Establishing and Protecting Digital Identity in Federation Systems

Abhilasha Bhargav-Spantzel Anna C. Squicciarini Elisa Bertino
CERIAS and Department of Computer Science, Purdue University
(bhargav,squiccia,bertino)@cs.purdue.edu

Abstract

We develop solutions for the security and privacy of user identity information in a federation. By federation we mean a group of organizations or service providers which have built trust among each other and enable sharing of user identity information amongst themselves. Our solution supports a step by step approach according to which an individual can first establish a digital identity followed by a secure and protected use of such identity. We first introduce a flexible approach to establish a single sign-on (SSO) ID in a federation. Then we show how a user can leverage this SSO ID to establish certified and uncertified user identity attributes without the dependence on PKI for user authentication. This makes the process more usable and enhances privacy. The major contribution of this paper is a novel solution for protection against identity theft of these identity attributes. Our approach is based on the use of zero-knowledge proof protocols and distributed hash tables. Revocation mechanisms of the identity attributes are also developed. We illustrate how current revocation techniques can benefit from the underlying federation framework and the use of distributed hash tables. Finally, we formally prove correctness and provide complexity results for our protocols. The complexity results show that our approach is efficient. In the paper we also show that the protocol is robust enough even in the case of semi-trusted "honest-yet curious" service providers, thus preventing against insider threat. We believe that the approach represents a precursor to new and innovative cryptographic techniques which can provide solutions for the security and privacy problems in federated identity management.

Categories and Subject Descriptors: D.4.6 [Operating Systems]: Security and Protection-Cryptographic Controls, K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms: Security, access control.

Keywords: Identity management, single sign-on, federation, identity theft, zero knowledge proof, distributed hash tables, revocation.

I. Introduction

Digital identity corresponds to the electronic information associated with an individual in a particular identity system. Identity systems are used by online service providers (SP) to authenticate and authorize users to services protected by access control policies. With the advent of distributed computing models such as web services, there are increased inter-dependencies among such SP's. As a result, the current trend [24], [25] is to focus on inter-organization and inter-dependent management of identity information [35] rather than identity management solutions for internal use. This is referred to as *federated identity management*. Federated identity is a distributed computing construct that recognizes the fact that individuals move between corporate boundaries at an increasingly frequent rate. Practical applications of federated identities are represented by large multinational companies which have to manage several heterogeneous systems at the same time [35], or by good computing systems in which scientists need to access a large number of machines at different institutions in order to perform computationally intensive tasks. An effort in this sense is represented by the notion of *Single Sign-On (SSO)* [40], which enables a user to login to multiple organizations or SP's by using the same username and password. This approach increases usability by reducing the number of passwords that need to be managed.However, there are several important security considerations, as SSO system may introduce a single point-of-failure. Therefore techniques like strong authentication should be implemented for a secure SSO system.

Emerging standards [24], [25] extend the notion of federated identity to other user information referred to as *identity attributes*. The main goal of such extensions is to enable interoperability and link together redundant user identities maintained by different SP's. An important requirement in this context is that the federation environment should enable SP's to exchange user data in a secure and trustworthy manner while also enforcing the original privacy preferences of users. Current federation solutions are built on top of SAML[1] specification which depends on PKI with additional trust relationships for its security. As such, federations have to rely on PKI for exchanging data among SP's, and between users and SP's authentication systems [35]. However, PKI has experienced numerous implementation problems because of its technical complexities. It is also oriented towards strong identity granted through Registration and Certification Authorities, which is not always suitable for user privacy. Hence, the assumption of relying on PKI for all types of interaction in the federation is not realistic. We thus need articulated identity solutions supporting multiple complementary options for digital identity.

A serious concern related with identity management, whatever solution is chosen, is the risk of identity

[1] Security Assertion Markup Language (SAML)

theft. Despite the guidelines that have been provided on how to protect against identity theft [28], not many identity theft protection solutions have been proposed so far. Sensitive information in the Internet is currently hard to track and also consistent usage of the proposed solutions is extremely hard to achieve. We believe however that in a federation environment it is possible to develop protocols able to achieve identity theft protection. As noted above, the security and privacy of the user identity information, both certified and uncertified, are of utmost importance today. Security prevents theft and impersonation when the identity attributes are used and privacy protects against the disclosure of identity when the user has the right or expectation of anonymity [27].

In this paper we propose a flexible approach to assign unique identifiers to users within a federation employing SSO ID's with strong guarantees against identity theft. To assure user privacy, our protocols do not rely on PKI for user authentication, so that one can easily use uncertified attributes and be eligible for services with low clearance. For cases in which certificates are required we show how SP's can leverage the SSO ID to issue certificates to users. We also show how we can easily employ PKI protocols if a PKI infrastructure is available.

The core of our federated approach for identity management is a set of cryptographic protocols specifically designed to protect user attributes against identity theft. The key idea is to associate the different kinds of sensitive information of a user with each other and with the user's SSO ID. In such way, any of such sensitive information is not acceptable without one or more of the other associated identifying information. We refer to a set of such sensitive information as *attributes Secured from Identity Theft (SIT attributes for brevity)*. Under our approach, SIT attributes are protected : if a user wants to use any of his/her SIT attributes, he/she has to give along one or more other SIT attributes as proofs of identity. We show how we can preserve user privacy without jeopardizing security with the help of cryptographic techniques like zero knowledge proofs [20], [41] and distributed hash tables [30]. The use of zero knowledge protocols makes it possible to hide user values of the proofs of identity even to entities like SP registrars[2]. To the best of our knowledge this is the first time a cryptographic solution to the problem of identity theft has been proposed in the context of federated digital identity management. The correctness of SIT attributes also significantly depends on the revocation mechanisms employed. We provide revocation techniques for SIT attributes and illustrate how current revocation techniques [22], [23], [33] can benefit from the underlying federation framework and the use of distributed hash tables.

[2] If a user trusts an SP to store his/her hidden SIT attributes then that SP is the registrar for that user. More details on this are given in Section IV.

A federation environment inherently protects user attributes more than an open environment. However it is usually assumed that all entities in the federation are completely trusted. In our solution we show how the protocol is robust enough even in the case in which semi-trusted “honest-yet curious” SP’s are in place. In the paper we also formally prove correctness of our protocols and provide complexity results; these results show that our approach is very efficient.

The remainder of the paper is organized as follows. In the next section we introduce preliminary concepts and definitions concerning digital identity in a federation. In particular, in Section II-B we illustrate the approach we adopt to establish digital identity in a federation for both servers and users. In Section III we show how certificates are issued and used in the federation. Section IV gives detailed description of our solution to the problem of identity theft with the help of a running example. In particular Section IV-A gives the basic registration protocol for establishing SIT attributes. This is followed by protocols for SIT attributes usage in Section IV-B and duplicate registration detection in Section IV-C. In Section V we present the revocation mechanism for the different types of SIT attributes. In Section VI we give the formal analysis for the correctness of the protocols and the complexity analysis. In Sections VII and VIII we discuss related work and we outline some conclusions, respectively.

II. Digital Identity in Federations

In this section we present preliminary concepts related with identity. We first briefly review the notion of identity and possible identifying techniques. Then, we present the identity system we have devised, focusing on the approaches we adopt for identifying both users and SP’s.

A. Preliminary Concepts and Definitions

A federation is a group of organizations which trust certain kinds of information from any member of the group to be valid. In this paper we consider federations involving two types of entities: SP’s and users. A SP is an entity providing one or more services to users within the federation. Services are protected by a set of rules defining the requirements users have to satisfy in order to use the service. Often such requirements are expressed as conditions against properties of users. Such properties are usually encoded by means of attributes or credentials.

To interact with the federation, users need to be identified. Identification is the process of mapping claimed or observed attributes of an individual to his/her associated *identifier*. Identifiers can be either encoded using attributes or certificates, or they might be user’s knowledge (i.e., passwords, etc). Identifiers are assigned to users by an *identity system*. Identifiers can be classified into weak and strong identifiers. A strong identifier uniquely identifies an individual in a population[3], while a weak identifier can be applied to many individuals in a population. Whether an identifier is strong or weak depends upon the size of the population and the uniqueness of the identifying attribute. Multiple weak identifiers may lead to a unique identification [46]. Examples of strong identifiers are a user's passport or social security number. Weak identifiers are attributes like age and gender. The types of possible identifiers and their organization are summarized in Figure 1. In the remainder of the paper when mentioning identifiers we will always refer to strong identifiers, unless explicitly stated otherwise.

An identity system should satisfy some requirements related with identification and authentication of the identified users. Identification ensures *accountability* of the users and the ultimate goal is to *authorize* users to obtain required services and/or data by SP's. By *authentication* we mean the process of establishing confidence in the truth of some claim. *Authorization* is the process of ensuring that the policies specifying who may execute which actions on which resources are followed. Authorization decisions do not require the unique identity of the requester to enforce policies. Finally, *accountability* is the ability to associate a consequence with a past action of an individual [36]. It is required that the individual can be linked to action or event for which he/she is to be held accountable. Unlike authorization, accountability requires the ability to uniquely identify the individual.

One of the most common approaches for authenticating the website of an enterprise in today's world is the use of public keys. Trusted commercial entities like Thwate and Verisign, certify that a given public key belongs to that enterprise. Following this practice, we assume that the SP's use the public key infrastructure (PKI). Public keys are thus used to identify and authenticate any SP throughout the federation. Although this approach is adequate for SP's, it is not however suitable for users. Indeed, the management of digital identity by means of user certificates has proven to be very difficult [44]. The two principles identified in [2] namely the "least revealing means" and "most convenient" help in understanding the reasons for such failure. Conforming to them one might selectively reveal elements of his/her identity. First, the *"least revealing means" principle* implies that the minimal identity information should be provided by the user to complete a particular transaction. Second, the *"most convenient means" principle* implies that a user would selectively reveal the combination of identifying information that are most convenient. Here the information should not exceed an upper bound of the amount of information the user is willing to reveal. The use of PKI does not satisfy the *least revealing means* principle because it associates a unique identifier with the user. This could possibly be used as an instrument of surveillance on

[3] This is with respect to the domain in which the user is being identified.

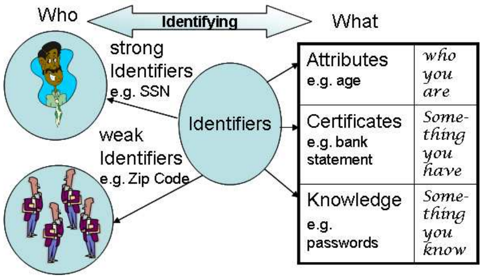

Fig. 1. Classification of User Identifiers

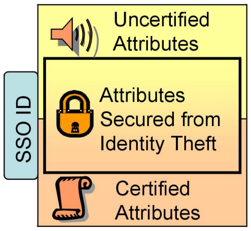

Fig. 2. Attribute Types

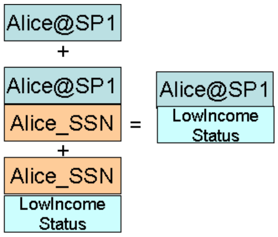

Fig. 3. Credential Ownership Example

the user's online activity hence compromising his/her privacy. Furthermore PKI suffers from compatibility issues which makes it difficult for the user to first establish the different keys and then use it for the various applications. Therefore PKI does not satisfy the *most convenient means* principle as well. As a result of the above shortcomings PKI is not adopted for user identification in the paper.

B. Establishing Unique Identifiers

In this section we focus on the approach adopted to uniquely identify entities in a federation, distinguishing among SP's and users.

Service Providers

SP's within a federation are identified uniquely by their public keys. We denote the two keys belonging to each service provider as K_SP_{Pub} and K_SP_{Priv}, denoting public and private key, respectively. SP's are also responsible for verifying user attributes and issuing digital certificates to certify them. Therefore, each SP also has an additional public key shared with all the providers in the federation, which is used to verify user's certified attributes issued by *any* SP. The shared public key can be generated by using group key algorithms [15] and can be used to identify any SP belonging to the federation. We denote this public key as K_{FED} and the corresponding private key as $K\_SP_{FedPriv}$. Note that, according to the protocols in [15] each SP has a different $K\_SP_{FedPriv}$ associated with the public key K_{FED}.

Users

A user digital identity is basically a set of his/her identifiers. In our work, we employ single sign-on (SSO) ID's to uniquely identify users within a federation. Users affiliated[4] with a SP

[4] By affiliated users we mean users who have repeated interactions with one or more SP's in a federation and are interested in a continuous relationship with the federation as opposed to external users who have only random interactions and are not interested in establishing a relationship with the federation.

are identified by their name and the SP name, separated by symbol $\$$. That is, if Alice is a user affiliated with SP1, her SSO ID would be $Alice\$SP1$. Users receive their ID's when joining the federation. We assume all sharable attributes of registered users to be certified and stored at the member SP they are affiliated with. External users can also join the federation by establishing their SSO user name and password with any SP within the federation. For example, if some user Bob wants to establish a SSO ID in the federation, Bob sends a request to $SP1$ with any desired user name Bob. If this user name does not already exist in $SP1$, $SP1$ registers Bob giving him the SSO ID $Bob@SP1$. Note that other user naming mechanisms could be used here. The essential property is that a user SSO ID be unique within the federation.

Once the user has successfully established a SSO ID in the federation, he/she can then establish different types of digital attributes. The SP services that a user can be eligible for depend on the service policy enforced by the SP and the corresponding attributes the user has. In this paper we consider three types of attributes (see Figure 2): 1) uncertified attributes, corresponding to voluntary information given by user; 2) certified attributes, corresponding to attributes that have been verified and issued as signed digital certificates by trusted SP's or CA's; and 3) attributes secured from identity theft (SIT attributes, for brevity) corresponding to identity attributes that are relevant for the user identification and thus need to be secured by our protection mechanism. Using his/her SSO ID the user can log on to different SP's and access the provided services. Here, different scenarios may arise. In case the user does not have any useful certified attributes, he/she can access services for which only voluntary user information needs to be provided.

For services requiring higher clearance and thus requiring certified information from the user, the user has to be issued the required certificates from either the SP's or third party trusted authorities. The third and most interesting scenario is when the user requires access to a given service with protection against identity theft for the user identity attributes that are to be supplied to get access to the service according the the service policies. These attributes can be both certified or uncertified. Protection of the identity attributes against identity theft is of course in the best interest for any user. However, also SP's may want to offer reliable and secure services and thus interested in requiring that user attributes be protected against theft. We elaborate in each of the above cases in the rest of the paper.

III. Usage of Certificates

For a user to be authorized for a service, the SP may require some certified identity information. Certified information is encoded as certificates issued by SP's within the federation or by external CA's.

Upon federation setting, it is agreed that the SP's will follow an acceptable well defined procedure for the verification and certification of different attributes. These certificates will be considered reliable within the federation.

A. Certificate Provisioning

Certificate provisioning refers to the process of obtaining some certified attributes. In order to obtain certificates usable in the federation the user has to assure that his/her uncertified attributes and claims are verified by any trusted SP or CA. We employ two basic approaches for verification. The first approach addresses the case in which a user that does not possess any initial digital certificate. In such case it is inevitable that the user has to go to a physical location to register any strong identifier. If for example Alice, who does not have any digital certificate, needs a certificate asserting that *Alice_SSN* is her social security number (SSN), she has to show it to an authorized personnel in a physical office at a SP (say SP1). As a result, she gets a signed digital identity which asserts that *Alice_SSN* belongs to user-id $Alice@SP1$.

The second approach is used when a user either already has some digital certificates, or the claimed information can be verified by accessing some reliable online databases. Additional certificates can be issued based on this information. We assume that certificate provisioning policies are in place at the SP. SP's can issue certificates to users depending on user properties and the certificate provisioning policies. SP can also issue a special type of certificate which attests the ownership of other certificates. These certificates certifying *credential ownership* can actually help associate user attributes from different certificates. An example to illustrate this is as follows:

Example 1 *Figure 3 demonstrates an example for the issuance of certificate that denotes credential ownership. As shown, user Alice has two certificates. The first certificate is issued by a SP and states that* $Alice@SP1$ *has SSN Alice_SSN. The second certificate is from a trusted CA and states that Alice_SSN has a Low_Income _Status, (*LIS*) for brevity. Alice wants to get a certificate stating that she (represented by her SSO identifier) has a* LIS *to be used within the federation without revealing her SSN. Alice can obtain such certificate by submitting a trusted SP the* LIS *certificate and the certificate associating her SSN with her SSO ID. In return she obtains the final certificate associating* $Alice@SP1$ *with* LIS*. Here, the actual disclosure of the SSN is not required, but just needs to be the same for the two certificates.* LIS *is signed by the private group key* K*-*$SP_{FedPriv}$ *of the issuing trusted SP and can be verified by any SP in the federation.*

An additional example showing the certificate issuance based on certificate provisioning policies is as follows:

Example 2 *When a user requires the issuance of new certificates, he/she must prove possession of pre-requisite certificates according to the certificate provisioning policies of SP's. If Alice@SP1 has the certificate associating the SSO ID with her SSN, she may be eligible to obtain a* Trusted-User *certificate. In this case the certificate provisioning policy the service provider may require Alice to be uniquely identified to be accountable and hence get the Trusted-User certificate.*

As shown by the example above, when a user requires the issuance of new certificates, he/she must prove possession of pre-requisite certificates according to the certificate provisioning policies of SP's. The user should also satisfy the federation requirements concerning the provisioning of certificates stating a claim for users. Note that in the example unique identification is required for accountability purposes. However, this condition is not always needed for accountability if pseudonymous systems are employed [11], [12], [13], [14].

B. Sharing User Attributes

In [8] we have shown how a user can negotiate with SP's to submit the appropriate certificates and attributes in order to obtain the requested service. The key idea is that if the user has agreed to share this information within the federation, the SP's would negotiate these sharable user attributes amongst themselves. Such an approach enhances user convenience and system usability that the user does not have.Attribute sharing can be achieved efficiently and with privacy guarantees. By using trust negotiation techniques [7] only minimal information about users is required to satisfy the requesting SP's service policy. If not, the privacy of the attributes may be vulnerable as they would reside in multiple locations within a federation of which some locations might not be trusted by the user. The use of tickets namely *trust tickets* and *session tickets* has been shown to be critical in order to determine the user's past activities and related information in the federation.

If the user does not want to share his sensitive identifiers, he/she can directly negotiate with the SP to provide this information when required. The above approach can be used by our identity system for privacy preserving sharing of user identity attributes. We can also use the standard attribute sharing protocols as given in [24], [25].

IV. Protection from Identity Theft

Identity theft occurs when a malicious person uses an honest user's personal information such as the users name, Social Security number (SSN), credit card number (CCN) or other identifying information, without his/her permission. In this section we offer one solution to prevent identity theft in a federation. As defined in the introduction we refer to the protected sensitive information as attributes secured from identity theft (SIT attributes for brevity). Our approach is similar to real world situations where a user is asked for different kinds of sensitive personal information to be assured that the user is really who he/she claims to be. However, in online transactions with different SP's it is desirable that one be able to prove the possession of a sensitive information without the actual revelation of this information in clear.

Our security model consists of two main entities in the federation, that is, users and SP's. SP's can also act as *registrars* for certain users. If a user trusts an SP to store his/her SIT attributes, then that SP is the registrar for that user. A key feature of our approach is that we avoid the use of a centralized registrar entity, thus being consistent with the truly distributed nature of protocols in the federation. Registrars are assumed to be semi-honest[5] for the user attributes they keep track of. A user can register his/her SIT attributes with *any* SP in the federation engaging a bootstrapping procedure. Once the registration is completed, a set of SIT attributes are associated with the user SSO ID and with each other. These attributes are used together with ordinary data to protect from identity theft. Here, and throughout the paper, by protection against identity theft we mean the inability to use a SIT attribute without the proof of additional identity information. To protect against identity theft, it is important that an adversary be prevented from registering as its own SIT attributes of other users; therefore our security model includes mechanisms to detect duplicates within a federation. We provide a detailed presentation of the techniques and algorithms above introduced in the following sections.

For the purposes of clarity we first introduce a running example which is used in the following sections.

Example 3 *User Alice has established an ID in the federation namely $Alice@SP1$. She intends to use her CCN and wants to protect it against identity theft. To do this she registers her SSN and CCN with the SP SP_{reg}. Now, within the federation her CCN is not valid unless she provides information regarding her SSN as registered earlier. So when another SP providing a service, say SP_{prov}, asks for her CCN, first her SSN information has to be validated. Following that the CCN information can be used with SP_{prov} successfully. Even if Alice uses her sensitive SSN as a proof of identity she still does not want to reveal*

[5] According to the accepted definition of semi-honest entities, we assume registrars will follow the protocol but may also want to learn more information than they are supposed to.

this information to the SP's in clear.

A. Bootstrapping: Registration Procedure

Since a SP is not considered completely trustworthy, the values of the sensitive attributes are not to be released in clear. The main goal of registration is thus to store unique and hidden sensitive SIT attributes to such semi-honest SP's. In the next subsections we describe the two alternative registration procedures we have devised.

1) Physical Registration: This type of registration requires the user to go to a SP in order to register his/her sensitive identifiers and attributes in person. Following from Example 3, Alice wants to register her SSN and CCN with SP SP_{reg}. An authorized officer of SP_{reg} verifies the actual $a = SSN$ and $b = CCN$ with the physical cards or certified papers. Then he/she lets Alice enter these values in an offline dual screen computer (or some special purpose device) such that the officer monitors with the help of the second screen that Alice enters the correct values. Such computer calculates $g^{-a} mod\ p$ with tag SSN_{tag} and $g^{-b} mod\ p$ with tag CCN_{tag}. Given the calculated value it is not computationally feasible for an attacker to get the secret values assuming the intractability of Discrete Log Problem (DLP). These values are stored with the user at SP_{reg} with the corresponding user id, that ism $Alice@SP1$. Here we assume that the officer is trusted and does not keep a copy of the sensitive information. The device itself is assumed to be trusted and tamperproof. We do not further elaborate on this aspect since it is outside the scope of this paper.

Physical registration is the strongest and sometimes the most reliable form of identification. However, referring to our principles introduced in Section II-A, even though this method reveals a minimal amount of information, it is not always the most convenient procedure. We therefore look into the second kind of registration which is executed through online message exchange.

2) Online Registration: Online registration of sensitive attributes is challenging in the absence of user public keys or any electronic certified information. To achieve the goal of privacy, we require that sensitive information of the user are *never* given in clear to even the SP storing this information. Of course this requirement adds a level of complexity to the whole registration procedure: the SP cannot guarantee that the information registered is correct, but it can guarantee that the user knows the secret information whose exponentiated value is stored with it. To reach this last goal the SP and the user engage in a zero knowledge proof of knowledge (ZKPOK) as given shortly. To understand the former concern consider three cases following from Example 3. The first case corresponds to the ideal situation, that is, Alice is honest and submits correct values of the sensitive attributes. The second case is when Alice submits a random value instead of the SSN and tags it with SSN_{tag}. If no one other than Alice knows the random correct value, this works perfectly fine. This is because in this case the proof of identity is not *something you have*, it is *something you know* (Please refer to Figure 1). However if the actual value of the SSN is required to be given in clear and has to be verified by an external third party, then Alice's transaction will fail. The final case is when a malicious user, say Carl, tries to register Alice's SSN. Now if Alice has already registered her SSN, the attempt by Carl would be detected by the federation as explained in a Section IV-C. If Alice has not registered this attribute and tries to register it later an alarm would be raised. The alarm would then trigger an auditing procedure to determine which user has already registered the SSN of Alice and a subsequent recovery procedure that will undo the registration made by Carl.   Preventing a malicious user to register using any other identity when the actual user has not registered is a major issue. Several approaches are possible to avoid this problem. One possibility is to mandate that at the time of registration the user provides the strong identifiers indicated by the policy of the registrar. Since our approach is based on multi-factor, a minimum number of attributes will be needed to actively participate in the federation. The exact type and numbers of attributes to register is part of the registrar policy and is to be published at the registrar site. For example, a registrar may require that the user submits at least three strong identifiers before he/she is enrolled in the federation. The registrar will have to validate these strong identifiers before the registration is completed. A policy language for expressing enrollment conditions needs to be deployed.

A more effective approach might be that of providing besides face-to-face registration on-line registration based on "digital introduction" of members. The idea is to allow on-line registration for users showing identity assertions issued by other federated members previously registered using physical registration. The rationale is that if the member user is trusted and his/her identity known to the federation he/she can be the grantor of the identity of the new incoming member. Of course, this approach requires the member to be sufficiently trusted. Also, it implies the use of PKI for signing the identity assertions. In case of detected identity theft from the granted member, the grantor can be held accountable of the attack. We plan to pursue the above two mechanisms as future work to support flexible and secure enrollments. An important issue that will be investigated in such context is the support of enrollments characterized by different identity assurance.

The general ZKPOK's [20] is explained as follows. The protocol allows a committer to have a private secret, and prove its possession without releasing it. The committer releases some information called the *commitment*. The protocol has two main properties: *hiding*, the verifier cannot compute the secret from the commitment; and *binding*, the committer cannot change his mind after having committed, but it can later open the commitment to reveal the secret to convince the verifier that this was indeed the original value it was committed to.

We use Schnorr's ZKPOK to commit the supposedly sensitive information as given in Protocol 1. Following from Example 3, if Alice wants to commit her SSN= a then she commits the value $c = g^{-a} mod\ p$ to SP_{reg}. Using the protocol she can prove that she knows the value a corresponding to the commitment. Similarly, she can commit other sensitive values.

Protocol 1 SIT Registration: Schnorr's Zero Knowledge Protocol

```
Require: Federation System Parameters: $p, q, g$ such that $q|p-1$, and $g$ is an order $q$ element in $\mathbb{Z}p^*$. $t$ is a security
    parameter. $User$ has a valid SSO ID $uid$, Service Provider $SP_{reg}$ is a member of the Federation. Time stamps
    $T_i$ can be generated.
Goal: User knows the secret $a$ of the committed value $c$ which is registered with $SP_{reg}$.
  $User \rightarrow SP_{reg} : c = g^{-a} mod\ p,\ uid, T_1$
  $User \rightarrow SP_{reg} : d = g^{r} mod\ p$ {User selects $r$ from $[1..q]$},$uid, T_2$
  $SP_{reg} \rightarrow User : e$ {$SP_{reg}$ selects $e$ from $[1..2^t]$}
  $User \rightarrow SP_{reg} : y = r + ea\ mod\ q,\ uid, T_3$
  $SP_{reg}$ : if $d = g^y * c^e mod\ p$ then return OK
```

At the end of the registration procedure in Example 3 the registrar SP_{reg} has the information as given in Table I. We now describe how the SIT attributes, that are registered through either the bootstrapping or the registration procedure, are used in the federation to protect against identity theft.

Tag	Committed Value	Registration Procedure
SSN_{tag}	g^{-a}	In Person, T_1
CCN_{tag}	g^{-b}	Online, T_2

TABLE I
INFORMATION REGISTERED FOR ALICE AT SP1
(Refer example 1).

B. Using SIT Attributes to Protect Against Identity Theft

The main aim of the protocols we present here is to make the use of SIT attributes possible only under the submission of a subset of additional SIT attributes which have been registered. The exact subset of the additional SIT attributes required is determined based on the user and/or SP's identification policy. Such attributes act as a proofs of identity and enable association of required SIT attributes with the other registered SIT attributes. This gives assurance that the user is in control of his/her sensitive attributes and is therefore honest. The solution we have devised to deal with identity theft consists of two main phases.

The first and key phase is a *ZKPOK and symmetric key exchange* protocol. Here the user proves that he/she knows the actual value of a specified SIT attribute without revealing its value in clear. In addition, at the end of a successful run of this protocol the user and SP share a single symmetric key related to the proof of knowledge of that SIT attribute. With repeated runs of this protocol multiple symmetric keys can be shared. These keys are used to retrieve the required SIT attributes from the final message given by the user to the SP. *Creation of the final message* is the second phase of the solution. Here, the user creates a message encrypted in a nested manner with the symmetric keys generated in the first phase. Next we elaborate on these two phases in detail.

Protocol 2 ZKPOK and single symmetric key exchange

```
Require: Federation System Parameters: $p, q, g$ such that $q|p-1$, and $g$ is an order $q$ element in $\mathbb{Z}p^*$. $t$ is a security
    parameter. $User$ has a valid SSO ID $uid$ and has registered his/her attributes with Service Provider $SP_{reg}$ and
    wants service from $SP_{prov}$. Both the $SP's$ are members of the Federation. Time stamps $T_i$ can be generated.
Goal: $SP_{prov}$ verifies correctly that $User$ knows the value $a$ registered with $SP_{reg}$ to retrieve the key $k$.
1: $User \rightarrow SP_{prov}$ : $msg$ $\{msg = SP_{reg}$ has my ($uid$) commitment for secret for $tag\text{-}of\text{-}a\}$, $T_1$
2: $SP_{prov} \leftrightarrow SP_{reg}$ : $c$ $\{$retrieves $c = g^{-a} mod\ p$ corresponding to $tag\text{-}of\text{-}a$ and $user\text{-}id\}$
3: $User \rightarrow SP_{prov} : d = g^r mod\ p$ $\{$User selects $r$ from $[1..q]\}$, $uid, T_2$
4: $SP_{prov} \rightarrow User : e$ $\{SP_{prov}$ selects $e$ from $[1..2^t]\}$
5: $User \rightarrow SP_{prov}$ : $y = r + ea\ mod\ q$, $y' = r + ea + x\ mod\ q$, $\{x$ is a random such that $k = d(g^x - 1)\}$,
    $uid, T_3$
6: $SP_{prov}$ : verifies $d = g^y * c^e mod\ p$ and evaluates key $\{(g^{y'} * c^e) - d\} mod\ p = k$
```

The ZKPOK and symmetric key sharing is given in Protocol 2. There are three entities involved in this protocol; namely the user, the SP from which the user wants service, denoted as SP_{prov}, and the SP which registered the SIT attributes of the user, denoted as SP_{reg}. In step 1 the user lets SP_{prov} know that SP_{reg} has his/her committed values. Then in step 2 SP_{prov} confirms this claim with SP_{reg} and gets the required commitments corresponding to the SIT attributes as proofs of identity and required SIT attributes. Similar to Schnorr's protocol in steps 3,4 and 5 the user generates a proof depending on the random challenge sent by SP_{prov}. The main difference is in step 5 where the user calculates y' depending on the random symmetric key k. Here x is randomly chosen so that the resulting key is also random. For the standard symmetric ciphers there is a short list of weak keys which are avoided at this step. The weak keys are mainly the keys with predicatable patterns like "0000.." or "010101". The user also generates y to prove it knows the value of the commitment like the original proof in Protocol 1. In step 6, SP_{prov} verifies the claim and retrieves the symmetric key generated by the user in step 5. SP_{prov} can get the key k only if it knows the value of the secret commitment stored in SP_{reg}. This protocol is repeated in order to obtain a symmetric key as a proof of knowledge for each SIT attribute required.

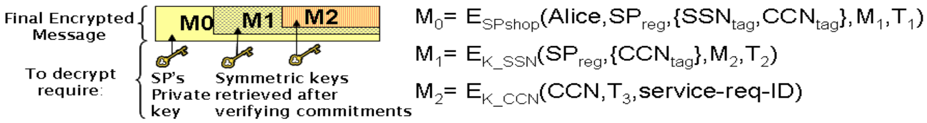
Final Encrypted Message
M0
M1
M2
To decrypt require:
SP's Private key
Symmetric keys retrieved after verifying commitments
M_0= E_{SPshop}(Alice,SP_{reg},{SSN_{tag},CCN_{tag}},M_1,T_1)
M_1= E_{K_SSN}(SP_{reg},{CCN_{tag}},M_2,T_2)
M_2= E_{K_CCN}(CCN,T_3,service-req-ID)

Fig. 4. Final message format from $Alice$ to SP_{prov}. *Refer example 1*

Once the symmetric keys are generated and shared, the user creates the final message. If there are n SIT attributes required as proofs of identity and m SIT attributes required for satisfying the requested service policy, then Protocol 2 is run $N = n+m$ times resulting in N symmetric keys. The final message is encrypted N times in a nested manner. The required information is revealed only in the inner-most encrypted portion. As a clarifying example we show in Figure 4 the format of final message that $Alice$ would give to SP_{prov} in Example 3. Through 2 iterations of Protocol 2, the keys K_{SSN} and K_{CCN} are retrieved. M_0 is the main message sent to SP_{prov} which is encrypted with the SP's public key. By reading this message SP_{prov} knows it has to use K_{SSN} to open one layer of encryption thus retrieving the deciphered version of M_1. It now knows it has to use K_{CCN} to get the final value, that is, CCN. SP_{prov} can also confirm at this point that this corresponds to the value committed to SP_{reg} which it received during the last run of Protocol 2.

C. Identifying Duplicates of SIT attributes

During registration it is very important to verify if the proposed commitments of sensitive attributes are already registered in the federation. Such verification is to prevent a malicious user from registering stolen SIT attribute values with his/her credentials. Therefore to prevent duplicates the responsible SP should check with all other SP's if the proposed commitment of a sensitive attribute is already present. Such a check can be executed incrementally or via a broadcast and it can be very expensive. We therefore propose the use of distributed hash tables (DHT)[30] for this purpose.

A DHT has no central server and partitions a key space among *n* servers. Each distributed server has partial list of where data is stored in the system and the keys are uniformly mapped to the servers according to rules specified by the federation. A *"lookup"* algorithm is required to locate data given the key for that data. There are two main functions for handling the data, namely $put(key, data)$ and $get(key)$.

For our purposes the DHT is used as follows. The unique identifier or the key for the DHT is the commitment $c = g^{-a}$ as given in Protocol 1. The key space therefore has the range $[0..p-1]$ and the data is the tuple $(user\text{-}ID, TAG, Type\text{-}of\text{-}Registration)$. Because a federation is a closed system, there is

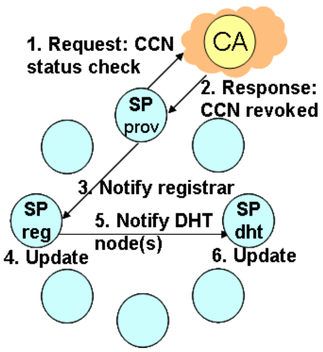

Fig. 5. Steps for certified SIT attribute revocation in a *push mode*.

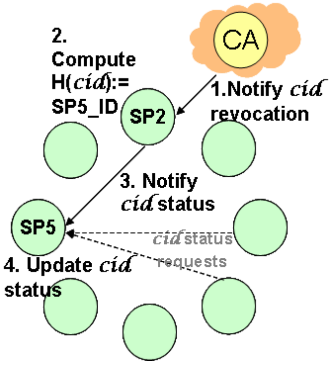

Fig. 6. New DHT for SIT attribute revocation in a *pull mode*.

an inherent trust amongst the SP's; therefore we can use the SP's for storing and retrieving the values [29]. During the formation of the federation a range of key values are given to each SP which will be responsible for storing the given keys and the corresponding data values. This is in addition to the SP with which the user had registered in the first place. When a user wants to register his/her attributes with an SP, that SP executes the $lookup$ algorithm to find if any duplicate is present. If a duplicate is present then an alarm is raised that triggers a procedure that determines the compromise of the sensitive attribute. Otherwise, first the SP registers the users commitments. Following that, depending on the range the committed value belongs to, the appropriate SP is given this key and the corresponding data. This replication adds to the robustness and security of the DHT's. As a result, we claim that by using the DHT's we can prevent duplication of registered sensitive identifiers, which is crucial to ensure effective protection against identity theft.

V. Revocation of SIT attributes

Digital attributes state certain well defined properties about the subjects they refer to. Such attributes may be indefinitely valid, or may be valid for a given time interval. Also, attributes may be revoked if some external events compromising their validity occur. *Revocation* thus refers to the *undo* of the claim associated with an attribute. Events that may cause attributes revocation include [22]:

- Compromising of the owner key: The owner key linked to this certificate has been compromised.
- Compromising of the issuer key: The issuer key used to generate this certificate has been compromised.
- Owner changed affiliation: The identification details of the certificate are no longer valid.
- Obsolescence of the certificate: The certificate is superseded by another certificate.

- Certificate terminated: The certificate has reached the end of its validity period and has not been renewed.

A. Preliminary Notions Concerning Revocation

Most of the work in the area of revocation has focused on the revocation of certified attributes or public key certificates [45]. A widely used standard for defining these digital certificates is the X.509 [23] format. The two most popular schemes which have been proposed to manage X.509 certificate revocations are Certification Revocation Lists (CRL's) [22] and Online Certificate Status Protocol (OCSP) [33]. Both aim at providing risk analysis based on certificate usage and efficient notification about the validity status of the certificate. CRL is essentially a list of certificate serial numbers which have been revoked and are therefore no longer valid. The CRL is always issued by the CA which issued the corresponding certificates. A PKI-enabled application consults this CRL to verify the validity of the certificate prior to its use. Due to its centralized nature, CRL is not scalable and also requires huge bandwidth in order to communicate with all its clients. OCSP supersedes CRL's by providing efficient notification, through the use of a distributed protocol. A typical OCSP defines a request-response protocol between OCSP client and an OCSP responder. The OCSP responder is a trusted entity that informs the requester about the validity information of the certificates. The OCSP responder can contact various backends including CRL's to retrieve the revocation information. One issue with OCSP is that the requester must know which OCSP responder to query. This information is typically specified in the Authority Info Access (AIA) extension of a certificate [33]. One limitation regarding the use of AIA extension is that it might give rise to deployment problems caused by the increased size of the certificate. Moreover one has to ensure that there are no compatibility issues because of the different versions or types of certificates using the AIA extension. The problem of finding OCSP responders can be elegantly solved in a federated environment with the help of DHT's and with an optional use of AIA extension. Another more important limitation of OCSP, and indeed any current revocation mechanism, is that it cannot be used for uncertified attributes. This is because there is no assigned CA which can revoke such attribute. We leverage the federation architecture and follow a policy based approach for uncertified attributes. We assume that the revocation status is essentially provided by either the uncertified attribute's owner, or it is the result of the feedback of other federation entities.

A comprehensive description of the revocation mechanisms for the various types of attributes is given in the next subsection. We show how the underlying collaborative environment of a federation provides opportunities for efficient solutions to the problem of attribute revocation.

B. Revocation of SIT Attributes

The SIT attributes introduced in this paper are useful only if they can be verified and revoked reliably when necessary. Generally speaking, revocation techniques should enable efficient notification to the potential consumer of the revoked attributes and prevent its subsequent usage. *When* and *how* should a SIT attribute be revoked depends on the type of the SIT attribute. As elaborated earlier, there are two types of SIT attributes, namely certified and uncertified (see Figure 2), which require different approaches to revocation. Precisely the adopted approaches are described as follows:

> **Certified SIT Attributes.** Certified SIT attributes should be revoked when the original issuer of the certificate (external CA, or an internal federation SP) disqualifies that certificate. This corresponds to the credential revocation criteria already well investigated [16], [23].
>
> Referring to Example 1, consider the case when SP_{prov} checks for the validity of Alice's CCN with the appropriate external CA and is notified that this CCN certificate is revoked. As a consequence of this notification, revocation steps have to be taken to update the information in the federation as shown in Figure 5. At step 3 SP_{prov} sends a signed revocation message to SP_{reg} with the SSO Id $Alice@SP1$ and the tag CCN_{tag}. Based on this SP_{reg} can retrieve Alice's identity record (see Table I). Now SP_{reg} can either remove the row corresponding to CCN_{tag} or add an additional column to record the status information as *revoked* and finally nullify the 'Committed Value'. In both the cases the *commitment* which is requisite for establishing proof of ownership of the corresponding SIT attribute is removed. Since an SIT attribute cannot be used without a valid proof as shown in Section IV-B, subsequent usage of the revoked SIT attribute is prevented. This is the identity record update corresponding to step 4 in the figure.
>
> In addition to updating the attribute information at the local registrar the DHT node saving the attribute commitment also has to be updated. This is because duplicate should be detected only for valid and unique identifiers. If an identifier has been revoked the federation policy may allow the re-registration of revoked attribute or not. Therefore in step 5 SP_{reg} sends a revocation message to SP_{dht} which is the DHT node saving CCN commitment. Depending on the revocation policy, SP_{dht} can either simply delete this information from its hash table or it can add the revocation information in the value corresponding to the commitment key. The update of the DHT node(s) completes the revocation process.
>
> **Uncertified SIT Attributes.** Uncertified attributes correspond to voluntary claims of individuals which do not have to be signed or verified by any trusted authority. Here the user itself is the

issuer of the SIT attribute. Therefore the trust in the claim is often considered uncritical. However serious security problems, like spam, phishing and pharming [17], [21], [38] attacks, arise from the incorrect usage of the uncertified attributes. It should thus be possible to revoke the usage of uncertified attributes. To the best our knowledge, revocation of uncertified attributes has never been explored.

Determining when uncertified attributes should be revoked is more complex than certified attributes because there is no entity who can assert accurately the validity of such attributes. However we assume that if a number of distinct revocation assertions are received for a certain attribute then the attribute is to be revoked. The number of accumulated assertions should be greater than a certain threshold, determined by the federation security policy. For example consider the case that user Alice subscribes her claimed email address $phony@myemail.com$ to SP_1, SP_2 and SP_3. Eventually due to bounced emails each of the SP concludes that the email is invalid. To revoke such attribute they separately send revocation requests to the designated registrar SP_{reg}. SP_{reg} saves these requests in an additional column of Alice's identity record (Table I). If the federation accepts a threshold of three then when the third such request is received the SP_{reg} revokes this attribute. The revocation steps thenceforth follow steps 4 to 6 of the protocol for the revocation of certified attributes (see Figure 5). Further usage of possibly incorrect uncertified attributes is thus prevented. Similar revocation procedure can be adopted if a strongly authenticated[6] user requests revocation of his/her attribute.

The notification mechanism described above for certified SIT attributes corresponds to a *pull mode* for revocation notification. This is a request reply mechanism where the reply is valid when it is from an authorized CA. In the case of OCSP, revocation information can also be from an authorized OCSP responder [33]. As highlighted earlier one problem is that the requester SP should know which OCSP responder it should contact to get the revocation information. One approach to address this issue is to define a *push mode* revocation notification where the main CA pushes the revocation notification to the federation SP's when a revocation event occurs. Here the SP's themselves play the role of OCSP responders and the revocation information requests can be satisfied within the federation. To support such solution we deploy an additional DHT (referred to as revoke-DHT for clarity) with the SP's as the distributed nodes. The revoke-DHT key in this case is the certificate ID itself. An external CA has to notify any one of the SP's in the federation. As an example in Figure 6 SP_2 is notified about the certificate

[6] Strong authentication is when more than one factors were used to authenticate the user.

identified by certificate ID cid. SP_2 computes the revoke-DHT hash $H(cid)$ to identify the DHT node (SP_5 in this case) where this cid should be stored. Subsequently it sends the revocation information to SP_5. Henceforth any SP which needs revocation information about certificate cid can directly access SP_5 by computing the same hash, thus identifying the DHT node responsible for providing cid's revocation information. This solves the problem of identifying the OCSP responders outside the federation.

Note the hash values can be pre-computed and stored in the AIA[7] extension's `accessLocation` field of the certificate [33]. This parameter essentially stores the location of the OCSP responder. AIA extension configuration is very useful, but it has to be done carefully since improper use of certificate extensions has led to severe deployment problems [26]. Therefore instead of adding this information into the certificate we can leverage the knowledge of the revoke-DHT's hash function to calculate the responder at runtime. This information can also be cached locally in the system. In this way we provide two alternative methods which can be used to implement OCSP for certified SIT attributes.

VI. Analysis and Discussion

We now analyze the security and complexity of the SIT protocols. In particular we assess identity theft protection in the presence of malicious parties and communication costs. Before starting the analysis we present an interesting paradox based on the desired properties for identity theft protection in the federation. The two required properties are as follows:

Property 1: *Identity Hiding.* Given $f(x)$, it is infeasible to compute the value of x.

Property 2: *Duplicate Detection.* Given $f(x)$ and $f(y)$ identify the case when $x = y$.

The first property is required so that the registrar SP, which stores the committed values of the user is not able to compute the actual secret value. If that SP could compute the values, the registration process could be simplified by storing all the values in clear. However, if such SP is compromised so are all the SIT attributes. The second property is required to prevent duplicates of sensitive identifier commitments in the federation. This requirement is needed to prevent a malicious user from registering stolen attributes with his/her identifier. It is interesting to observe that property 2 enables one to launch a brute force dictionary attack. This is a realistic attack for most identifiers. For instance a SSN is composed of 9 digits, therefore it has $10^9 < 2^{30}$ possible SSN's. Listing 2^{30} possible strings for a 30 bit value is not difficult for an adversary with moderate computational resources. It is not obvious how padding or randomization can be added in a useful manner.

[7] Authority Info Access.

To solve this problem we suggest that each strong identifier be appended with weak identifiers associated with it. For example a credit card number can be appended with expiry date and name. This results in a longer length of this committed proof of identity, which in turn raises the bar for the brute force attack. The probability of forgery in Protocols 1 and 2 is $1/2^t$ where t is the security parameter. This parameter is useful to determine how many weak identifiers will need to be appended to a given strong identifier. A more general solution, however, is an open problem which may be solvable using innovative cryptographic techniques and security models. One possible direction would be to investigate elliptic curve cryptography (ECC) based zero-knowledge proofs [5], [32] where the intractability of the original discrete log problem [43] is exactly the same, although ECC typically provides good security with small secrets and little processing.

A. Identity Theft Protection in the Presence of Malicious Parties

We now prove some relevant properties of the SIT attribute registration and the SIT attribute usage protocol. As illustrated in Example 3 a federation is mainly composed by three kinds of entity: the user; the registrar SP_{reg}; and the SP_{prov} which will be providing the service. In our analysis, we assume that the SP_{reg} is semi-honest, unless stated otherwise. The following two theorems prove the correctness and confidentiality properties of the registration protocol. By correctness we mean that an honest user can execute the protocol successfully thus achieving the specified results. By confidentiality we mean privacy preservation of the registered user attributes.

Theorem 1 *Let U be a user and let $Attr$ be the set of attributes U wishes to protect. Protocol 1 ensures registration attributes $Attr$ to be identity-protected, even in the presence of malicious users.*

PROOF. We prove that a malicious user cannot compromise a honest user SIT attribute registration. Two possible cases arise: i) U registers a set of attributes *before* a malicious user tries to re-register a subset of those attributes with his/her actual attributes instead, ii) U registers a set of attributes *after* a malicious user has registered U's stolen attributes.

In case i), after the honest user has successfully registered his/her SIT attributes, a malicious user will not be able to re-register those attributes with his/her own SSO ID. This is ensured by the duplicate detection mechanism given in Section IV-C. Here, we assume that the exact value of the committed sensitive attribute is important for the validity of the attribute. Therefore when the adversary attempts to re-register the commitment will look identical to the one sent by the honest user. The registrar service provider detects this duplicate and hence denies the registration.

In case ii), attributes are obviously SIT attributes only after they are registered. If the user tries to register with any SP in the federation, like in case (i), a duplicate is detected and physical verification is requested by the SP. In this case only the actual attribute owner (user U) can give a valid in-person proof. As a result, once the fraud has been detected, U can re-register the values successfully. The thesis thus holds. □

Theorem 2 *The SIT attribute registration protocol satisfies confidentiality.*

PROOF. We prove that the actual sensitive values of the registered attributes is not revealed even to the registrar SP_{reg}. This result is directly related to the *hiding* property of Schnorr's ZKPOK protocol. The key assumption is that the Discrete Log Problem [31][8] is hard for a polynomially bound adversary. If the length of the committed attribute is more than sixteen bits, then by current standards it is infeasible for an adversary to launch a dictionary attack to guess the value of the committed value or compute the discrete log. The actual secret of the commitment thus remains confidential. □

Corollary 2 Let U be a user and let Attr be the set of attributes registered by U at SP_reg. A registrar SP_reg cannot infer *other* attributes related with $Attr$.

Sketch of Proof. It has been shown in [46] how combining attribute information about a user can help infer his/her other not disclosed attributes. Because the actual values of the attributes remain confidential even to SP_{reg}, we see that it is not possible to infer other attribute information from the set of committed values. The only values given in clear are the $tags$ associated with the given commitments. These $tags$ are required to be generic so that they do not leak information about the corresponding secret attribute. For example instead of having a tag *Purdue-Student-ID* the tag should be *Affiliated-Institution-ID*. Here the latter generic tag does not have the specific identifiers like *Purdue* and *Student*. There is no other information in clear which could leak information; therefore inferring information about the user is hard.

The above result can be strengthened with a small variation to the protocol proposed in Section IV-A. Instead of leaving the attribute tags in clear it is possible to hide them using hashing functions. The actual tag values will thus be known only to the user and never revealed to anyone. Inference in this case is impossible. We do not use such approach in the current version of our protocols because, in most cases, inference carried out by tag analysis is not significant and does not violate privacy of the users.

[8] Given a multiplicative group $(G, *)$, an element g in G having order n and an element y in the subgroup generated by g, we have to find the unique integer x such that $g^x mod\ n = y$. Here x is the discrete logarithm $log_g y$.

Also, an hybrid approach where some of the attributes tags are hidden (typically the most sensitive or less generic ones) is possible.

The next two theorems prove the correctness and confidentiality of the SIT attribute usage protocol. By correctness of SIT attribute usage protocol we mean that the proofs of identity can be used successfully to prove ownership of the attributes required by the SP_{prov}. Only after this proof is executed, does SP_{prov} obtain the required attributes. Correctness of the SIT attribute usage protocol ensures mitigation of identity theft. By confidentiality we mean that the protocol is privacy-preserving with respect to the user attributes such that none of the SP's learn more information than required about the user.

Theorem 3 *Let U be a user and SP_{prov} be the service provider U interacts with. SIT attribute usage protocol is correct if and only if at least one of the interacting parties is semi-honest.*

PROOF. To prove that SIT attribute usage protocol is secure we need consider two cases:
i) U is malicious and SP_{prov} is semi-honest.
ii) SP_{prov} is malicious and U is semi-honest.

Case i). A malicious user cannot provide the commitment corresponding the the different proofs of identity required by the SP_{prov}'s policies. We assume that not all the sensitive SIT attributes are compromised. Due to incorrect commitments the symmetric key exchange in Protocol 2 fails. Referring to the protocol, the malicious user could easily articulate y' using the equation: $\{(g^{y'} * c^e) - d\} mod\, p = k$ used by the SP_{prov} in step 6 to exchange a symmetric key successfully. However, y can only be verified if the user knows the secret using the original ZKPOK. Therefore this protocol is secure against a malicious user.

Case ii). In our context a malicious SP_{prov} is a SP wishing to use SIT attributes of the user *without* verifying the identity proofs. This not possible because of the format of the final message disclosed by the user (see Figure 4). Protocol 2 for key exchange is successful if and only if the committed value is verified correctly with SP_{reg}, for each such attribute. The messages are encoded in a nested manner such that only after decrypting with the keys corresponding to the proofs of identity can the SP_{prov} retrieve the required user attributes. This forces the SP_{prov} to follow the protocol and prevent misuse of an honest user SIT attributes. □

It is important to notice that in Protocol 2 we require at least one party between the user and the service providing SP is semi-honest during the message exchanges. The worst case arises when the registrar and another SP in the federation collude. In this case, we cannot prevent the leak of information such that the registrar collects clear attributes from $SP'_{prov}s$. However since we consider multi-factor authentication, which requires proofs of other additional sensitive identifiers, the leaked attributes cannot be used successfully without the knowledge of the other users' ids. For example, to use a CCN, a user needs to prove that it has the knowledge of her SSN. Eventually, the user reveals the CCN which is required for the service (and not SSN). If the dishonest SP reveals the CCN value to the registrar the actual value of the SSN is still a secret and therefore the SSN cannot be used as proof of identity in subsequent transactions.

We do not consider a malicious-malicious case since an active malicious user and malicious service provider can agree on not following the protocol.

However, for passive malicious or faulty systems (corresponding to SP's or users) some mitigation techniques can be discussed. An extension on this topic is part of our future work.

Theorem 4 *The SIT attribute usage protocol satisfies confidentiality.*

PROOF. SP_{prov} is not required to learn any information about the sensitive attributes used as proofs of identity in the SIT attribute usage protocol. Only the tag corresponding to the identity proof is given to SP_{prov} to query SP_{reg} and retrieve the corresponding commitment. This tag as specified earlier has to be generic to avoid leaking any secret information. If SP_{prov} can get the value of the secret identifier in the commitment, then it would be equivalent to solving the DLP problem. This contradicts our assumption that DLP is hard. Therefore information about the SIT attributes as proofs of identity remains confidential to the SP_{prov} type SP's. □

In addition, the illustrated protocols are secure against man-in-the-middle and replay attacks. This is because of four main reasons. First, any message sent from the user to the SP is encrypted with the public key of the SP. Second, an explicit naming convention [1] is used by including the SSO ID of the sender. Third, timestamps are used to maintain freshness of the messages. Finally, the challenges sent by the SP are random[9] and cannot be predicted. For the final message if an adversary could successfully replay this message then it could essentially use the SIT attribute with the attached proofs of identity. This is not possible because the symmetric keys are generated in response to random challenges sent by the SP and the proofs of identity. Interestingly, in this manner even the SP_{prov} which successfully retrieves the

[9] Note that these random challenges can be made non-interactive assuming a random oracle model [6] but this is outside the scope of the paper.

user's SIT attributes as required cannot maliciously use them with any other SP. The timestamps in the final message also prevents timing and replay attacks. Note that timestamps can be replaced by counters or nounces, as suitable for the federation environment.

B. Complexity Analysis

The complexity cost is estimated in terms of the number and sizes of messages exchanged among SP's and users. In the registration phase the Protocol 1 is executed for each registered SIT attribute. The number of messages exchanged for each iteration is four. The sizes of these messages are in $log\ (p)$ or $log\ (q)$ depending of the modulus. For Protocol 2 in the attribute usage protocol, the number of messages exchanged is five. Furthermore, let n be the number of proofs of identity required to gain assurance regarding the validity of a user, and m be the number of required attributes. Then the number of times Protocol 2 has to be run is $N = n + m$. The sizes of most messages are of the same order of the sizes of the messages exchanged during the registration phase. The size of the final message after all iterations of Protocol 2 are executed is proportional to N, and the number of nested encrypted messages is N. If symmetric cipher AES is used in the CBC mode [4], then the size of each nested block is at least 128 bits. As one cipher block is added with each encryption the size of the final message is approximately $128 \times N$ bits. The registrar SP_{reg} acts like a database of information, and thus it does not represent a bottleneck in the system. To enhance efficiency, Protocol 1's multiple attributes registration can be executed in parallel because the commitments are independent of each other. Similarly Protocol 2's multiple symmetric key retrieval can also be made parallel and according to any order.

VII. Related Work

In this section we first explore the most relevant federated digital identity management initiatives and then solutions to the identity theft problem in federations. We also compare our work to some known cryptographic schemes namely anonymous credential and identity based encryptions.

In the corporate world there are several emerging standards for identity federation like Liberty Alliance [24] (LA) and WS-Federation. Because the projects are very similar we describe the former in more detail. LA is based on SAML and provides open standards for SSO with decentralized authentication. SSO allows a user to sign-on once at a Liberty-enabled site in order to be seamlessly signed-on when navigating to another site without the need to authenticate again. This group of Liberty-enabled sites is a part of what is called a *circle of trust*, which is a federation of SP's and identity providers having business relationships based on the Liberty architecture. The identity provider is a Liberty-enabled entity that creates, maintains and manages identity information of users and gives this information to the SP's. As compared to LA which uses PKI for user authentication, we show how we can also leverage the SSO ID for establishing from simple to complex digital user attributes. This adds privacy, flexibility and usability to the identity system. In addition our specific identity theft protection protocols can prove valuable when used in the Liberty identity federation framework.

Shibboleth [25] is an initiative by universities that are members of Internet2. The goal of such initiative is to develop and deploy new middleware technologies that can facilitate inter-institutional collaboration and access to digital contents. It uses the concept of federation of user attributes. When a user at an institution tries to use a resource at another, Shibboleth sends attributes about the user to the remote destination, rather than making the user log into that destination, thus enabling a seamless access. The receiver can check whether the attributes satisfy its own policies. Our approach differs with respect to Shibboleth in that we do not rely on a central identity provider providing all user attributes. User attributes in our framework are distributed within the different federation members, each of which can effectively be an identity provider. We also provide a mechanism by using which a user can receive certified attributes from the federation members and use them to obtain further certified information.

Concerning the problem of identity theft, LA, Shibboleth project and other organizations like Better Business Bureau and Federal Trade Commission have initiated some efforts aiming at educating consumers and preventing identity theft. A LA paper [18] points out that the use of SSO in federations helps reducing ID theft by reducing the number of login names and passwords which might be related to other user attribute information. The paper also discusses how attribute sharing in a federation inherently prevents user attributes theft "*by controlling the scope of access to participating websites, by enabling consent-driven, secure, cross-domain transmission of a users personal information.*" LA mitigates ID theft by having the organizations in the federation adopt superior standards of security by distributing information in order to avoid single point of failure, by having access control on these attributes based on user preferences, and by coordinating response to incidents and frauds. However to the best of our knowledge no identity provision protocols to mitigate ID theft have been developed. In particular, none of the proposed techniques in LA takes into account the case of not completely trusted members. Also LA approach currently does not address the problem of impersonation attacks where an attacker attempts to use stolen identifiers to commit fraud. Solutions dealing with such case would provide protection against insider threat or when a SP is compromised. Our solution not only exploits the advantages of a federation, as the general usage case, but extends it even further with the concept of SIT attributes and

SIT attribute usage.

RSA Laboratories' product Nightingale [34] implements a secret-splitting technology, which is designed to be integrated into application software as a server module for the back-end of any network. Secret splitting is a cryptographic technique that breaks a piece of data into two components. Learning one of these components reveals no information about the original data. Using secret-splitting the sensitive data is cryptographically distributed across two locations - the Nightingale module/server and an application server thus avoiding a single point of failure. The secret data can be of three types: (1) user authentication data like SSN, passwords; (2) business data like customer records and their CCN; (3) the cryptographic keys them-self. Nightingale can thus be used to mitigate identity theft by making it hard to retrieve the stored user information. Interestingly, the secret splitting can be used with the solution proposed in this paper to split the committed values of user identity attributes. Such an approach may provide even better identity theft protection which we will explore as a part of our future work.

Several privacy-enabled identity management systems have been based on the notion of anonymous credential [12], [13]. In anonymous credential systems, organizations know the users only by pseudonyms. Different pseudonyms of the same user cannot be linked. Yet, an organization can issue a credential to a pseudonym, and the corresponding user can prove possession of this credential to another organization (who knows her by a different pseudonym), without revealing anything more than the fact that she owns such a credential [19]. The main idea regarding use of pseudonyms in IdM systems is in that "the identity provider generates an *opaque handle* that serves as the name identifier the service provider and the identity provider use in referring to the user when communicating with each other" [39]. Rudimentary non-linkability is achieved, since an outside observer cannot infer any information about the actual user based on the random session based opaque handles. The first approach that proposed to replace user identifiers by pseudonyms is by Chaum [14]; the key idea was to use one time pseudonyms for a series of transactions to provide unlinkability among different transactions with organizations, yet at the same time transfer certified attributes among these organizations. A credential system was also employed, to ensure that only the information required for the transaction is revealed on a *need to know* basis.

Idemix [12] is the first system implementing anonymous credentials in a federated identity management system. Idemix provides mechanisms for efficient *multi-show*[10] credentials and a flexible scheme for issuing and revoking anonymous credentials. It also provides a mechanism for *all or nothing* sharing and

[10] Credentials can be used multiple times. Possession of a multi-show credential can be demonstrated an arbitrary number of times; these demonstrations cannot be linked to each other [12].

a PKI-based non-transferability. The security properties of anonymous credentials are however limited to certified attributes. Therefore anonymous credentials are not adequate for several real e-commerce applications since most of interactions rely on use of uncertified attributes. Moreover, anonymous interactions are not possible for most web services that require strong identifiers. We differ from these approaches in that we do not hide the user identity even if we protect his/her identity attributes. More specifically, we do not only protect user privacy but also protect the use of its strong identifiers without requiring anonymity.

Table II presents comparison between various anonymous credential schemes [12], [13] and our proposed SIT IdM approach according to some fundamental criteria. Two specific criteria dealing with identity theft addressed in our work are strong authentication and federation duplicate detection which to the best of our knowledge are not covered by anonymous credential schemes. Strong authentication is when the authentication requires at least two forms of identifiers. Detection of duplicate registration of strong identifiers is also a mechanism to timely prevent identity theft as elaborated in this paper.

For completeness we now describe another direction that has been followed by researchers to address phishing attacks which can potentially cause identity theft. This is the notion of Identity Base Encryption (IBE) which was first introduced and defined by Shamir in 1984 [42] and then extended by several other researchers [3], [9], [37]. An IBE scheme is a public-key cryptosystem in which any string is a valid public key. In particular, email addresses and dates can be public keys [10]. The private key is then computed by a master authority in possession of the master secret, and delivered to the proper user after proper authentication, usually via a separate channel. As a result, parties may encrypt messages or verify signatures with no prior distribution of keys to individual participants. This is extremely useful in cases in which pre-distribution of authenticated keys is inconvenient or not feasible because of technical constraints. However, to decrypt or sign messages, the authorized user must obtain the appropriate private key from the private key generator. A caveat of this approach is that this private key generator must be highly trusted.

In the approach by Adida et al. [3] the IBE is used to define and implement a cross domain identity-based ring signatures. The ring structure of these signatures provides repudiability. With identity-based public keys, a full PKI is no longer required. Separability allows ring constructions across different identity-based master key domains. Together, these properties make signature constructions a possible solution to the email spoofing problem. Our approach greatly differs from the IBE schemes because we do not provide a mechanism to encrypt data or manage certificates. Instead we focus on providing

Criteria	Anonymous Credential Scheme [12], [13]	**Proposed SIT IdM Scheme**
Digital Identity	Pseudonyms, PKI-based anonymous credentials, primary focus on *weak identifiers*.	PKI-based credentials, *uncertified attributes*, *SIT attributes*, primary focus on *strong identifiers*.
Privacy	Provides mechanism for minimal attributes disclosure from a user credential and provable unlinkability.	Based on IdM built in policies for minimal attribute disclosure and basic unlinkability based on handles. Privacy preserving multi-factor using ZKPOK of strong identifiers.
Anonymity	Provides provable unlinkability, multi-show credentials and is mainly based on disclosure of weak identifiers.	Simple unlinkabilty based on IdM architecture (IdP's and SP's), the federation privacy policies and protocols based on the use of *opaque handles*. User is not anonymous in the case of disclosure of strong identifiers.
Strong Authentication	Authentication depends on a single PKI based cryptographic secret for underlying ZKPOK therefore does not provide strong authentication.	Authentication depends on multiple non-PKI based secrets of strong identifiers for ZKPOK resulting in multi-factor and strong authentication.
Duplicate detection	No mechanism to detect duplicate registration of strong identifiers is provided.	Duplicate detection based on DHT to identify incorrect registration of strong identifiers.
Accountability	Provides global anonymity revocation mechanism which reveals user identity if she misuses her credentials. This is the PKI-assured non-transferability property. PKI based signatures and ZKPOK provides mechanism for provability.	Despite uncertified attributes accountability is provided by multi-factor authentication of strong identifiers. ZKPOK binding property provides mechanism for provability.
Confidentiality	Provides provable unlinkability, *multi show of credentials* property by which even when SP's collude no user information is leaked.	Linking of data is possible if SP's collude who have strong identifiers of a user. However due to multi-factor ZKPOK the user's personal data cannot be used. Confidentiality of data also depends on the security of the IdP databases containing the user attributes and the privacy policies.
Integrity	Unforgeability of credentials is ensured based on the assumption of strength of RSA.	Integrity is ensured based on both the binding property of the ZKPOK and on the strong RSA assumption.
Deployment Environment	PKI-based open distributed environment with external certification authority. PKI-enabled federation is included.	Federated IdM system. Public keys for SP's desired but not required for protocols. No external certification required.

TABLE II

COMPARISON OF ANONYMOUS CREDENTIAL SCHEME AND SIT IDM SCHEME

the infrastructure and methodologies to protect the identity of a user from misuse. Typically in IBE the public information, like the email address, is assumed to be correct and is denoted as the identity of the receiver. There is no clear methodology to verify and guarantee if this public information is correct and really belongs to the intended recipient. Therefore the problem of identity theft is not addressed. It will be interesting to investigate, as a part of future work, how IBE can be incorporated in our infrastructure. This might strengthen the IBE scheme by ensuring strong authentication and identity theft protection.

VIII. CONCLUSION

In this paper we have proposed a flexible and privacy-preserving approach that allows a user to establish a unique identifier and then proceed to establish other complex identity attributes in a federation. Our approach relaxes the dependence on PKI for user authentication which is currently a bottleneck for many trust management solutions. We also presented a novel solution to the problem of identity theft based on cryptographic techniques. In the paper we have also analyzed the security and complexity of the proposed protocols. The analysis also highlights the assumptions and properties that are required in order to implement the given identity theft protection solution.

Our future work includes moving from a context characterized by the semi-honest paradigm to a context where malicious SP's can collude with each other and the development of techniques to detect misbehavior. Moreover we will investigate in detail the problem of flexible and secure on-line registration as mentioned in Section IV-A.2. As pointed out, preventing a malicious user to enroll using an honest user's identity if that honest user has not registered is a major issue. Currently, this is only fully prevented if we mandate physical registration. We will design mechanisms to achieve such assurance even in case of on-line registration.

We will also explore how other cryptographic techniques can be integrated with the one presented in the paper. To this extent we will investigate the three cryptographic tools described in the related work section. First we will focus on how secret splitting protocol might be an additional mechanism to avoid a single point of failure in an IdM system. Second is with respect to anonymous credentials, where we plan to investigate how they can be used to provide the proofs of strong identity which is the primary requirement of our solution. Lastly we plan to investigate how the IBE scheme can use our multi-factor authentication to guarantee that the public information used for encryption is valid. In conclusion, we believe that our approach encourages the development of innovative cryptographic techniques and applications to address security and privacy problems in digital identity management.

ACKNOWLEDGEMENTS

The work reported in this paper has been partially sponsored by NSF under the ITR Project 0428554 "The Design and Use of Digital Identities" and by the sponsors of CERIAS. We wish to thank James Kobielus and Keith Frikken for useful comments and discussions.

REFERENCES

[1] Martin Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Trans. Softw. Eng.*, 22(1):6–15, 1996.

[2] Hal Abelson and Lawrence Lessig. Digital identity in cyberspace. In *White Paper Submitted for 6.805/Law of Cyberspace: Social Protocols*, 1998.

[3] Ben Adida, Susan Hohenberger, and Ronald L. Rivest. Separable identity-based ring signatures: Theoretical foundations for fighting phishing attacks". In *Proceedings of DIMACS Workshop on Theft in E-Commerce: Content, Identity, and Service*, 2005.

[4] AES. http://csrc.nist.gov/cryptotoolkit/aes/.

[5] Sultan Almuhammadi, Nien T. Sui, and Dennis McLeod. Better privacy and security in e-commerce: Using elliptic curve-based zero-knowledge proofs. In *CEC '04: Proceedings of the IEEE International Conference on E-Commerce Technology (CEC'04)*, pages 299–302, Washington, DC, USA, 2004. IEEE Computer Society.

[6] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

[7] Elisa Bertino, Elena Ferrari, and Anna C. Squicciarini. Trust-χ: A Peer-to-Peer Framework for Trust Establishment. In *IEEE Transactions on Knowledge and Data Engineering*, pages 827– 842. IEEE, July 2004.

[8] Abhilasha Bhargav-Spantzel, Anna C. Squicciarini, and Elisa Bertino. Integrating federated digital identity management and trust negotiation. In *review IEEE Security and Privacy Magazine*, 2005.

[9] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science*, 2139:213–240, 2001.

[10] Dan Boneh and Matthew Franklin. Identity-based encryption from the weil pairing. *SIAM J. Comput.*, 32(3):586–615, 2003.

[11] Holger Buerk and Andreas Pfitzmann. Value exchange systems enabling security and unobservability. *Computer and Security*, 9(9):715–721, 1990.

[12] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*, pages 21–30, New York, NY, USA, 2002. ACM Press.

[13] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *EUROCRYPT '01: Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques*, pages 93–118, London, UK, 2001. Springer-Verlag.

[14] David Chaum. Security without identification: transaction systems to make big brother obsolete. *Commun. ACM*, 28(10):1030–1044, 1985.

[15] Xiaofeng Chen, Fangguo Zhang, and Kwangjo Kim. A new id-based group signature scheme from bilinear pairings. In *Cryptology ePrint Archive, Report*, 2003.

[16] Jong Hyuk Choi, Sang Seok Lim, and Kurt Zeilenga. A new on-line certificate validation method for improved security, scalability, and interoperability. In *6th IEEE Information Assurance Workshop*, 2005.

[17] Rachna Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

[18] William Duserick and Fidelity Investments. Whitepaper on liberty protocol and identity theft. In *Liberty Alliance Project*, 2004.

[19] IBM Zurich Research Laboratory: Privacy enhancing Cryptography and Pseudonym Management. http://www.zurich.ibm.com/security/privacy/.

[20] U. Fiege, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 210–217, New York, NY, USA, 1987. ACM Press.

[21] Greg Goth. Phishing attacks rising, but dollar losses down. *IEEE Security and Privacy*, 3(1):8, 2005.

[22] R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure certificate and crl profile, 1999.

[23] R. Housley, W. Polk, W. Ford, and D. Solo. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile, 2002.

[24] Identity-Management. Liberty alliance project. http://www.projectliberty.org.

[25] Internet2. Shibboleth. http://shibboleth.internet2.edu.

[26] Microsoft Windows 2000 Server Public Key Interoperability. http://www.alw.nih.gov/pki/.

[27] Ken Klingestein. Emergence of identity service providers. In *EDUCAUSE Center for Applied Research, Research Bulletin*, volume 2002, 2002.

[28] Identity Theft Management. http://www.identitytheftmanagement.com/.

[29] Gurmeet Singh Manku. Balanced binary trees for id management and load balance in distributed hash tables. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 197–205, New York, NY, USA, 2004. ACM Press.

[30] Gurmeet Singh Manku. *Dipsea: a modular distributed hash table*. PhD thesis, 2004. Adviser-Rajeev Motwani.

[31] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. Handbook of applied cryptography, 2001.

[32] Victor S. Miller. Elliptic curves and their use in cryptography.

[33] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 internet public key infrastructure online certificate status protocol - ocsp, 1999.

[34] RSA Laboratories' Nightingale. http://www.rsasecurity.com/rsalabs/node.asp?id=2424.

[35] Eric Norlin and Andre Durand. Whitepaper on towards federated identity management. In *Ping Identity Corporation*, 2002.

[36] National Research Council of the National Academies. *Who Goes There? Authentication Through the Lens of Privacy*. The National Academies Press, Washington, D.C., 2003.

[37] Kenneth G. Paterson. Id-based signatures from pairings on elliptic curves. *Cryptology ePrint Archive, Report*.

[38] Vipul Ved Prakash and Adam O'Donnell. Fighting spam with reputation systems. *Queue*, 3(9):36–41, 2005.

[39] Liberty Alliance Project. Liberty protocols and schemas specification, version 1.0, 11 july 2002.

[40] Vipin Samar. Single sign-on using cookies for web applications. In *WETICE '99: Proceedings of the 8th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises*, pages 158–163, Washington, DC, USA, 1999. IEEE Computer Society.

[41] Claus P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 239–252, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[42] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[43] Victor Shoup. Lower bounds for discrete logarithms and related problems. *Lecture Notes in Computer Science*, 1233:256–268, 1997.

[44] Alma Whitten and J. D. Tygar. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In *8th USENIX Security Symposium*, 1999.

[45] Petra Wohlmacher. Digital certificates: a survey of revocation methods. In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 111–114, New York, NY, USA, 2000. ACM Press.

[46] David Woodruff and Jessica Staddon. Private inference control. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 188–197, New York, NY, USA, 2004. ACM Press.