A design towards personally identifiable information control and awareness in OpenID Connect identity providers

Rafael Weingärtner and Carla Merkle Westphall
Department of Informatics and Statistics
Federal University of Santa Catarina
Florianópolis, Santa Catarina, Brazil
E-mails: rafael.weingartner@posgrad.ufsc.br, carla@lrg.ufsc.br

***Abstract*—Cloud computing usage has increased in recent years as a consequence of its benefits such as agility on resource provisioning, elasticity, and reduced costs. However, once organizations migrate to cloud environments they lose control of the underlying structure of their applications such as physical networking, storage, and servers. Therefore, the cloud may pose privacy threats to sensitive data that are used to identify users; hence, data stored in identity providers that are deployed on cloud platforms may be accessed by curious/malicious system administrators. We present a design that addresses some privacy issues within the personally identifiable information that is stored in identity providers of federated identity management systems. Further, we provide an overview of the addition of the proposed design into the MITREid OpenID Connect implementation developed by MIT.**

***Keywords*-Identity federation; cloud computing; identity providers; privacy of PIIs; OpenId Connect**

I. Introduction

Cloud computing is widely used to provide on-demand services, bringing reduced costs, flexibility and agility on resource provisioning. Organizations that use federated environments can take advantage of those benefits to speed the development and deployment of new services. However, once user's personally identifiable information (PII) is sent to identity providers (IdPs), she/he may lose control on how that data is disclosed, stored and used. Malicious system administrators are the main cause of that loss of control over user's PIIs. Moreover, as discussed in [1], data stored in the cloud may be sensitive and if linked with its owner identity may violate his/her privacy. That means, if IdPs are corrupted (by an attacker) leading to the usage of users identification data to violate their privacy.

There have been cases of security breaches as noticed in [2], [3], and [4] that resulted in identification data leaks. For this reason, identity management systems (IMS) should focus on protecting the privacy of PII data and providing control of that data to users. Although those security breaches did not happen in IdPs, they can be a good example of what can happen if an attacker gains access to IdPs' PIIs data tables.

Users as owners of PIIs that are stored in IdPs have the right to effectively manage that data and have to be aware of any processing and disclosure that might happen. There are laws that aim to provide users with awareness and protect them against unwilling data processing and dissemination. However, current IMS do not have mechanisms to provide such guarantees [5], [6], [7].

Neither OpenID Connect [8] nor Security Assertion Markup Language (SAML) and their implementations such as MITREid [9], Shibboleth [10] or OpenAM [11] deal with privacy upon PIIs data that are stored in IdPs. Our design proposal uses cryptography to store PIIs data into IdPs, being implemented on top of OpenID Connect protocol using the MITREid application.

This paper addresses privacy threats found on IdPs of federated environments. We propose a design to improve privacy capabilities of identity providers of OpenID Connect federations. Our contributions are the following: (i) we provided control to users over their PII in the dissemination process; (ii) we also created mechanisms to control users' PIIs that are stored in IdPs and (iii) we improved user support during the dissemination process, providing guidance to users, lowering the risks of unaware/unintentional data dissemination. All of those contributions are presented in Table I, enabling the comparison with related works presented in this paper.

The remainder of this paper is structured as follows. Section II provides an overview of the concepts that are used throughout this paper. Section III presents and discusses related works and how they compare with ours. Section IV presents and discusses the current identity management model that is most deployed and used to create federated environments and its issues. Section V describes and presents the model to address privacy in IdPs. Section VI provides an overview of the extensions we developed to add our proposals into the OpenId Connect protocol. Section VII concludes the paper and presents future works.

II. Background

This section presents an overview of privacy and identity management concepts.

978-1-5386-0958-3/17 $31.00 © 2017 IEEE
DOI 10.1109/CIT.2017.30

A. Identity management

A user identity can represented as a PII that is a set of attributes that can be used to identify a user; e.g. social security number, credit card numbers, telephone numbers and address [12]. Identity management is the process of creation, management and use of user identities and the infrastructure that provides support for this set of processes [13]. Additionally, Chadwick in [14] defines that a federation can be created with an association of service providers (SPs) and identity providers (IdPs) through the establishment of trust between the parties involved. A federation enables users to access resources in different administrative domains authenticating in their home domains in a transparent and secure manner [5].

A federation has two major entities; (i) identity providers, which have the responsibility to manage users PIIs, perform the authentication process and disseminate data to SPs; (ii) service providers, which are the entities that deliver the resource/service to users and commonly performs the authorization using the PIIs that are disseminated from IdPs.

B. Privacy

Landwehr et al. [15] define privacy as the control of the release of personal data that users should have. In addition, privacy can be defined as a state in which one is free from interference and monitoring of unwanted/uninvited entities. Moreover, the same rights that people have offline must also be extended to the online environment [16].

There are plenty of laws that aim to protect user privacy on the Internet. In Europe, there is the Data Protection Directive [17]; in the USA, the Health Insurance Portability and Accountability Act (HIPAA) [18] and in Brazil, the Internet Bill of Rights [19]. All of those aforementioned laws aim to protect users against unwilling data disclosure and processing.

III. Related works

Orawiwattanakul et al. in [5] tackled the lack of control on PII disclosure in federations. They enabled users to select among all non-mandatory attributes, which ones users wish to disclose to the SP that is being accessed. They developed their approach on top of a Shibboleth well-known extension called uApprove [20].

Sánchez et al. in [21] proposed a reputation protocol that weights the reputation of entities in a federation in order to support data disclosure. This way, users can check SPs reputation among federation members before they send any data to it. They also provided a way in which users would have the ability to check what is being done with their data, and based on that they could lower or increase the provider reputation.

Betgé-Brezetz et al. in [22] proposed a method to define if a user trusts or not in a cloud provider and the level of trust. Based on how much the user trusts the cloud provider, she/he could send data in plain text, partially encrypted (encrypted with some metadata in plain text) or fully encrypted to the cloud. Paper [22] also proposed a package called PDE (Privacy Data Envelope) to carry user data to the cloud. That package holds the data (encrypted or not) with policies that state how, where, by whom and when it can be used.

Chadwick and Fatema in [23] proposed a series of web services that analyze policies that are uploaded within the data before any action is executed. Therefore, once an application receives a request to process some data, the application should consult proposed web services if it can proceed with the requested action.

Betgé-Brezetz et al. in [24] combined the approaches of reputation presented in [22] with policies presented in [23]. They used stick policies with the PDE proposed in [22] to carry altogether policies and data to the cloud. The proposal adds on cloud service providers points that evaluate those policies before using the data, these points are called data protection module (DPM) which would guarantee the evaluation of defined policies before any data processing. Paper [24] also defined that the PDE containing the policies and data would just be sent (processed, copied and stored) into cloud nodes that have the DPMs modules deployed.

Having presented related works we compare them with our proposal using Table I; we can categorize papers as having one or more of the following properties:

- Use of cryptography – use of cryptography to store data at a provider;
- Based on reputation – use of reputation to support users' decision of which data and how it is sent to SPs;
- Use of Policies – policies that regulate how data is used/disclosed at a provider;
- Disclosure awareness – provide feedback to make users aware of data dissemination;
- Disclosure support – provide means to support users when they are disseminating data from an IdP to SPs;
- Used in IMS – it indicates if the proposal found in the paper was implemented in an IMS.

Table I
Related works properties

	Characteristics					
Papers	Use of cryptography	Based on reputation	Use of Policies	Disclosure awareness	Disclosure support	Used in IMS
[5]			X	X		X
[21]		X				
[22]	X	X				
[23]			X			
[24]	X	X	X			
[6]	X		X	X	X	X
Our work	X	X	X	X	X	X

Table I matches aforementioned properties with presented related works. Our design proposal strives to enhance the support and privacy in IdPs using cryptography to provide

control over PIIs stored in IdPs, policies to disclose user PIIs and to provide support during the disclosure process through the use of SPs reputations. Additionally, our proposal was added and tested into MITREid.

IV. Current identity management model

We found an abstract overview of the current identity management systems in [7]. Shibboleth and MITREid can be cited as examples of IMS. Figure 1 shows in more details the operation of current identity management systems when a user is requesting a protected service/resource.

The browser in Figure 1 acts in favor of the user. That component has to be carefully chosen and maintained, in order to avoid attacks. Moreover, in Figure 1 is presented the SP as an aggregation of security manager (SM) and application per se. The security manager has to intercept all requests and check if they have a security context (SC) if they do not have an SC, the SM redirects the user to the authentication process in an IdP. The steps executed in Figure 1 are described as follows:

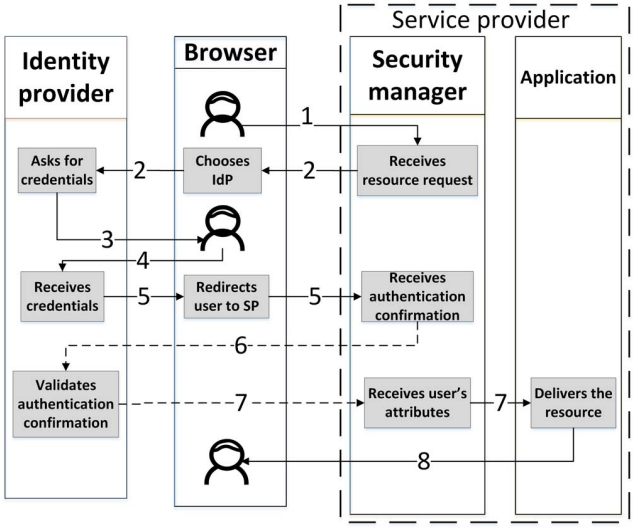

Figure 1. Current federated identity management systems

1) The user requests a protected resource at an SP – security manager (SM) intercepts the request and checks for a security context (SC);
2) The SP requests user authentication – there is no SC in the protected resource/application, the SM has to request the user to authenticate with an IdP in which the user is enrolled;
3) Authentication process – IdP asks the user about his/her credentials to perform the authentication process;
4) Consensus and awareness – when the authentication process completes successfully, the IdP informs the user the reason for the data dissemination and obtains the permission to release attributes to the SP;
5) IdP sends to SP a proof of authentication – the IdP creates a ticket to confirm the authentication process in the IdP to send to the SP, then the IdP redirects the user (browser) to the SP with the proof of authentication;
6) Validation of the proof of authentication – SM intercepts the request, validates the proof of authentication and obtains extra data about the user (PIIs);
7) Gathering of user's attributes – PIIs are delivered to the SP that performs the authorization process;
8) The desired service/resource is delivered.

Steps six (6) and seven (7) happen without user awareness. They are executed solely by the IdP and SP in order to guarantee that the user is not trying to change the information exchanged during the authentication and dissemination processes.

The OpenID Connect was chosen to serve as the basis for our implementations mainly because it is open source, maintained by a foundation, and it has a standard protocol for messages exchanged between IdP, user and SP. OpenID connect also uses a lightweight message format, JavaScript Object Notation (JSON) which is faster than the usual SAML that uses XML [25].

Despite the fact that OpenID Connect uses terms OpenID provider (OP) and relying upon party (RP) to represent respectively IdPs and SPs, throughout this paper we maintain the use of IdP and SP terms that are widely known and used in the literature.

A. Risks to privacy in current federated identity management systems

In current IMSs, there are problems regarding the privacy of PIIs that are stored in IdPs such as:

- Lack of control over user PIIs: users do not have effective means to manage their attributes that are stored in IdPs [21];
- Absence of transparency in the dissemination process: laws previously mentioned state that users should be aware of the release of PII data to SPs [5]. This problem is present in current systems such as Shibboleth, when it is used without extensions such as uApprove [20];
- Lack of support during the dissemination process: Zhang et al. in [26] presented that users fail to define their policies for PII dissemination. Moreover, Hansen, Schwartz and Cooper in [27] argued that a single default setting for attributes disclosure will not meet the needs of all users. Thus, users should be able to customize IdPs, defining which attributes can be sent to each SP.

The lack of control that users have over their sensitive data increases once they migrate to cloud services. As presented by Mather, Kumaraswamy and Latif in [28], once organizations have migrated to cloud services, they lose control

over their structure, relying on third parties administrators. Moreover, Zhang et al. discussed in [29] that the loss of control of the underlying structure can lead to data leaks as a consequence of curious/malicious system administrators of those systems. Moreover, there have been cases of security breaches as published in [2], [3], and [4] that resulted in identification data leaks

B. Threat modeling

Knowing the threats that may exist in a system helps us define and select security measures that have to be developed and deployed to secure it [30].

One of the methodologies used to perform threat modeling is known as STRIDE and developed by Microsoft [31]. However, that methodology does not cover threats related to privacy. Therefore, in our work we used the extended methodology presented in [30] which adds privacy aspects in STRIDE methodology.

Methodologies presented in [30] and [31] require a description of the application (IdP), its dependencies, actors that interact with the application and their actions, endpoints of the application that receive requests and resources that have to be secured. The resource that has to be secured in the application (IdP) is the user's PII data. The IdP has few dependencies such as a web container (Tomcat[32]) and interacts with non-authenticated users, logged users, SPs and system administrators.

The data flow diagram (DFD) of IdPs interactions with mapped actors is presented in Figure 2. Developers and administrators (container, network, server and database) have to perform their duties without accessing PIIs data of users. Users can get enrolled into the IdP presenting their PII attributes that are required for the registration; depending on the IdP the user may have to send more or less data to create an account. Users that are already enrolled can present their credentials and use the IdP functionalities. SPs interact with IdPs to confirm user identity and to retrieve some of user PII attributes that SPs require to provide a service.

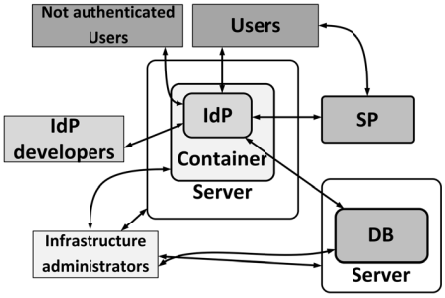

Figure 2. Data flow diagram of IdPs interactions with mapped actors.

After we had mapped IdPs dependencies, actors, endpoints and resources we created the threat tree of the privacy threats that the IdP is vulnerable and that we needed to deal with. The root level of a threat tree represents the problem we want to tackle; the second level is the cause that creates the problem and the third level shows the consequence if the problem is not addressed.

There is a lack of mechanisms to ensure control of user attributes that are stored in IdPs. Figure 3 shows the threat tree that illustrates the aforementioned problem, which may be caused by the following factors:

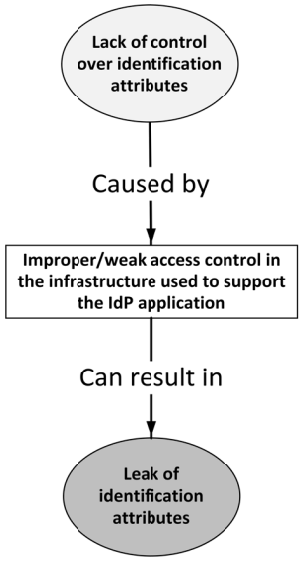

Figure 3. Threat: lack of users control over their PII data store in IdP

- Improper/weak access control in the database – if it is granted access to the PII data table to some account, it has unrestricted and unlimited access to those PIIs. If that account does not have a proper password set or if the password is weak, an attacker could take advantage of that and get access to the PII data table;
- Improper/weak access control in database server – the administrator of the server in which the database runs or an attacker that gains access to that machine that can have access to the database files in the file system and can use them to recreate the database somewhere else;
- Improper/weak access control in application's server the administrator of the server in which the IdP application runs or an attacker that gets access to that machine has access to the server memory. Therefore, she/he can dump the memory and uncover user PII data that may have been loaded into memory.

One could argue that if we improve the security mechanisms on the underlying structures (servers, databases, application servers) we could address the aforementioned problem; and that is true, by using better security measures, we can lower the risk of a hack. However, the system administrators still have unlimited access. We could deploy an audit mechanism, which would rely on someone or some

third party to deploy and maintain. Our design goal is to address the aforementioned situation, without needing to rely on such mechanism.

V. Model to provide privacy and control of PIIs in identity providers

The government has the need to manage and organize the PII data of its agents and users. However, the centralization of such information of government agents may attract attackers; data about presidents, governors, senators and other vital positions for national security would be concentrated at a single point (IdP). Our design proposals enhance the identity flow between IdPs and SPs, using an extra layer of cryptography that is centered on users. We want to soften the harm that breaches such as the hack of U.S. voters database [33], and others [4], [2], [3] can cause. Not to mention the insider threats that is may exist and never gets exposed.

There are plenty of implementations of identity management systems such as Shibboleth, OpenAM and MITREid; however, those systems do not provide protection to situations such as the aforementioned ones.

In order to simplify the understanding and presentation our design was divided into two parts: the first focus on addressing user awareness and control of PIIs that are stored into IdP; while the second presents the use of reputation to guide users during the PII disclosure process.

A. A design to address PII control and awareness in IdPs

To address user awareness and PII control we used a combination of cryptography approach proposed in [22] and [24], with the proposal to provide awareness and user control presented in [5].

To access federated services, users are required to get enrolled with an IdP that validates their credentials and PII data, performs the authentication process and dissemination of PII data to SPs. The flow of authentication, dissemination and service usage are presented in Figure 1. When users register or get enrolled in an IdP, we assume that they present a set P of PII data, being $p \in P$ a single PII.

Our design proposal changes the current identity management model presented in Section IV. The proposal intends to reduce threats to the privacy of user PIIs that are stored in IdPs. Figure 4 presents the design; we assume that all of the communication between users, IdP, SP and validation services are performed over an SSL/TLS channel. Thus, it requires a change in the way PIIs data are stored and used. The PII has to be encrypted with a key that just the user has access.

Depicted in Figure 4 is the whole cycle of creation, validation and use of identity attributes. In our proposal, the identity life cycle can be divided into three (3) distinct phases. Phase one (1) is the process of enrollment of the user within an IdP, sensitive attributes should be encrypted in this process; phase two (2) is the validation process that

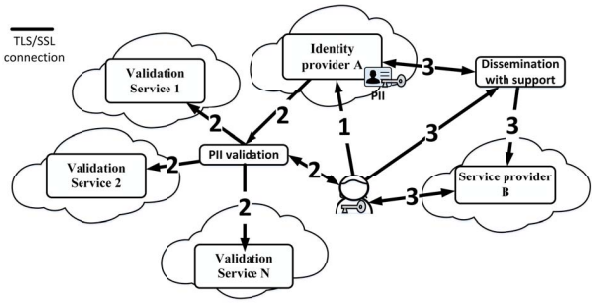

Figure 4. Design to provide PII control and awareness in federations

does not exist in current identity management systems. The validation process is required to validate the encrypted PIIs that are stored by the IdP; phase three (3) is the use of PIIs, this is the moment in which users can select which attributes are going to be disclosed to SPs and open the encrypted ones to be disclosed.

We propose that users encrypt their PII data before it is sent to IdPs. This way, users have control over the data that is stored in IdPs, which prevents internal agents to use or disseminate PIIs without prior consent and knowledge of the user. When users are registering with an IdP they encrypt their sensitive PIIs data (e.g. SSN, credit card numbers, telephone number and address) with a key, and later when disclosing that data to some service, they would be asked to open it. Therefore, during the process of enrollment, an agent running on the user's side encrypts her/his sensitive PIIs using Equation (1). The cryptographic primitive used in Eq. (1) is the public key cryptography.

We used the following notation to present Equation 1:

- ps – is a sensitive attribute and $ps \subset P$ that is encrypted;
- $K1$ – is the user's key used to encrypt ps;
- ps' – is the result of $E(K1, Ps)$;
- E – is the function used to encrypt the data;
- $ECPS$ – is the set of ps'.

$$E(K1, ps) = ps' \tag{1}$$

During that process, we also retrieve the hash of each ps element, this way we can check if the user is modifying them during the disclosure process. The hashing process occurs before the execution of the Equation (1). The hashing process is described in Equation (2), we used the one-way hash cryptographic primitive.

The following notation is used to present the hashing Equation 2:

- H – is the hash function;
- Hps – is the hash of the ps element;
- $salt$ – is a random generated set of bits used to increase the system entropy;

- $SHPS$ – is the set of Hps that is sent to the IdP.

$$H(ps, salt) = Hps \tag{2}$$

Our design proposal to store encrypted attributes in IdPs does not mean that the IdP is not trustworthy; hence, if there is no trust there would be no federation. However, even if we trust an organization, it does not necessarily mean that we have to trust all of its employees that have access to personal data it stores.

Encrypted attributes make it impossible for the IdP to validate and ensure the accuracy of the data it stores. We propose the use of specialized SPs to validate attributes, called Validation Services (VSs) that have the responsibility to validate user's attributes. VSs do not store any user PII data; they are deployed into the system solely to check the accuracy and validity of the data that is presented by users.

VSs are designed to validate specific attributes such as email addresses, social security numbers, credit card numbers and other PII data that may be considered sensitive. In order to validate one of the user's PII, the IdP provider directs the user to access the VS responsible to that specific attribute validation to open and disseminate the data to be validated. This process happens once for each one of the sensitive PIIs that need to be validated.

We do not intend to create a new type of organization to act as VS. Civil registration agencies, banks, and other governmental and non-governmental agencies which issue user data can act as VS. As a matter of fact, they already do. For instance, when presenting a birth certificate to a school; the school may check the validity of presented information with the issuing party; the same occurs when presenting university degrees for an employee. We base the proposal design on the same principles.

The validation process occurs asynchronously. First, there is a validation request, sent from the IdP to the VS, the user is active in this process, she/he has to start the process in the IdP. The validation request generates a token u that identifies the validation request. The token u is unique for each validation request and it is used by the IdP, to question the VS about the validation result. Later, the IdP questions the VS about a particular validation with the token u and stores the result.

The validation process requires users to open a ps' using Equation (3), which will result in the original ps encrypted with Equation (1). The ps is sent to the VS to proceed with the validation process, it is also generated a token u to identify that validation request. On the VS side, the validation process may require some extra data $d, d \neq ps$ and $d \notin P$.

An example of d is a code that is used to confirm the ownership of an email account or cellphone. To validate an email account, the user would have to start the validation process to confirm the ownership of an email account; the email address is disclosed to the VS through the dissemination process. The VS sends an email with a code; then the user has to present that code to the VS.

After the validation, the IdP requests the VS about the validation result of a ps', using the token u. The IdP receives a true or false that indicates if the data ps' was either validated with success or not.

Because we changed the way sensitive data are stored into IdP, we had to introduce VSs that is a trusted third party, into our model to validate the data entered into the IdP by users. The VS module still requires more work, they may impose some threats to users data, hence, they have access to the raw data entered by users into the IdP. However, the harm that an attacker can do if it gains access to VS is lower than the harm of an attack into an IdP that has all of the users data unencrypted. The data is stored temporally until the validation process ends, and after that, it just has a logic value denoting success or not of the validation process.

Another change that was introduced in the dissemination process between IdPs and SPs, we needed to provide means to users to open their sensitive PIIs; otherwise, attributes would be sent to the SP encrypted.

The process of decryption happens after the authentication, during the dissemination phase, where users can decide which attributes will be sent to the SP. The user may have to open some ps' that she/he wants to disclose to the SP. The Equation (3) is used to open ps'.

We use the following notation in Equation 3:

- D – is the decryption function used to open ps' with a key $K2$;
- ps – is the data initially presented by the user;
- $K2$ – is the second key of the pair of keys that are used to encrypt and decrypt the PII.

$$D(K2, ps') = ps \tag{3}$$

After the opening of ps' with user's key $K2$, it is also possible to select others p elements to send to the SP, creating a set $S \subseteq P$. We need to encrypt S elements to send to the SP, hence, the IdP is the entity responsible for delivering it to the SP and we do not want the IdP knowing the contents of S. Therefore, we apply the Equation (4) on $s \subset S$, then we send s' to the IdP to be delivered to the SP.

We use the following notation to present the temporary encryption Equation (4):

- $tempE$ – is the encryption function used to temporally encrypt s with a key SK. This function uses the symmetric key cryptography primitive, in which the same key that is used to encrypt the data is used to decrypt it;
- s – is the data that is sent to the SP;
- SK – is the session key used to encrypt the data and shared between user (browser) and SP;
- s' – is the result of $tempE(SK, s)$;

- S' – is a set of s'.

$$tempE(Sk, s) = s' \tag{4}$$

When we send to the IdP the set S' it answers with a token that the user should present to the SP. The agent that runs in the user's browser retrieves this token and send it together with the key SK to the SP, this way the SP can open the S' data.

The key SK is encrypted with the SP's public key $SPPK$, we used the Equation (1) to encrypt the SK. Instead of the parameter $K1$, we use the $SPPK$; and instead of the parameter ps, we use the SK, this process will result in SK that is then sent to the SP. The SP uses the Equation (3) to retrieve the SK and open the disclosed data, changing the parameters $K2$ and ps' respectively to SP private key $SPPK2$ and session key encrypted SK', the result of the function will be the original SK.

The IdP adds the $HpsS$ of each ps to $SHPS$ when responding the validation request to the SP. The SP retrieves that data and opens it with Equation 5

$$tempD(Sk, s') = s \tag{5}$$

The SP validates each element of S with its corresponding hash Hps, it computes the hash of s and check if it matches the hash of the received s, if everything is OK the SP can perform the authorization process and deliberate about the delivery of the service/resource.

B. Design to support users during the dissemination process

To address the lack of support in the dissemination process we used a combination of the reputation proposals found in [21] and [22] with the policies proposals to control the use of data presented in [24] and [23].

Birrel and Schneider in [34] argue that controlling the dissemination of PIIs can become inconvenient because it forces users to decide recurrently which data can be sent to which SPs. Moreover, in [26] is presented that for users it is difficult to successfully establish their PII dissemination policies. Hansen, Schwartz and Cooper in [27] add that a single default setting for attributes release will not properly fit the different needs of users. Therefore, we propose that IdPs use the reputation of SPs to assist users in which data sets are better or worse to be disseminated.

This paper does not propose a method for measuring the reputation of entities in a federation. We assume that there is a reputation measurement system in place. Our design can use those information to guide users during the dissemination process, as illustrated in Figure 5. The reputation measurement system may be an implementation of works [21] or [22] where it is presented models and methods to quantify reputation among federated entities. The dissemination with reputation process is the following:

- First, the user tries to access a protected service/resource. User is not authenticated at the SP, then it is redirected to the IdP to executed the login process;
- then, the users has to present his/her credentials;
- after the authentication process, the IdP load the SP's reputation from the reputation measurement system;
- the user is re-directed to the dissemination screen, and selects which PIIs she/he wants to release to the SP; during this process, based on the SP's reputation, the IdP tries to guide users to select only the recommended PIIs. The users still have the freedom to choose and disseminate the PIIs she/he wants; even if the release is not recommended by the IdP policies;
- after that, the selected PIIs are disseminated.

The reputation system also has other processes such as user data gathering and SPs audit service. Those phases are used to create a score that represents the SP's reputation. The design for such tasks are out of this scope, and can be found at [21] or [22].

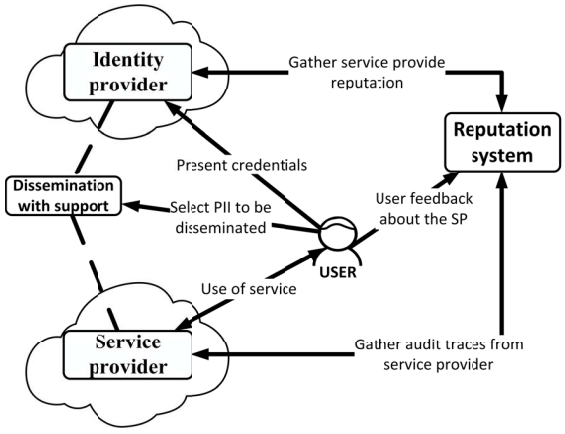

Figure 5. Providing support during the dissemination process

The protocol used in this paper (OpenID Connect) has an object called scope, which represents a set of data to be sent to the SP, and the SP may request more than one set of data at the same time. Users can choose the requested sets of attributes that are disseminated to the SP. Thus, we extended that object, adding an attribute that represents the level of reputation required for an SP to receive the scope. Using reputation metrics, IdPs can create policies to guide users during the disclosure process, helping them decide which attributes are more appropriate to be disseminated.

We modeled the scope object as tuple $O\{pii, R\}$, where, $pii \subset P$ the PII attributes that may be disclosed to the SP, a pii may also contain any $ECPS$ data, hence $ECPS \subset P$; R is the reputation that is the required to receive the scope O. The SP can request more than one scope to the IdP, and the user deliberates about the disclosure of the scope and its attributes according to IdPs recommendations.

VI. Adding the proposed design into OpenId Connect

This section describes changes that were added to the MITREid framework that implements the OpenId Connect protocol in Java. We briefly describe the implementation and frameworks we used to apply the design in Section V.

Moreover, the proposal requires that users during the IdP enrollment process encrypt sensitives PIIs; however, normal attributes storage systems that could be used such as Lightweight Directory Access Protocol (LDAP) do not provide that functionality. Therefore, we had to create an administrative system that provides means to users for encrypting their sensitive PIIs with a key. Figure 6 presents the system we created to allows users to execute that task.

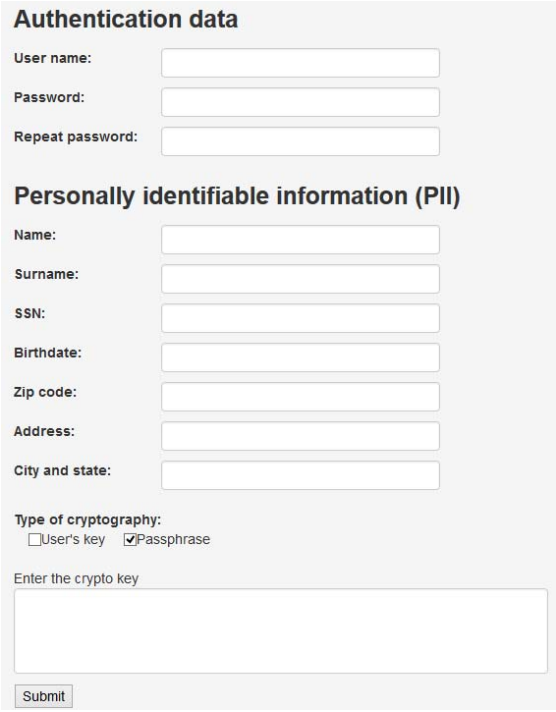

Figure 6. User enrollment with encrypted PIIs.

As presented in figure 6, we provided two methods for users to encrypt their attributes; (i) we generate a pair of keys with a given passphrase; (ii) the user enters a key from a pair of keys that the user already owns.

To apply the proposals presented in Section V-A we developed a set of libraries using Java, Java Server Pages (JSPs), HTML and JavaScript.

The Equation 1 was developed using the Cryptico library [35]. However, we changed the key derivation method that was used to derive a pair of keys; instead of a simple Sha-256 hash of the pass phrase to derive the keys, we applied PBKDF2 with a sufficiently large numbers of iterations to strengthen the system against brute force attacks, the number of iteration used was 20483, we tried with a few different numbers of interactions, starting at 1000, scaling in prime number bases, we found the number 20483, which seemed to create an effective computational cost while maintaining the system overall performance.

The tests were performed in a notebook with a core I7-4500U (1.80GHz) and 8GB of RAM. To generate a single hash using that configuration of PBKDF2 the code was taking 100 milliseconds.

The login and password fields are not encrypted with the user's key since we use them to perform the authentication process, the password is stored using PBKDF2.

The hash Equation (2) was developed using the PBKDF2 implementation found in Stanford JavaScript Crypto Library (SJCL) [36]. The number of iterations used for the PBKDF2 loop was also 20,483.

As presented in Section V-A, the process of data dissemination between IdP and SP had to be extended due to the use of encrypted PII; otherwise, the attributes would be sent to the SP encrypted.

To apply the proposed changes into the dissemination process, it was necessary to create a set of classes that temporarily store PIIs that are sent to the SP at the IdP.

It was also necessary to modify a file known as "approve.jsp", that is responsible for rendering the dissemination page. In one hand, we developed the Equation (3) using the Cryptico library [35] to open the PIIs selected to be disseminated. On the other hand, we changed its layout to improve user awareness and control of attributes, allowing them to select which attributes of each scope is disclosed. Figure 7 presents the dissemination page that we created applying the proposal presented in Section V-A.

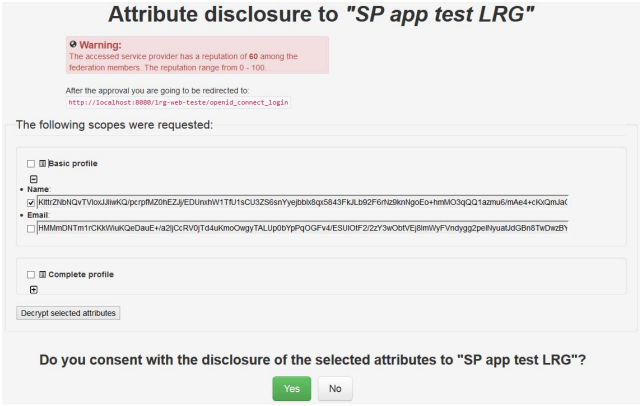

Figure 7. Dissemination process extended

As shown in Figure 7 users attributes were persisted into the IdP encrypted. Therefore, during the dissemination process users have to open the attributes they want to disseminate, bear in mind that not all attributes may be encrypted, only the sensitive ones. In Figure 7 all of the

attributes were encrypted, but that does not need to happen, one can decide that user's name are no sensitive and store them in clear text.

When the user opens PIIs that have to be disseminated to the SP, we use the Equation (4) to encrypt them before it is sent to the IdP to be delivered to the SP. The Equation (4) was developed using the SJCL library; to generate the SK we used the library SJCL that uses a pseudo-random number generator that uses an implementation of the Ferguson and Schneier generator Fortuna [37].

In order to apply the proposal presented in Section V-B we added JavaScripts events in "approve.jsp" file. We also had to create and configure a filter that intercepts all the requests that use the file "approve.jsp". That filter is responsible for loading and setting in the request object all the metadata needed to guide users during the disclosure process, the SP reputation loaded from the reputation system and the scopes from the database and make it available to the "approve.jsp" file that renders the disclosure interface.

VII. Conclusion

Users as owners of data that are stored in IdPs should have full control of those attributes that are stored; as presented in Section II-B there are laws that aim to provide such protection to users. The presented design goes towards the fulfillment of those laws by using encrypted PIIs data that are stored in IdPs with a key that only the user has.

This design is an evolution of our previous one [6], in which we presented an initial proposal to protect user PIIs that are stored in IdPs. In our initial work, we proposed the use of policies that are inserted by users in IdPs to assist and automate the dissemination process. However, the development and maintenance of data release policies are complex and difficult tasks to be performed by users. Instead of transferring this responsibility to users, we believe that dissemination policies should be developed and maintained by IdPs, in order to provide a better user experience. Thus, the usage is transparent to users, and they receive feedback about their choices in the dissemination interface.

Related works summarized in Table I may lack: (i) awareness to users about PIIs that are disseminated from IdPs to SPs; (ii) support to users during the dissemination process; or (iii) use of cryptography to protect user data. The presented design considers the use of policies to provide guidance to users during the process of PIIs dissemination. The design also considered the use of encrypted PIIs to prevent its use without user awareness and consent due to actions of curious/malicious entities. Additionally, the proposed design has been added into MITREid Connect as a proof of concept.

The proof of concept developed implements the countermeasures mapped using the aforementioned security methodologies. However, we foresee the need to investigate methods to adapt our proposal to real federated environments, such as the conversion of current open attributes that are used to encrypted ones. Also, the extra layer of encryption can impact the performance and usability of the proposed design; thus, requiring further investigation.

We also envision the following future works: (i) investigate the use of semantics into our proposals to facilitate the adaptation of systems already developed and to decouple identity management models and protocols from the technology aspect; (ii) define an agreement protocol between IdPs and SPs; (iii) add policies that are disseminated from IdPs to SPs with PII data, in order to manage PIIs usage by SPs; (iv) evaluate user responses when facing an IMS using the presented design. (v) look for means to increase the usability of the system, users are not normally familiar with the use of cryptography.

References

[1] S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "Managing and accessing data in the cloud: Privacy risks and approaches," in *7th Risk and Security of Internet and Systems (CRiSIS)*. IEEE, 2012, pp. 1–9.

[2] D. Kocieniewski, "Adobe announces security breach," *The New York Times*, Outubro 2013, retrieved: February, 2014. [Online]. Available: http://www.nytimes.com/2013/10/04/technology/adobe-announces-security-breach.html

[3] C. Sang-Hun, "Theft of data fuels worries in south korea," *The New York Times*, January 2014, retrieved: February, 2014. [Online]. Available: http://www.nytimes.com/2014/01/21/business/international/theft-of-data-fuels-worries-in-south-korea.html

[4] B. Barrett, "Hack brief: Hackers may have breached oracles cash register system," *Wired*, September 2016, retrieved: September, 2016. [Online]. Available: https://www.wired.com/2016/08/hack-brief-hackers-may-breached-oracles-cash-register-system/

[5] T. Orawiwattanakul, K. Yamaji, M. Nakamura, T. Kataoka, and N. Sonehara, "User-controlled privacy protection with attribute-filter mechanism for a federated sso environment using shibboleth," in *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*. IEEE, 2010, pp. 243–249.

[6] R. Weingärtner and C. M. Westphall, "Enhancing privacy on identity providers," *The Eighth International Conference on Emerging Security Information, Systems and Technologies – SECURWARE*, 2014.

[7] G. Alpár, J.-H. Hoepman, and J. Siljee, "The identity crisis. security, privacy and usability issues in identity management," *arXiv preprint arXiv:1101.0427*, 2011.

[8] OpenID, "Welcome to openid connect," 2014, retrieved: March, 2015. [Online]. Available: http://openid.net/connect/

[9] MIT, "Mitreid connect," 2014, retrieved: October, 2014. [Online]. Available: https://github.com/mitreid-connect/OpenID-Connect-Java-Spring-Server

[10] Shibboleth, "Shibboleth," 2015, retrieved: March, 2015. [Online]. Available: https://shibboleth.net/about/

[11] FrogeRock, "Give customers access to more," 2015, retrieved: August, 2015. [Online]. Available: https://www.forgerock.com/products/access-management/

[12] I. 29100:2011, "Information technology – security techniques – privacy framework," 2011, Retrieved: February, 2016. [Online]. Available: ISO/IEC 29100:2011

[13] H. Lee, I. Jeun, and H. Jung, "Criteria for evaluating the privacy protection level of identity management services," in *Third Emerging Security Information, Systems and Technologies.* IEEE, 2009, pp. 155–160.

[14] D. W. Chadwick, "Federated identity management," in *Foundations of Security Analysis and Design V*. Springer, 2009, pp. 96–120.

[15] C. Landwehr, D. Boneh, J. C. Mitchell, S. M. Bellovin, S. Landau, and M. E. Lesk, "Privacy and cybersecurity: The next 100 years," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1659–1673, 2012.

[16] H. R. Council, "The promotion, protection and enjoyment of human rights on the internet (a/hrc/20/l.13)," 2012, retrieved: February, 2014. [Online]. Available: http://ap.ohchr.org/documents/alldocs.aspx?doc_id=20280

[17] E. Directive, "Directive 95/46/ec of the european parliament and of the council," *Official Journal of the EC*, vol. 23, no. 6, 1995, retrieved: February, 2014. [Online]. Available: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML

[18] U. S. Congress, "Health insurance portability and accountability act of 1996," 1996, retrieved: February, 2014. [Online]. Available: http://www.gpo.gov/fdsys/pkg/PLAW-104publ191/html/PLAW-104publ191.htm

[19] C. Civil, "Lei n°12.965, de 23 abril de 2014," 2014, retrieved: July, 2014. [Online]. Available: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l12965.htm

[20] SWITCH, "uapprove - user consent module for shibboleth identity providers," 2012, Retrieved: July, 2014. [Online]. Available: https://www.switch.ch/aai/support/tools/uApprove.html

[21] R. Sánchez, F. Almenares, P. Arias, D. Díaz-Sánchez, and A. Marín, "Enhancing privacy and dynamic federation in idm for consumer cloud computing," *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 1, pp. 95–103, 2012.

[22] S. Betgé-Brezetz, G.-B. Kamga, M. Ghorbel, and M.-P. Dupont, "Privacy control in the cloud based on multilevel policy enforcement," in *IEEE 1st, Cloud Networking (CLOUDNET)*. IEEE, 2012, pp. 167–169.

[23] D. W. Chadwick and K. Fatema, "A privacy preserving authorisation system for the cloud," *Journal of Computer and System Sciences*, vol. 78, no. 5, pp. 1359–1373, 2012.

[24] S. Betgé-Brezetz, G.-B. Kamga, M.-P. Dupont, and A. Guesmi, "End-to-end privacy policy enforcement in cloud infrastructure," in *IEEE 2nd, Cloud Networking (CloudNet)*. IEEE, 2013, pp. 25–32.

[25] A. Sumaray and S. K. Makki, "A comparison of data serialization formats for optimal efficiency on a mobile platform," in *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, ser. ICUIMC '12. New York, NY, USA: ACM, 2012, pp. 48:1–48:6. [Online]. Available: http://doi.acm.org/10.1145/2184751.2184810

[26] Q. Zhang, Y. Qi, J. Zhao, D. Hou, T. Zhao, and L. Liu, "A study on context-aware privacy protection for personal information," in *16th ICCCN, Computer Communications and Networks.* IEEE, 2007, pp. 1351–1358.

[27] M. Hansen, A. Schwartz, and A. Cooper, "Privacy and identity management," *Security & Privacy, IEEE*, vol. 6, no. 2, pp. 38–45, 2008.

[28] T. Mather, S. Kumaraswamy, and S. Latif, *Cloud security and privacy: an enterprise perspective on risks and compliance*. " O'Reilly Media, Inc.", 2009.

[29] W. Han-zhang and H. Liu-sheng, "An improved trusted cloud computing platform model based on daa and privacy ca scheme," in *Computer Application and System Modeling (ICCASM), International Conference on*, vol. 13, Oct 2010, pp. V13–33–V13–39.

[30] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," *Requirements Engineering*, vol. 16, no. 1, pp. 3–32, 2011.

[31] S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover security design flaws using the stride approach," *URL:http://msdn.microsoft.com/en-us/magazine/cc163519.aspx*, November 2006, retrieved: January, 2015.

[32] A. S. Foundation, "Apache tomcat," 2015, retrieved: August, 2015. [Online]. Available: http://tomcat.apache.org/

[33] W. M. A. Robert Windrem and K. Dilanian, "Russians hacked two u.s. voter databases, officials say," *Wired*, September 2016, retrieved: August, 2016. [Online]. Available: http://www.nbcnews.com/news/us-news/russians-hacked-two-u-s-voter-databases-say-officials-n639551

[34] E. Birrell and F. B. Schneider, "Federated identity management systems: A privacy-based characterization," *IEEE security & privacy*, vol. 11, no. 5, pp. 36–48, 2013.

[35] R. Terrell, "Cryptico: An easy-to-use encryption system utilizing rsa and aes for javascript." 2012, retrieved: March, 2015. [Online]. Available: http://wwwtyro.github.io/cryptico/

[36] M. H. Emily Stark and D. Boneh, "Stanford javascript crypto library," 2009, retrieved: March, 2015. [Online]. Available: http://bitwiseshiftleft.github.io/sjcl/

[37] N. Ferguson and B. Schneier, *Practical cryptography*. Wiley New York, 2003, vol. 141.