User Centricity: A Taxonomy and Open Issues[*]

Abhilasha Bhargav-Spantzel[†]
Purdue University
bhargav@cerias.purdue.edu

Jan Camenisch
IBM Zurich Research Laboratory
jca@zurich.ibm.com

Thomas Gross
IBM Zurich Research Laboratory
tgr@zurich.ibm.com

Dieter Sommer
IBM Zurich Research Laboratory
dso@zurich.ibm.com

ABSTRACT

User centricity is a significant concept in federated identity management (FIM), as it provides for stronger user control and privacy. However, several notions of user-centricity in the FIM community render its semantics unclear and hamper future research in this area. Therefore, we consider user-centricity abstractly and establish a comprehensive taxonomy encompassing user-control, architecture, and usability aspects of user-centric FIM. On the systems layer, we discuss user-centric FIM systems and classify them into two predominant variants with significant feature sets. We distinguish *credential-focused* systems, which advocate offline identity providers and long-term credentials at a user's client, and *relationship-focused* systems, which rely on the relationships between users and online identity providers that create short-term credentials during transactions. Note that these two notions of credentials are quite different. The further one encompasses cryptographic credentials as defined by Lysyanskaya et al. [30], the latter one federation tokens as used in today's FIM protocols like Liberty.

We raise the question where user-centric FIM systems may go—within the limitations of the user-centricity paradigm as well as beyond them. Firstly, we investigate the existence of a *universal* user-centric FIM system that can achieve a superset of security and privacy properties as well as the characteristic features of both predominant classes. Secondly, we explore the feasibility of reaching beyond user-centricity, that is, allowing a user of a user-centric FIM system to again give away user-control by means of an explicit act of *delegation*. We do neither claim a solution for universal user-centric systems nor for the extension beyond the boundaries of user-centricity, however, we establish a starting point for both ventures by leveraging the properties of a credential-focused FIM system.

Categories and Subject Descriptors: H.4.3 [Information Systems Applications]: Communication Applications; H.4.m [Information Systems Applications]: Miscellaneous; K.6.5 [Management of Computing and Information Systems]: Security and Protection—authentication; K.4.1 [Computers and Society]: Public Policy Issues—privacy, transborder data flow

General Terms: Security, Design, Management

Keywords: Identity management systems, user centric, user control, taxonomy, delegation, security, privacy

1. INTRODUCTION

An individual's identity in the digital world is represented by a set of attributes. These attributes can simply be claims made by that user that have not been certified by a third party, or attributes verified and endorsed by a third party. An individual can potentially have several different identities, corresponding to different sets of associated attributes. The life cycle of an identity roughly consists of enrollment, storage, retrieval, provisioning and revocation of identity attributes.

A *federated identity management* (FIM) system consists of software components and protocols that handle the identity of individuals throughout their identity life cycle. A FIM system involves three main entities, namely *user*, *identity provider* (IdP) and *service provider*. The IdP manages and potentially issues user credentials, and the service providers (also known as *relying parties*) are entities that provide services to users based on their attributes. Note that there are several social, economic, and legal requirements to realize a FIM system. For example, the legal requirements would have to dictate how the contracts for transactions limited to the physical world get adopted when these transactions are performed electronically. Those non-technical requirements are to be addressed when building a FIM system, but they are out of the scope of this paper as our focus is only on the technical issues. See for example Europe's PRIME project [35] for material regarding such requirements.

1.1 User Centricity

A recent paradigm of identity management is *user-centric identity management*, which is the primary focus of this paper. A user-centric identity management system needs to support user control and consider user-centric architectural and usability aspects. Based on current user-centric FIM systems, we differentiate between two

[†] Work performed during an internship at the IBM Zurich Research Laboratory in Switzerland.

[*] Part of the work reported in this paper is supported by the European Commission through the IST Project PRIME. The PRIME project receives research funding from the European Community's Sixth Framework Programme and the Swiss Federal Office for Education and Science.

DIM'06, November 3, 2006, Alexandria, Virginia, USA.
Copyright 2006 ACM 1-59593-547-9/06/0011 ...$5.00.

predominant notions namely *relationship-focused* and *credential-focused* identity management. Both models put the user in better control of her attribute data, but by using fundamentally different approaches. In the relationship-focused approach, a user only maintains relationships with identity providers and thus each transaction conveying identity information to a service provider involves the appropriate identity provider. The user has control over her attributes in that she is involved in every identity provisioning transaction. On the contrary, the credential-focused approach is based on the user obtaining long-term credentials from the identity provider and storing them locally. These credentials can then be used to provide identity information without involving the identity provider. Similar to the relationship-focused notion, the user is involved in every identity transaction as well.

For clarity, we can think of an analogy between the two notions of user centricity in the physical world: A credit card can be considered a specific relation with an identity provider (the authority issuing the credit card). At the time of use of the credit card, the credit card company is usually contacted to approve the transaction. This resembles the relationship-focused system with the credit card being the relation with the credit card company. On the other hand, the use of a passport for age verification in a bar corresponds to the credential-centric notion, the credential by itself is sufficient and the identity provider (passport issuing authority) is not involved. This requires that the passport credential itself be hard to forge. Though, when a passport is used to leave or enter a country, its (revocation) state is checked with an on-line authority to enhance security and account for timely propagation of (revocation) information.

Each of the above paradigms has advantages of its own, neither one qualifying as being clearly better than the other. At this point we raise the question, whether it is possible to go beyond the current notions to obtain a *universal* FIM system incorporating the advantages of both the user-centric system types. Moreover, such a universal FIM system should be able to combine various other aspects of user centricity, not necessarily addressed in current systems, as needed per application.

Ironically, the major advantage of user centricity—user control through her involvement in each transaction—amounts for the major drawback of user centricity: not being able to handle delegations. Though, a universal FIM system can go beyond the restrictions of user centricity to provide a complete identity management solution.

1.2 Evolution of Identity Management

To motivate why user centricity is becoming a key paradigm in identity management, we provide a brief sketch of the evolution of identity management.

The most predominant identity management system deployed in current-day Internet is what is commonly known as the *silo model*. Here the users handle their identity data and provide it separately to organizations that do not have any mechanisms to share this identity information with other organizations. This makes the identity provisioning cumbersome for the end user and the identity management system inflexible and closed. Therefore, as a next step, the so-called *centralized federation model* like Microsoft Passport emerged, which looked into a possible solution to avoid the redundancies and inconsistencies in the silo model and to give the user a seamless experience. Here a central IdP became responsible for collection and provisioning of the user's identity information in a manner that enforced the preferences of the user. This approach had several drawbacks as the IdP not only becomes a single point of failure but also may not be trusted by all users and service providers.

The next step was then to decentralize the responsibility of the IdP to multiple such IdPs which can be selected by the end users. In such *federated systems*, multiple IdPs are distributed and can store partial identity information of users if required. Other rules and particular protocols are defined by several well-established or upcoming standards [25, 23, 22]. This avoided the problem of a single point of failure, but required that an IdP be chosen that is also trusted by other entities. In most of these systems the user had to be dependent on an online IdP to provide the required credentials and hence these systems were referred to as *provider centric*. They clearly lacked user control on her credentials, and therefore the current trend is moving away from them.

As a result, a currently emerging paradigm is that of *user centricity*, that is, the idea of giving the user full control of transactions involving her identity data. This paradigm is embraced by multiple industry products and initiatives such as Microsoft CardSpace [31], SXIP [24] or the open-source Higgins Trust Framework [20]. In the recent past, the exact definition of what it means to be user centric has been argued extensively without a clear conclusion. Other terminology often used closely with user centric are "user control," "user consent," and "user in the middle". Interestingly, the silo model may be considered to have good user control, however, as mentioned above, this was more of a burden than an advantage. Thus, incomplete understanding and implementation of the new user-centric systems may bring us back to square one of the evolution of identity management systems if the new systems do not incorporate the advantages presented by the previous approaches. Our aim is to understand the concept of user centricity and also investigate the next steps in the evolution of the FIM systems.

1.3 Contribution

As a first contribution, we take a conservative approach and aim at consolidating the different aspects of user-centric identity management. In particular, we aim at elaborating the different aspects of user centricity in a FIM system. In Section 2, we establish an abstract taxonomy of an ideal user-centric system and describe its various aspects.

In Section 3, we provide, as a second contribution, a detailed discussion of the predominant paradigms of user-centric systems and how they satisfy certain aspects of our taxonomy. More precisely, we compare the relationship-focused with the credential-focused systems and discuss the distinguishing features of each of the two paradigms. This is followed by investigating the notion of a *universal* user-centric system that incorporates the advantages of both the systems. In particular, we see how we can go beyond the boundaries of existing systems and further, the boundaries set by user-centric FIM systems themselves. One example of a limitation of the typical user-centric approach is that of delegation and we elaborate on the open issues in this aspect.

Finally, as a third contribution, we discuss in Section 4 which technical mechanisms can be used to construct identity management systems that satisfy a given set of properties of our taxonomy. We distinguish between a weak and a strong trust model and put a focus on mechanisms that help realize better user privacy.

2. TAXONOMY OF USER CENTRICITY

Realizing a user-centric identity system concerns several distinct aspects. A key property of a user-centric FIM system is that of *user control* for which we first provide a comprehensive and detailed analysis. In addition, we note that deploying a user-centric system is not trivial based on current technologies which are predominantly provider centric. Therefore, we briefly describe the specific

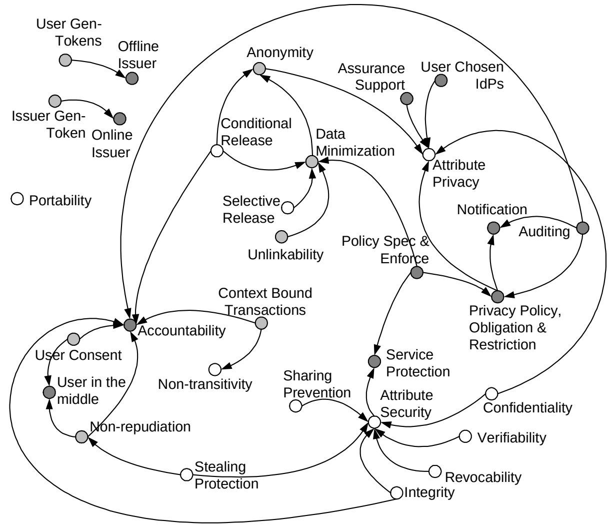

Figure 1: Taxonomy of the user control aspect of user-centric identity management

architectural properties needed for the deployment of a user-centric system. Finally, we complete the taxonomy by highlighting the usability concerns that are critical aspects that should be addressed while realizing such a system. In essence, our taxonomy consists of three main aspects, namely *user control, architecture and deployment*, and *usability*.

While reasoning about the security and privacy properties, we also refer to the OECD principles [34]. The OECD guidelines are widely accepted and form the cornerstone of fair information practices and regulations designed to protect personal information around the world. The user-centric FIM should satisfy the given OECD principle while providing the system property as relevant.

We also refer to Cameron's *Laws of Identity* [15] which are a recent set of prevalent guidelines regarding digital identity management. They aim at explaining the successes and failures of digital identity systems. They include design principles and rules desired to achieve several security and dependability properties.

While reasoning about which properties are required to achieve the required system, it is important to define the trust model of such a system. We consider two distinct models based on the parties that are controlled by the attacker and whether passive or active attacks are considered.

1. **Weak Trust Model.** The identity provider is completely trusted to follow the protocols and to not engage in any attacks, not even passive ones.[1] That is, the issuer is not controlled by the attacker.

2. **Strong Trust Model.** The identity provider is controlled by the attacker and is thus not trusted to follow the protocols and can engage in arbitrary attacks.

In both of the models above the service providers are also controlled by the attacker and the users can be possibly malicious and try to get an advantage.

Based on the above principles and models we first elaborate on the user control aspect of a user-centric FIM system.

2.1 User Control

The key idea in user-centric FIM which separates it from other systems is the user control on her attributes, in particular on the aspect of releasing attribute information. User control and consent is also defined as the first law of identity in Cameron's Laws of Identity. User control is achieved by realizing manifold system properties. Some of these properties are *basic properties* in that they are not based on the realization of other properties, while others are *composite properties*, composed of other basic properties. The properties of our taxonomy related to user control are illustrated as nodes of the directed graph (more precidely, directed forest) in Figure 1. Composite properties tend to be more general and may depend on several basic properties, although it depends heavily on the assumed trust model for a particular system whether one property is required for another. Based on this categorization we highlight the various aspects of user control in the following.

2.1.1 Basic Properties

The basic properties of user-centric FIM systems either apply to 1) the entire FIM system, 2) transactions in the system, and 3) the identity information or credentials of the entities involved. Though, this classification is not exclusive. These basic properties are the nodes with indegree 0 in the directed graph in Figure 1. System properties in the graph are represented by dark grey nodes, transaction properties by light grey nodes, and properties of identity data by white nodes.

2.1.1.1 FIM System Properties.

We identify four basic system properties. First is the *user-chosen IdP* property which means that the user can choose between multiple IdPs. Thus the user is not confined to a defined IdP which she may or may not trust. This choice also helps in achieving the justifiable parties rule in the Laws of Identity. Second is *policy specification and enforcement* which relates to the definition, management and realization of multiple policy-related issues. Several of the other properties build on the capabilities of the system to be capable of dealing with those policy-related issues. The third system property is that of *auditing*. The audit subsystem needs to be defined in such a way that it can be used to achieve the other desired properties of the FIM system using appropriate mechanisms. In particular, the granularity of audit logs and definition of event classes being logged are key issues to be considered. The fourth system property is *assurance support*, that is, providing mechanisms that allow the user to gain assurances from the server system or its owner organization. Assurances help the user in deciding whether and to what extend to trust a potential recipient of her identity data.

2.1.1.2 Transaction Properties.

Transaction properties concern all the transactions which deal with identity-related information.

First is the *context bound transactions* property which requires that the messages of a transaction be bound to the context that transaction is being executed in. That is, the messages of the transaction are worthless in a different context. Second is the *unlinkability* of transactions which means that transactions can be unlinkable to each other with respect to the end entities (like verifier or issuer). This is assuming that the identity assertion being conveyed in the transaction does not establish linkability. For example, if the assertion contains a unique identifier referring to a particular user, then another transaction using the same information is trivially linkable.

An important property of a transaction is to include *user consent*. A user giving her consent means that the user's view of the trans-

[1] A passive attack is defined by the attacker taking advantage of the information obtained from all controlled parties, but always correctly following the protocols.

action corresponds to the actual transaction and that the user agrees to the execution of the transaction. The significance of this is also highlighted in Cameron's first law.

Depending on who generates the identity token being provided to the service provider, we can distinguish between the case of *user-generated tokens* and *issuer-generated tokens*. Using appropriate mechanisms, in both cases the same security properties can be obtained. Though only protocols with user-generated tokens can provide some privacy properties in the strong trust model.

2.1.1.3 Identity Information Properties.

A well-established taxonomy divides several computer-security-related issues into three categories, namely confidentiality, integrity, and availability. We identify the desired properties for identity information first based on these properties and then go beyond this to address further issues that come into play in a FIM system.

Confidentiality may be defined as the protection of sensitive information from unauthorized disclosure. This requires that the identity information is only accessible by the intended recipients. If an attacker can retrieve this information, then the user control on the attribute release and usage is (partially) broken. It is therefore essential that the credential disclosure subsystem provide mechanisms for confidential release of the user's attributes and that identity information be protected accordingly at all times. This property can also be related to the directed identity rule of the Laws of Identity.

Integrity is defined as the condition that data has not been altered in an unauthorized way. In our discussion we use integrity to specify that the identity information as issued by the identity provider has not been changed. We note the special case of self-asserted identity, where the user is their own identity provider. Integrity is a generic property essential for any identity management system. The fifth OECD principle named as security safeguard principle also indicates the requirement of securing user data from being tampered with. Certification of attributes is a method to meet such a requirement. This is important since no real guarantees can be based on attributes which are simply voluntary claims especially if they deal with sensitive information and assurance that it is being provided by the owner of the information. We also define the *verifiability* property meaning that the user can verify that the identity provider provides the correct identity data about the user and according to the user's intention. This property is related to the user consent property.

Stealing protection applied to identity data and in particular credentials and private keys addresses the issue of protecting against malicious viruses, hackers, or other unauthorized entities illegitimately trying to get hold of a user's data items. Without stealing protection it is impossible to achieve properties like non-repudiation or attribute security.

Revocation of identity information is required to maintain the validity of the information where this has a major implication on the security of the information. More specifically, if the information is endorsed by an identity provider, e.g., through a certificate, then there should be a way to revoke the endorsement. Security of the attributes in an identity management system can only be guaranteed with appropriate revocation mechanisms for already issued credentials. Revocation in systems where the issuer is providing the required credential to the user each time she needs to use it is simple to solve. Such credentials are typically short term, and cannot be used without consulting the issuer again. If, however, the credentials are indeed stored with the user, such as a long-term credential issued by the appropriate authority, then building an appropriate revocation system becomes more challenging and critical.

Selective release of identity information means that identity information can be released at a fine-granular level by the user.

Related to selective release is *conditional release* that is concerning the release of identity information such that it becomes available to the recipient only once a condition is fulfilled. The recipient obtains a guarantee that they will obtain the information once the condition is fulfilled. Conditional release can be useful for anonymity revocation in anonymous settings: The user conditionally releases identifying attributes that can get available to the recipient once a well-defined revocation condition is fulfilled.

Sharing prevention prevents users from giving their credentials to other parties who use them in an unauthorized way, e.g., to illegitimately access services. Moreover, malicious users could pool their credentials to attain higher privileges than each of them would have on their own. An access control decision based on a pooled combined set of credentials would be flawed and lead to security threats. Pooling prevention is a special case of sharing prevention. Due to the security threats, sharing prevention should be enforced by a user-centric system.

The final property in this category is that of *portability* of identity information referring to support for the user in using her credentials on multiple of her devices. This flexibility is used for many typical user scenarios, for example, ones involving a desktop machine, a laptop, and a smart phone. This property may require intricate mechanisms depending on the identity management mechanisms and protocols being used.

2.1.2 Composite Properties

We define several composite properties which can build on one or more of the basic properties. These composite properties are illustrated as the nodes with an indegree of greater than zero in the directed graph in Figure 1. We highlight the dependencies by drawing a directed edge from a property A to the composite property B with the semantics that A is required or helpful in achieving B. Which basic or composite property helps in achieving a composite property can depend on the assumed trust model.

The *attribute security* property reflects a comprehensive notion of security of a user's attributes. A main focus is on the correctness of attributes in the view of a service provider meaning that the attributes belong to the person executing the transactions. This requires the attribute information to be integrity protected and stealing protection and sharing prevention must be in place in order to avoid another person maliciously or with the user's help taking over the user's identity. Furthermore, revocation of identity information must be feasible. Attributes in certain cases must be kept confidential with respect to other parties than the ones involved in the transaction.

Service protection accounts for the security of attributes and the enforcement of the service release policies of the service provider based on these attributes of the requester, that is, service access is only granted to authorized requesters.

The *non-repudiation* property of messages means that a non-repudiable message can be linked to the entities involved. Linkability is restricted to when the conditions defined by the policy are satisfied. Non-repudiation is a generic security property desired in any identity management system. Mutual non-repudiation gives a guarantee that the user cannot later deny having executed a particular transaction and the service provider cannot deny having been involved in the transaction. The requirement for non-repudiation is also indicated in the seventh OECD principle named individual participation principle. Interestingly, the property of *repudiation* may also be desired in certain interactions when the user may need to be anonymous.

The *non-transitivity* property addresses the impossibility for a recipient of identity information to reuse this identity information using the obtained security tokens.[2]

Data minimization deals with the minimal data release within a transaction. Minimal here means that only data be requested and released that are required by the service provider to provide the service. Data minimization can be achieved by having according policy system support, by having unlinkable transactions, and by having a data release system that allows for selective release and conditional release of identity information. This corresponds to the first OECD principle relating to collection limitation. This principle is also reflected in the European Data Protection Directive 95/46/EC [18] and the national data protection laws within the European Union. It is to be noted that data minimization must not harm the attribute security or service protection. An additional requirement for data minimization concerns the service release policies of the service providers that must be capable of supporting this property. In a user-centric system, the users should have the option to provide minimal information required to qualify for a service. This is beneficial both to preserve the privacy of the user and decrease the service provider's cost for regulatory compliance and decreases potential liability in case of exposure of user identity information. Data minimization has been emphasized in related work of [14] to have diverse and important implications for an identity management system.

Attribute privacy refers to the concept of giving the user control over her attribute data. This is supported by giving the system assurance support and allowing for user-chosen IdPs. Both those properties account for user-centric decisions on which IdP to trust. Anonymity and its dependend properties play a major role in attribute privacy in that it helps avoid the unnecessary release of (identifying) information. An orthogonal property essential for reaching attribute privacy is the support of privacy policies, obligations, and restrictions. Confidentiality ensures that attributes are not unintentionally disclosed to any party. However, similar to protection against malware, additional mechanisms may be required to provide resistance to the different types of identity theft. For example, if a particular user-centric FIM system lets the user store her own credentials on her device then further measures to secure the credentials are needed in case this device is lost. Securing user data where it is stored is also stressed in the fifth OECD security safeguard principle.

Accountability refers to the ability of holding entities responsible for their actions. This is concerning user transactions and use of identity information at the service provider and identity provider. FIM systems have typically been focused on underpinning accountability in business relationships and checking adherence to regulatory controls. As in user-centric systems the identity information of a user is provided via the user's client, security properties have to hold, in particular integrity, such that accountability still holds and the person can be held accountable. Accountability also becomes a significant issue if a user-centric system enables the user to stay anonymous as accountability and anonymity are per se contradicting properties. Nevertheless, conditional release of identity information can help in obtaining accountability in anonymous transactions. The eighth OECD accountability principle is devoted to understanding accountability, especially as it relates to privacy.

The property *privacy policy, obligations, and restrictions* deals with the policies defined in the system and their enforcement. While protecting the privacy of a user's identity information, it is important to define the circumstances when identity information can be used and for what purpose. Defining such requirements needs a *privacy policy* to be provided at the time of the release of the attributes. Most OECD guidelines aim at general standards for privacy rules and the third principle especially highlights the purpose specification of the released data. Several policy languages have been developed [17, 3] which address this concern and the enforcement of such policies remains a crucial aspect which needs to be addressed to make such policies meaningful. The policies incorporate user consent on the release and usage of her identity information. This is also related to the fourth OECD principle, the use limitation principle. Obligations [4] are concerned with commitments of the involved parties in a given transaction. In most of the related work, obligations have been considered from the service provider's point of view. However, in a user-centric system, if the user is given complete control of the release of her credentials then it becomes essential for this user to satisfy the defined obligations. Similarly, the methodologies to define restrictions provide a way to understand conflict-of-interest concerns and other regulatory aspects depending on the temporal events of the user.

Anonymity in transactions deals with the subjects remaining anonymous within the anonymity set, that is, being not identifiable within this set. Anonymity is a specific notion related to data minimization, obtainable when the released attributes are not identifying the user. Anonymity is supported by unlinkable transactions, without unlinkability the anonymity set shrinks quickly in practice when executing several transactions. Pseudonymity—the use of pseudonyms as user identifiers—is a concept strongly related to anonymity. Conditional anonymity—anonymity that holds only as long as a well-defined condition has not been fulfilled—can be provided based on conditional release of the identity information. In this way, mechanisms providing for anonymity are still useful as they can be complemented with those for realizing accountability.

Another important composite property is that of *notification*. It is desired to enhance the control of the user so that she is able to receive ("push" model) and retrieve ("pull" model) notifications regarding the usage of her identity data. This is particularly important when there is a security breach leading to user's identity information to be compromised at an external entity. The sixth and seventh OECD principles of openness and individual participation can potentially be satisfied using comprehensive notification mechanisms.

A final important system property discussed here is the *user in the middle* property. This property represents how the user is involved in providing the identity information to the service provider. This property can come in two flavors: In one case, the user's client is the one which simply transfers the final token provided by the issuer. In another case the actual user is involved in constructing the token and sending it to the verifier. This property corresponds to the OECD principle of individual participation and has been of concern for several identity management systems.

2.2 Architecture and Deployment

Deployment consists of those activities that need to be performed with a software product after it has been released [19]. Deployment of identity management software includes installing, configuring, and updating the program or components, that is, to enable a user to execute the different components of the system. Requirements of a user-centric deployment were highlighted in [29] which stressed on the following three aspects: 1) to have an *interference-free* deployment such that the new components do not disrupt the already installed components of the user system; 2) to have *independent deployability* and absence of strict dependencies to allow for flexibility and choice of configurations to the user; and finally 3) *compatibility with legacy code* which is especially crucial because of

[2] Clearly, the identity information itself can be reused in a typical setting as it becomes known by the recipient.

the update and management of a large number of components that already exist with all the different users.

The *user in the middle* paradigm on the architectural layer defines that the identity data always flows through the user's identity client. It is claimed that in a user-centric system the identity provider does not have a priori knowledge of the service provider, only a trust relationship from the service provider to the identity provider must exist. Note that *user in the middle* does not make any assertions about the involvement of the human user, e.g., for approving every transaction.

A unique property which is essential for a user-centric system is that of *multi-device management*. If, for example, a user has her credentials stored in a local PC and then has a separate laptop, a user-centric system should provide functionality to let the user use her credentials regardless on which device they are stored or have been obtained with. This is closely related to the portability aspect of identity information.

There are other properties generic to all FIM systems and therefore we do not elaborate on them. This includes the system being *fault tolerant* and *dependable*. The user-centric system should be able to survive failures of the federation entities, and the service should be able to comply with the different dependability requirements as appropriate for a given application. The system should be deployable in a *cost effective* manner. One of the key goals of a federation system is the cost effectiveness which should not be compromised while integrating user-centric features. Moreover, the cost of establishing, using, and maintaining user credentials should be adequate. Another aspect of cost is the *efficiency* of the protocols themselves.

2.3 Usability

Usability addresses the relationship between the user-centric tools and their users. In order for a tool to be effective, it must allow intended users to accomplish their tasks in the best way possible. The key principle for maximizing usability is to employ iterative design, which progressively refines the design through evaluation from the early stages of design [1]. Some key aspects are 1) to have *consistent user experience*, 2) an *intuitive and easy UI* which may also help required functionality from the user like policy specification, and finally 3) *process automation*, that is, automating user-side processes of identity management as far as possible through policy and preferences-driven methods.

3. BEYOND THE BOUNDARIES

Where Section 2 covers abstract properties of user-centric FIM, in this section, we move on to existing systems and their limitations. In Section 3.1, we analyze classes of systems that follow the user-centric paradigm and point out their differences as basis for our further discussion. Taking this as basis, we ask the fundamental question, how we can move beyond the boundaries and limitations of these classes.

Within Section 3.2 we ask the question whether there exists a universal system in the user-centric space, i.e., a user-centric system that satisfies all properties in the taxonomy of Section 2 as well as subsumes the design classes shown in Section 3.1. Such a system would be a perfect solution for user centricity, however, still remain within the limitations of the paradigm itself.

In Section 3.3, we consider solutions that go beyond the problem space of user centricity, that is, beyond the inherent limitations of user centricity. We consider the problem that a user may give up the user control established by a user-centric system again and leverage the advantages of non-user-centric FIM as well. We believe that the key aspect for this solution will be efficient and flexible delegation of identity information and rights associated with them.

3.1 Existing Systems

Clearly, the space of user-centric FIM systems is very heterogeneous. For the sake of this discussion, we classify the systems according to one important distinguishing feature that influences multiple key aspects of the system, namely, the *design focus*. As *design focus*, we define the type of identity data or meta data that (a) is presented to the user in the user interface; (b) a user's client of a user-centric FIM system manages. We claim that there are, in principle, two extremes of user-centric systems: *relationship-focused* systems and *credential-focused* systems. Of course, the parameters of each system class can be altered such that they get more similar to the other one, however, for the purpose of this discussion we choose typical instances of these classes. In the following two subsections we describe the advantages and disadvantages of the typical systems of both classes. We describe these notions briefly and sketch an overview in Table 1.

A *relationship-focused* system (Table 1, middle column) is characterized by the FIM system only managing relations to identity providers and collaboration partners. We call this design focus *relationship focused*. In such a system, the user's client queries an identity provider, with which the user has a relationship, in each transaction and retrieves identity information dynamically during the transaction. Usually the identity information is transferred in short-term identity federation tokens, such as SAML [33], Liberty [28], or WS-Federation [25] tokens. Such a *security token* is usually a statement about a user's identity or attributes that is simply signed by the identity provider.

In a *credential-focused system* (Table 1, right column), the FIM system manages the user's credentials directly, i.e., the client holds the user's long-term credentials in a local wallet. Thus, the user can leverage the credentials in multiple transactions without involving the original identity provider again. In such a transaction, the user's client either reveals the full credential (in the case of an X.509 client certificate) or shows properties of the credential (such as in systems based on zero-knowledge proofs of knowledge like Brands' system [7] or *idemix* [10]).

We need to clarify that a credential as used in credential-focused systems is not just a security token with long-term lifetime, but a non-transitive cryptographic credential as defined by Lysyanskaya et al. [30]. A credential empowers the user to make statements about her identity and attributes by herself (i.e., the user needs to own a piece of data equivalent to a private key) once the credential has been obtained from an identity provider while retaining the certification by the identity provider. The use of the credential leads to a token to be provided to the relying party or an interactive protocol.

The *non-transitivity* we mention here means that the entity that receives the credential or a proof statement generated from a credential cannot reuse it to make the same claim.

3.1.1 Relationship-Focused Systems

3.1.1.1 Advantages.

Relationship-focused identity systems typically issue *short lifetime* tokens used for immediate access control by the user. Such a restriction limits the risk and damage in case this token is stolen. It also mitigates the possibility of sharing of these credentials within the period of their validity [*Sharing Prevention*].

An online identity provider in such systems allows for online verification of the validity of the user's account. This guarantees freshness and up-to-date attributes of the identity tokens issued (*Cor-*

Table 1: Design focus of user-centric FIM systems.

	Relationship Focused	**Credential Focused**
User holds	Reference to issuer	Long-term credentials
Issuer	online	offline
Token validity	short-term	long-term
Setup	Establish relationship	Issue credential
Transaction	Upon user request, issuer creates new short-term token.	Issuer not involved. User "shows" credential or property thereof.
Transitivity	Restricted by audience	Enforced by crpytographic means

rectness, *Integrity*). Consequently, there is no immediate need of *Revocation* capabilities.

In general, relationship-focused systems can be lightweight and do not necessarily require a rich user client, such as in the case of browser-based protocols (passive requestor profiles [33, 26]). Also, they only need to rely on well-known public-key cryptography.

3.1.1.2 Disadvantages.

By construction, relationship-focused FIM systems require the identity provider to be online during transactions. This renders the identity provider a single point of failure for these systems and imposes requirements of high uptime and quality of service on the identity providers (*availability*). Consequently, identity providers are costly to deploy and operate (*cost effective*, *efficiency*).

As the identity provider is always involved in user transactions, the identity provider can trace the user's activities (partners, URIs, attributes revealed, timing) and, therefore, potentially infringe the user's privacy (*aata minimization*, *anonymity*).

Sharing and theft are still possible in relationship-focused systems.[3] Possible means of theft are spoofing or man-in-the-middle attacks, which can possibly compromise a token for the brief interval of its lifetime. Also, the information used for the initial authentication at the identity provider such as a user's username and password combination may be stolen or shared.

Relationship-focused systems are endangered when it comes to the transitivity of their tokens. A token is called *transitive* if a principal that receives the token may use it to impersonate the token holder. In existing relationship-focused systems transitivity is prevented by suitable setting of provider and audience fields[4] in messages and the assumption that honest principals will reject messages not intended for them. As soon as the audience contains more than one principal or is even left out, a user may fall prey to impersonation attacks by so called bogus merchants (*authentication*, *Authorization*), that is, a principal acting as service provider misusing the credentials received in a valid protocol run.

3.1.2 Credential-Focused Systems

3.1.2.1 Advantages.

By construction, the identity provider of credential-focused systems is offline during transactions, such that there is no runtime dependency on the provider's quality of service (*availability*).

This principle also guarantees that identity providers cannot trace the behavior of their users, which is a major step towards *Data Minimization* and *Anonymity*.

When it comes to non-transitivity, credential-focused systems need to be non-transitive by construction or their long-term credentials will be vulnerable by impersonation attacks. Given systems indeed enforce non-transitivity by cryptographic means such that a token generated from a credential cannot be used later by its recipient or other party as valid token in other contexts (*authentication*, *Authorization*).

3.1.2.2 Disadvantages.

Theft or sharing of the long-term credentials of such a system may inflict a large damage and thus introduce a risk to credential-focused systems. Therefore, *Sharing Prevention* is a must-have of credential-focused systems.

Clearly the possible impact of loss or compromise of long-term credentials and the dynamic nature of certain identity information imposes the requirement of *Revocation* of credentials on credential-focused systems. The revocation capabilities give a user the option to terminate the lifetime of her credential as soon as she or some other entity perceives the loss or misuse of the credential and also allow the identity provider to revoke a credential on their behalf once a user does not qualify for the credential any more. Research already provides several ways of handling revocation of credentials efficiently, even while maintaining the *Data Minimization*, *Unlinkability*, and *Anonymity* properties of these systems.[10]

Credential-focused systems naturally generate a higher workload on a user's client and require a rich client to be executed.

3.2 Does a Universal System Exist?

With respect to the proposed two classes of user-centric FIM, the universal system is a system that is capable of fulfilling both classes seamlessly. This means we are looking for a universal user-centric FIM system that can have long-term as well as short-term credentials, online as well as offline identity providers, and satisfying a large set of the properties we named in the taxonomy of Section 2, particularly in the security and privacy area.

To our knowledge, the universal system has not been found, yet, that is, there exists no single system that combines advantages of both flavors and fulfills a superset of their properties.

Nevertheless, we believe that such a universal user-centric FIM system may exist and will pursue a discussion how to achieve it. We can hope that it is a viable way to take a given system as starting point and extend it by further properties to achieve this goal.

3.2.1 Relationship-Focused as Starting Point

We start our discussion with a relationship-focused system as basis. One could cache the tokens used in such a system and simply extend their lifetime. However, we have seen that relationship-focused systems existing today only have limited means of restricting the transitivity of their tokens. Either the token is issued with a limited audience set, which in turn pre-determines the use of the token. Or the audience encompasses a large set of trusted principals that raises the risk of bogus merchant attacks (*authentication*, *authorization*).

Also, the relationship-based systems need to present a full token in order to get a signature validation of the token, which always reveals the full data set (*data minimization*). Moreover, because of the static signature bound in such a token, the tokens of a relationship-focused system are per se linkable, which allows tracing and infringes the user's *anonymity*.

We conclude that today's relationship-focused systems are diffi-

[3] This may be limited by hardening the operating system or binding the client to a hardware module such as a Trusted Platform Module (TPM).

[4] WS-Federation, SAML, and Liberty have in common that they allow an identity provider of a security token to specify a set of principals that may accept the token as part of the message. Principals must reject a message that does not name them in the audience field.

cult to use in a long-term credential setting and may infringe security as well as privacy properties.

3.2.2 Credential-Focused as Starting Point

Once we consider credential-focused systems as a starting point on our quest for a universal user-centric FIM system, we observe that such a system may be trivially set to short-term credentials. Furthermore, the means of sharing prevention in credential-focused systems may even be used to enforce that a credential can only be shown once or k times by a user.

Some credential-focused systems already provide strong *Data Minimization* and *Anonymity*. We want to keep these properties as we reduce the lifetimes of the credentials and involve the provider in more transactions. In general this is possible, as these systems allow a selective release of single attributes (*data minimization*). For the *Anonymity* property, we can leverage the property that the policy of the credential issue is decoupled from the policy of the service provider and the service provider's address.

The credential-focused systems introduce a significant workload to a user's client and the provider that does not amortize such well anymore once the credential lifetimes get shorter. Of course, one can limit a credential-focused system by having the provider only "showing" a credential on behalf of a user instead of handing over the credential itself, which renders the system much more efficient and truly relationship-focused.

We believe that a credential-focused system gives us a good starting point for achieving a universal user-centric FIM system. Section 4, we proceed by looking at a concrete credential-focused FIM system and its properties.

3.3 Beyond User-Centricity

Currently, we already perceive several viable solutions for user-centric FIM in the real world. They have in common that they follow the paradigm to put the user in control of her identity and to involve the user's client in all transactions. Though we generally agree that user-centricity is a good paradigm, we perceive an inherent downside introduced by it.

That is, a user may desire to execute her user control over her identity by giving up a part of her immediate control and delegate explicit permissions to other entities as a willful act. This may happen because of efficiency reasons (for instance with small devices), organizational or convenience reasons. Therefore, we perceive *Delegation* as distinguished desired property that may move a user-centric FIM system beyond its inherent boundaries and allow a user to leverage advantages of non-user-centric systems as well. Obviously, *Delegation* is *meta* to the user-centric paradigm as it allows to move beyond its problem space and, therefore, introduces a solution that goes beyond the problem space of user centricity.

How does such a delegation from a user-centric FIM system look like in a simple example scenario?

EXAMPLE 1. *A user called Alice prefers to go to a hospital called Health-Central for various types of health screenings. Several times she has to get certain health examinations and tests done elsewhere, the results of which are required for deducing more comprehensive results of the health screenings itself. Due to the laws and regulations most of the health examination centers do not reveal this data to anyone but the user whose data is processed. Therefore Alice herself has to retrieve the data required by Health-Central every time it is needed. Since this is cumbersome, Alice would like a way to allow Health-Central to retrieve such data from the various centers. This is more efficient, and furthermore such capability is vital to handle cases of emergency.*

This example illustrates where a user-centric FIM system is essential because of the privacy of the personal health information, however, there is an evident need of delegation capabilities in such a system.

Within the next paragraphs we elaborate how user-centric FIM systems may achieve delegation and which steps have already been done in prior research. We again rely on our classification into relationship-focused and credential-focused systems to structure the discussion.

3.3.1 Delegation Background

A well-accepted definition of *Delegation* is that a principal or group of principals is explicitly appointed to represent another principal or group of principals. For an user-centric FIM system that handles not only authentication but also attribute statements this notion needs to be extended: here, a principal is appointed to represent another possibly anonymous principal with certain attributes. We focus our discussion to the case that the delegating principal is a user.

Delegation has been explored extensively in trust management systems for public-key infrastructures like PolicyMaker [6], KeyNote [5] and RT [27]. In most of these systems, the user herself can delegate some of her authority to a known delegatee. The process of making access control decisions involves finding a delegation chain from the source of authority to the requester. Thus, a central problem in trust management is to determine whether such a chain exists and, if so, to find it. This was named credential chain discovery problem [27].

3.3.2 Delegation in Relationship-Focused Systems

In most of the FIM systems, the user does not have the capability to re-issue tokens issued to her—and to enable this capability is non-trivial. This is because the credentials considered in trust management have more complex semantics and structure as the tokens considered in FIM systems. In relationship-focused systems, delegation could potentially be implemented based on a delegation policy defined by the user and given to the IdP. The IdP will then govern which privileges and delegation rules to apply at any given context. This is possible because each time a credential is used the appropriate IdP is consulted.

3.3.3 Delegation in Credential-Focused Systems

A similar reasoning can be applied when trying to execute simple delegation in credential-focused FIM systems. However, considering the specific properties of anonymity resulting from unlinkability and minimal data disclosure and various anonymity revocation capabilities in a multi-party transaction, the problem of delegation becomes non-intuitive and complex.

In general, we find two variants of delegation in credential-focused systems. One requires the presence of the identity provider at delegation time and, therefore, generates a similar flow as the relationship-focused systems. Though having the same structure as relationship-focused delegation, the credential-focused solutions maintain the positive properties of this class (*Data Minimization*, *Anonymity*, *Integrity*, *Sharing Prevention*), while adding additional functionality such as k-times use delegations.

The second class allows a user to create a delegation credential by herself without involving the identity provider in the delegation. To do that while maintaining the properties of the credential-focused FIM systems is an open research problem that was defined as cryptographic problem by Chase and Lysyanskaya [16]. There exists no (efficient) solution to this problem, yet.

In general, we perceive it as valuable to user centricity to continue the research on delegation, in particular considering the open problems in credential-focused systems. To our judgment, research may provide powerful tools for moving beyond the boundaries of this paradigm.

4. MECHANISMS

This section discusses mechanisms that can be used to obtain an identity management system with given properties of our taxonomy of Section 2. We structure our discussions by the weak and strong trust models we presented in Section 2 as they lead to major differences in requirements on mechanisms. Furthermore, we structure the section by picking composite properties and elaborate on those and the properties required or useful for obtaining them. We chose a selection of properties from the taxonomy with a focus on privacy.

4.1 Weak Trust Model

The weak trust model, that is, assuming a fully trusted issuer, allows for the construction of powerful identity management protocols in the relationship-centric domain and ones based on traditional certificates. Due to the weak trust model the protocols can—despite their power—be constructed based on standard cryptographic techniques such as signature schemes like DSA or RSA and (hybrid) encryption schemes using prudent engineering practices to devise correct protocols [2].

4.1.1 Attribute Security

Confidentiality of identity information is mainly accounted for by using a secure channel to the intended recipient and to the identity provider to preclude undesired exposure of identity information to unauthorized parties. An additional means for ensuring confidentiality is to have attribute information secured appropriately in storage on a user's devices. *Integrity* is achieved by the identity provider issuing tokens that are signed using a traditional signature scheme like DSA [32] or RSA [36]. This allows for directing the tokens to the relying party via the user while preventing the user from applying unauthorized changes to the identity information. Understanding the semantics of the security tokens that a user receives from an identity provider allows for verifiability of attribute correctness.[5] For the *revocation* of identity information, certificate revocation lists (CRLs) [21] can be employed in the case of the traditional certificate-centric approach. In the relationship-centric approach, revocation can be easily achieved by the identity provider keeping state of the revoked parties and eventually not issuing further tokens.

4.1.2 Anonymity

Unlinkability is achieved by having the identity provider issue a fresh token for each transaction the user carries out. This approach is taken in the relationship-centric systems. Unless the identity information in the tokens establishes linkability, the tokens used in multiple transactions can remain unlinkable using the assumed weak trust model of a fully trusted identity provider. Unlinkability cannot be achieved in case of a system with a user having long-time credentials based on traditional signature schemes such as RSA or DSA as the credential has to be disclosed in each transaction thus making all the transactions linkable. *Selective release* builds on the very same idea of the identity provider issuing fresh tokens for every transaction containing precisely the identity information as required by the relying party. For *conditional release*, the identity provider can provide an encryption of to be conditionally released data under the key of a trusted decryption authority or the identity provider guarantees to provide the identity information to be conditionally released once the agreed condition is met. Again, this is made possible by the weak trust model that assumes having an honest identity provider.

4.1.3 Accountability

Context-bound transactions can be realized by signing the transaction context using a traditional signature scheme. This could be done by either the user or the identity provider, depending on what kind of system is being used. One can obtain *non-repudiation* by using traditional digital signature schemes where the identity of the signer is bound to the signature verification key, e.g., via PKI. Furthermore, non-repudiation requires *stealing protection*. This can be achieved by appropriately securing the computing environment or relevant parts of it against malicious viruses. Running the critical subsystems in a separate hypervised operating system addresses the issue. Binding credentials or secret keys to hardware, e.g. by having them on a separate token like a smart card, can help obtain appropriate solutions as well.

4.2 Strong Trust Model

In the strong trust model it is much harder to obtain the integrity property while still allowing for properties like unlinkability, anonymity, selective release, and conditional release. For this reason we shall elaborate on what the problems are and how they are overcome. The basic idea is to deviate from the traditional signature schemes to achieve integrity and use special signature schemes and protocols. The basic idea behind those special signature schemes is that they allow the user to break the linkability between the certificate (credential) she receives from the identity provider and the token she sends to a relying party. Breaking the linkability means that the bit string of the issued certificate and the token provided to the relying party cannot be linked to each other by an adversary controlling both the identity provider and the relying party. A prominent example for such a scheme is the scheme by Brands [7] which features single-use certificates only, that is, certificates may only be used once if unlinkability is to be retained. Two more recent schemes by Camenisch and Lysyanskaya [10, 12] feature multi-show certificates, that is, unlinkability can be retained even if the same certificate is used multiple times. In the case of using the signature schemes of Camenisch and Lysyanskaya, the issuer can be offline after having issued a certificate, that is, need not be involved in the transaction for providing user identity information to a relying party, provided that certificate revocation is still accounted for.

4.2.1 Attribute Security

Integrity is achieved by having identity information certified by the identity provider using one of the special signature schemes outlined above. *Verifiability* is analogous to the weak model by requiring the user to understand the semantics of the credentials (certificates) and check them for correctness. *Revocability* is harder to achieve in the strong trust model due to the unlinkability of transactions, that is, a straightforward matching of bit representations of certificates on revocation lists cannot work. A mechanism that solves the problem are dynamic accumulators [11], that help realize certificate revocation in a setting using the special signature schemes. Sharing prevention can, for example, be achieved by either or a combination of the following mechanisms: Binding credentials to a master secret key of the user and mandating appropriate protection of this key; binding credentials to the hardware directly or to hardware-bound credentials and never releasing them

[5] This property is only required in the strong trust model.

[8, 9]; using the concept of all-or-nothing non-transferability that shares all of a user's credentials once one credential is shared [11].

4.2.2 Anonymity

The *unlinkability* is accounted for by the use of the special signature schemes and their properties of obtaining of a certificate being unlinkable to its use. Brands' scheme and Camenisch and Lysyanskaya's schemes in addition allow that the user can *selectively release* information of the attributes of a certificate in a transaction by using zero-knowledge proofs of knowledge. *Conditional release* is rather hard to achieve in the anonymous setting without involving the identity provider in the transaction: The mechanism of verifiable encryption [13] makes it possible for a user to encrypt attribute information from a certificate and provide a proof that the corresponding plaintext actually is the attribute of her certificate. In addition, a decryption condition can be bound to it as a label of the encryption.

4.2.3 Accountability

Achieving *accountability* in a setting with unlinkability and anonymity in a strong trust model with a dishonest identity provider is much harder than in a weak trust model. Though, it is possible to achieve it with non-standard cryptographic mechanisms. *Context-bound transactions* can be achieved by using the Fiat-Shamir heuristic for signing contextual information of a transaction and thus binding it to the transaction. *Conditional release* realized by verifiable encryption (see e.g., [13]) provides for the possibility to allow recovering identity information of the message originator in case a defined condition is fulfilled. This does not involve the identity provider, yet gives the relying party assurance of the user's identity being recoverable.

5. CONCLUSION

We contributed to the notion of user-centric identity management by i) a taxonomy unifying today's notions, ii) a discussion of the differences of the two predominant paradigms, and iii) a discussion on mechanisms useful for accomplishing properties described in our taxonomy. We investigated the idea of how we can have a universal user-centric system, incorporating the advantages of various current approaches. Moreover, we highlighted some limitations of user centricity, and examined how we can go beyond the notion of user centricity to achieve additional desired properties. The particular drawback we identified in pure user-centric systems is the lack of delegation capabilities due to the inherent user-in-the-middle aspect. The main open research question we raise is the search for credential-based user-centric systems that cross the boundaries of user centricity and allow for delegations if requested by the user. We suggest that our approach in unifying the notions in user-centricity may be useful for the field of user-centric federated identity management systems.

6. REFERENCES

[1] Introduction to usability, 2005. `http://www.usabilityfirst.com/intro/index.txl`.

[2] Abadi, M., and Needham, R. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering 22*, 1 (1996), 6–15.

[3] Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. *Enterprise Privacy Authorization Language (EPAL 1.1)*, 2003.

[4] Bettini, C., Jajodia, S., Wang, X. S., and Wijesekera, D. Provisions and obligations in policy rule management. *J. Netw. Syst. Manage. 11*, 3 (2003), 351–372.

[5] Blaze, M., Feigenbaum, J., and Keromytis, A. D. KeyNote: Trust management for public-key infrastructures (position paper). *Lecture Notes in Computer Science 1550* (1999), 59–63.

[6] Blaze, M., Feigenbaum, J., and Lacy, J. Decentralized trust management. Tech. Rep. 96-17, 28, 1996.

[7] Brands, S. *Rethinking Public Key Infrastructure and Digital Certificates— Building in Privacy*. PhD thesis, Eindhoven Institute of Technology, Eindhoven, The Netherlands, 1999.

[8] Brickell, E., Camenisch, J., and Chen, L. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security* (New York, NY, USA, 2004), ACM Press, pp. 132–145.

[9] Camenisch, J. Protecting (anonymous) credentials with the trusted computing group's trusted platform modules v1.2. In *Proceedings of the 21st IFIP International Information Security Conference (SEC 2006)* (2006).

[10] Camenisch, J., and Lysyanskaya, A. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In *Advances in Cryptology — EUROCRYPT 2001* (2001), B. Pfitzmann, Ed., vol. 2045 of *LNCS*, Springer Verlag, pp. 93–118.

[11] Camenisch, J., and Lysyanskaya, A. Dynamic accumulators and application to efficient revocation of anonymous credentials. In *Advances in Cryptology — CRYPTO 2002* (2002), M. Yung, Ed., vol. 2442 of *LNCS*, Springer Verlag, pp. 61–76.

[12] Camenisch, J., and Lysyanskaya, A. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology — CRYPTO 2004* (2004), LNCS, Springer Verlag.

[13] Camenisch, J., and Shoup, V. Practical verifiable encryption and decryption of discrete logarithms. In *Advances in Cryptology — CRYPTO 2003* (2003), D. Boneh, Ed., LNCS.

[14] Camenisch, J., Sommer, D., and Zimmermann, R. A general certification framework with applications to privacy-enhancing certificate infrastructures. In *Proceedings of the 21st IFIP International Information Security Conference* (2006).

[15] Cameron, K. Laws of identity, 5/12/2005.

[16] Chase, M., and Lysyanskaya, A. On signatures of knowledge. Cryptology ePrint Archive, Report 2006/184, 2006.

[17] Cranor, L., Langheinrich, M., Marchiori, M., Presler-Marshall, M., and Reagle, J. *The Platform for Privacy Preferences 1.0 (P3P1.0) Specification*.

[18] European Parliament. Directive 95/46/ec of the european parliament and the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Communities* (1995).

[19] Hall, R. S., Heimbigner, D., and Wolf, A. L. A cooperative approach to support software deployment using the software dock. In *ICSE '99: Proceedings of the 21st international conference on Software engineering* (Los Alamitos, CA, USA, 1999), IEEE Computer Society Press, pp. 174–183.

[20] Higgins Trust Framework, 2006. `http://www.eclipse.org/higgins/`.

[21] Housley, R., Polk, W., Ford, W., and Solo, D. RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Apr. 2002. Status: Informational.

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

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

[24] J. Merrels, SXIP Identity. DIX: Digital Identity Exchange Protocol. Internet Draft, March 2006.

[25] Kaler, C., and Nadalin, A. Web services federation language, 2003.

[26] Kaler, C., and Nadalin, A. Ws-federation: Passive requestor profile, 2003. Available from: `ftp://www6.software.ibm.com/software/developer/library/ws-fedpass.pdf`.

[27] Li, N., Winsborough, W. H., and Mitchell, J. C. Distributed credential chain discovery in trust management: extended abstract. In *ACM Conference on Computer and Communications Security* (2001), pp. 156–165.

[28] Liberty Alliance. Liberty alliance id-ff 1.2 specifications. Available at `http://www.projectliberty.org`.

[29] Lüer, C., and van der Hoek, A. Jploy: User-centric deployment support in a component platform.

[30] Lysyanskaya, A., Rivest, R., Sahai, A., and Wolf, S. Pseudonym systems. In *Selected Areas in Cryptography* (1999), H. Heys and C. Adams, Eds., vol. 1758 of *LNCS*, Springer Verlag.

[31] Microsoft. A technical reference for InfoCard v1.0 in windows, 2005.

[32] National Institute for Standards and Technology (NIST). Digital signature standard (dss), 2000.

[33] OASIS Standard. Security assertion markup language (SAML) V2.0, 2005.

[34] OECD. OECD guidelines on the protection of privacy and transborder flows of personal data, 1980.

[35] PRIME Consortium. Privacy and Identity Management for Europe (PRIME). Web site at `www.prime-project.eu`.

[36] Rivest, R., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*, 2 (Feb. 1978), 120–126.