combining structural and instance-based ontology similarities for mapping web directories, in: Internet and Web Applications and Services, 2008. ICIW '08. Third International Conference on, 2008, pp. 596–601. doi:10.1109/ICIW.2008.37.
URL http://dx.doi.org/10.1109/ICIW.2008.37

[9] W. Priesnitz Filho, C. Ribeiro, T. Zefferer, Towards privacy-preserving attribute aggregation in federated eid systems, in: CAiSE '16 Forum at the 28th International Conference on Advanced Information Systems Engineering, http://ceur-ws.org, 2016.
URL http://ceur-ws.org/Vol-1612/paper13.pdf

[10] L. De Vocht, M. Van Compernolle, A. Dimou, P. Colpaert, R. Verborgh, E. Mannens, P. Mechant, R. Van De Walle, Converging on semantics to ensure local government data reuse, in: 5th Workshop on Semantics for Smarter Cities, S4SC

2014 at the 13th International Semantic Web Conference, ISWC 2014, Vol. 1280, 2014, pp. 47–5[illegible], cited By 0.
URL http://www.scopus.com/inward/record.url?eid=2-s2.0-84909[illegible]199[illegible]&partnerID=40&md5=16edd6912c5ed5164b5cba8445a06ddf

[11] R. Rezaei, T. K. Chiew, S. P. Lee, A review on e-business interoperability frameworks, Journal of Systems and Software 93 (0) (2014) 199 – 216. doi:http://dx.doi.org/10.1016/j.jss.2014.02.004.
URL http://www.sciencedirect.com/science/article/pii/S016412121400051X

[12] R. Roberto Gatti, L. Carbone, V. Mezzapesa, State of play of interoperability ? report 2014, Tech. rep., European Union (2014). doi:10.2799/865679.
URL https://publications.europa.eu/pt/publication-detail/-/publication/d6d432ea-37a3-4434-b3f1-2d97f1b973b7

[13] T. Berners-Lee, J. Hendler, O. Lassila, et al., The semantic web, Scientific american 284 (5) (2001) 28–37.

[14] H. Nacer, D. Aissani, Semantic web services: Standards, applications, challenges and solutions, Journal of Network and Computer Applications 44 (0) (2014) 134 – 151. doi:http://dx.doi.org/10.1016/j.jnca.2014.04.015.
URL http://www.sciencedirect.com/science/article/pii/S1084804514001143

[15] H. Leitold, Challenges of eid interoperability: The stork project, in: IFIP PrimeLife International Summer School on Privacy and Identity Management for Life, Springer, 2010, pp. 144–150.

[16] L. Hind, D. Chiadmi, L. Benhlima, How semantic technologies transform e-government domain, Transforming Government: People, Process and Policy 8 (1) (2014) 49–75. arXiv:http://dx.doi.org/10.1108/TG-07-2013-0023, doi:10.1108/TG-07-2013-0023.
URL http://dx.doi.org/10.1108/TG-07-2013-0023

[17] F. Layouni, Y. Pollet, An ontology-based architecture for federated identity management, in: 2009 International Conference on Advanced Information Networking and Applications, 2009, pp. 162–166. doi:10.1109/AINA.2009.124.

[18] P. Shvaiko, J. Euzenat, Ontology matching: State of the art and future challenges, IEEE Transactions on Knowledge and Data Engineering 25 (1) (2013) 158–176. doi:10.1109/TKDE.2011.253.

[19] P. Arnold, E. Rahm, Enriching ontology mappings with semantic relations, Data & Knowledge Engineering 93 (2014) 1 – 18, selected Papers from the 17th East-European Conference on Advances in Databases and Information Systems. doi:http://dx.doi.org/10.1016/j.datak.2014.07.001.
URL http://www.sciencedirect.com/science/article/pii/S0169023X14000603

[20] S. Pearson, Privacy, Security and Trust in Cloud Computing, Springer London, London, 2013, pp. 3–42. doi:10.1007/978-1-4471-4189-1_1.
URL http://dx.doi.org/10.1007/978-1-4471-4189-1_1

[21] A. Gholami, E. Laure, Big data security and privacy issues in the cloud, International Journal of Network Security & Its Applications (IJNSA), Issue January.

[22] OECD, The oecd privacy framework, Tech. rep., OECD (2013).
URL http://www.oecd.org/sti/ieconomy/oecd_privacy_framework.pdf

[23] F. Paci, R. Ferrini, A. Musci, K. Steuer, E. Bertino, An interoperable approach to multifactor identity verification, Computer 42 (5) (2009) 50–57. doi:10.1109/MC.2009.142.
URL http://dx.doi.org/10.1109/MC.2009.142

[24] N. Shang, F. Paci, E. Bertino, Efficient and privacy-preserving enforcement of attribute-based access control, in: Proceedings of the 9th Symposium on Identity and Trust on the Internet, IDTRUST '10, ACM, New York, NY, USA, 2010, pp. 63–68. doi:10.1145/1750389.1750398.
URL http://doi.acm.org/10.1145/1750389.1750398

[25] J. David, J. Euzenat, F. Scharffe, C. T. dos Santos, The alignment api 4.0, Semantic Web journal 2 (2011) 3–10. doi:10.3233/SW-2011-0028.

[26] J. Euzenat, An api for ontology alignment, in: The Semantic Web-ISWC 2004, Springer, 2004, pp. 698–712.

[27] S. Malik, N. Prakash, S. Rizvi, Ontology merging using prompt plug-in of protégé; in semantic web, in: Computational Intelligence and Communication Networks (CICN), 2010 International Conference on, 2010, pp. 476–481. doi:10.1109/CICN.2010.151.

[28] N. F. Noy, M. A. Musen, Algorithm and tool for automated ontology merging and alignment, in: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-00). Available as SMI technical report SMI-2000-0831, 2000. URL https://vvvvw.aaai.org/Papers/AAAI/2000/AAAI00-069.pdf

[29] W. E. Djeddi, M. T. Khadir, S. B. Yahia, Xmap: results for oaei 2015., in: OM, 2015, pp. 216–221.

[30] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, in: Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98, ACM, New York, NY, USA, 1998, pp. 604–613. doi:10.1145/276698.276876.
URL http://doi.acm.org/10.1145/276698.276876

[31] M. Datar, N. Immorlica, P. Indyk, V. S. Mirrokni, Locality-sensitive hashing scheme based on p-stable distributions, in: Proceedings of the Twentieth Annual Symposium on Computational Geometry, SCG '04, ACM, New York, NY, USA, 2004, pp. 253–262. doi:10.1145/997817.997857.
URL http://doi.acm.org/10.1145/997817.997857

[32] Q. Lv, W. Josephson, Z. Wang, M. Charikar, K. Li, Multi-probe lsh: Efficient indexing for high-dimensional similarity search, in: Proceedings of the 33rd International Conference on Very Large Data Bases, VLDB '07, VLDB Endowment, 2007, pp. 950–961.
URL http://dl.acm.org/citation.cfm?id=1325851.1325958

[33] P. Boufounos, S. Rane, Secure binary embeddings for privacy preserving nearest neighbors, in: Information Forensics and Security (WIFS), 2011 IEEE International Workshop on, 2011, pp. 1–6. doi:http://dx.doi.org/10.1109/WIFS.2011.6123149.
URL http://dx.doi.org/10.1109/WIFS.2011.6123149

[34] C. Blundo, E. De Cristofaro, P. Gasti, Espresso: Efficient privacy-preserving evaluation of sample set similarity, in: R. Di Pietro, J. Herranz, E. Damiani, R. State (Eds.), Data Privacy Management and Autonomous Spontaneous Security, Vol. 7731 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 89–103. doi:10.1007/978-3-642-35890-6_7.
URL http://dx.doi.org/10.1007/978-3-642-35890-6_7

[35] J. Zhang, H. Lu, X. Lan, D. Dong, Dhtnil: An approach to publish and lookup nilsimsa digests in dht, in: High Performance Computing and Communications, 2008. HPCC '08. 10th IEEE International Conference on, 2008, pp. 213–218. doi:10.1109/HPCC.2008.26.

[36] Z. Jianzhong, Y. Boyang, Z. Hongbo, L. Xiaofeng, Peernil: An approach to publish and lookup nilsimsa digest in chord, in: Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on, 2008, pp. 202–207. doi:10.1109/CHINACOM.2008.4685003.

[37] J. Oliver, C. Cheng, Y. Chen, Tlsh – a locality sensitive hash, in: Cybercrime and Trustworthy Computing Workshop (CTC), 2013 Fourth, 2013, pp. 7–13. doi:10.1109/CTC.2013.9.

[38] A. Azab, R. Layton, M. Alazab, J. Oliver, Mining malware to detect variants, in: Cybercrime and Trustworthy Computing Conference (CTC), 2014 Fifth, 2014, pp. 44–53. doi:10.1109/CTC.2014.11.

[39] H. Leitold, B. Zwattendorfer, STORK: Architecture, Implementation and Pilots, Vieweg+Teubner, Wiesbaden, 2011, pp. 131–142. doi:10.1007/978-3-8348-9788-6_13.
URL http://dx.doi.org/10.1007/978-3-8348-9788-6_13

[40] V. Koulolias, A. Kountzeris, H. Leitold, B. Zwattendorfer, A. Crespo, M. Stern, Stork e-privacy and security, Proceedings

- 2011 5th International Conference on Network and System Security, NSS 2011 (2011) 234–238doi:10.1109/ICNSS.2011.6060006.
URL http://dx.doi.org/10.1109/ICNSS.2011.6060006

[41] S. El Haddouti, M. D. E.-C. El Kettani, Towards an interoperable identity management framework: a comparative study, International Journal of Computer Science Issues (IJCSI) 12 (6) (2015) 98–106, copyright - Copyright International Journal of Computer Science Issues (IJCSI) Nov 2015.
URL http://www.ijcsi.org/papers/IJCSI-12-6-98-106.pdf

[42] C. Esposito, Interoperable, dynamic and privacy-preserving access control for cloud data storage when integrating heterogeneous organizations, Journal of Network and Computer Applications 108 (2018) 124 – 136. doi:https://doi.org/10.1016/j.jnca.2018.01.017.
URL http://www.sciencedirect.com/science/article/pii/S1084804518300316

[43] I. Diffeo, py-nilsimsa: Python implementation of nilsimsa locality-sensitive hash (2014).
URL https://code.google.com/archive/p/py-nilsimsa/

[44] B. Sigurd, M. Eeg-Olofsson, J. Van Weijer, Word length, sentence length and frequency – zipf revisited, Studia Linguistica 58 (1) (2004) 37–52. doi:10.1111/j.0039-3193.2004.00109.x.
URL http://dx.doi.org/10.1111/j.0039-3193.2004.00109.x

Authors Biography

Thomas Zefferer received his PhD in Computer Science from the Graz University of Technology. He is Senior IT Security Expert at the Secure Information Technology Center – Austria (A-SIT). His research interests include IT security, e-Government, risk assessments, and topics related to mobile and cloud security. He is the author of numerous scientific papers published in peer-reviewed international journals and conference proceedings.

Carlos Ribeiro received the BSEE degree in 1989, the MSc degree in 1993, and the PhD degree in computer science in 2002, all from the Technical University of Lisbon (IST/UTL, Instituto Superior Técnico), Portugal. He is a professor in the Computer and Information Systems Department at IST/UTL, where he teaches operating systems and computer architecture classes. From 1995 to 1998, he was a security adviser for the Portuguese National Security Authority. He has been a researcher at INESC since 1988, where he has participated in several European projects. His main research area is security, although he is also interested in distributed operating systems and mobility.

Walter Priesnitz Filho is a PhD student in Information Security at the Instituto Superior Técnico of Universidade de Lisboa, Portugal. He received his Bachelor degree in Information Systems in 1999 from Centro Universitário Franciscano (UNIFRA) and his master's degree at Computer Science in 2003 from Universidade Federal de Santa Catarina (UFSC). He is a professor at the Federal University of Santa Maria (UFSM) - Brazil, where he teaches application protocols and network management classes. His research interests include IT security, e-Government, and topics related to attribute aggregation in e-ID systems.

Highlights

The proposed solution enables a privacy-preserving attribute aggregation, using ontology-alignment approaches with a history-based improvement function, Locality Sensitive Hashing (LSH) functions, and Third Party Attribute Providers. The presented solution is compatible to current eID Federations. It can handle partially federated scenarios (scenarios where some attribute providers require local authentication). The solution also handles entities (service providers and attribute providers) with different ontologies and languages. Moreover, it does so without compromising privacy, which nevertheless provides results with high confidence levels.