USER CENTRICITY: A TAXONOMY AND OPEN ISSUES

by Abhilasha Bhargav-Spantzel and Jan Camenisch and Thomas Gross and Dieter Sommer

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

User Centricity: A Taxonomy and Open Issues [1]

Abhilasha Bhargav-Spantzel [a], Jan Camenisch [b], Thomas Gross [b], and Dieter Sommer [b]

[a] *Department of Computer Science, Purdue University*
[b] *IBM Zürich Research Lab, Switzerland*

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. We highlight the various mechanisms to achieve the properties identified in the taxonomy. We show how these mechanisms may differ based on the underlying technologies which in turn result in different trust assumptions. We classify the technologies into two predominant variants of user-centric FIM systems 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 former encompasses cryptographic credentials as defined by Lysyanskaya et al. [37], and the latter encompasses 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.

Keywords. Identity Management, 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

[1] 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.

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 types of entities, namely the *user*, *identity provider* (IdP) and *service provider*. The IdPs manage and provide user identities and potentially issue user credentials, and the service providers (also known as *relying parties*) are entities that provide services to users based on their identity (attributes). A FIM system is characterized by a user having identity information certified by multiple IdPs. A relying party may accept identities from multiple IdPs, according to their policy. That is, the FIM system puts no restrictions on the identity provisioning relationships between IdPs and relying parties. An important aspect of FIM systems is that the various components of management are distributed amongst multiple policy domains. We have a more general notion of FIM in that user profiles at different IdPs need not be linked to each other by the IdPs as in the traditional notion of FIM.

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 [47] 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 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 IdPs and thus each transaction conveying identity information to a service provider involves the appropriate IdP. 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 IdP and storing them locally. These credentials can then be used to provide identity information without involving the IdP. 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 IdP (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 IdP (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. This is analogous to checking the revocation status in a credential-centric system. Note that whether revocation is to be performed in a credential-centric system entirely depends on the policy of the relying party.

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 parties.

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 [32, 29,28]. 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 [38], SXIP [31] or the open-source Higgins Trust Framework [26]. 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

Currently there is no unified understanding of user centricity as people refer to it for either of the different notions of user centricity as outlined further above. 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 dealing with the various properties of an ideal user-centric system and describe its various aspects.

As a second contribution, in Section 3 we discuss the technical mechanisms that can be used to construct identity management systems that satisfy a given set of user-centric properties of our taxonomy. We distinguish among three different types of systems based on the technologies, namely, traditional or *standard certificate* systems, *current FIM* systems and *anonymous credential systems*. We show that achieving the properties in these systems need different types of mechanisms and hence different trust requirements. We also distinguish between a weak and a strong trust model and put a focus on mechanisms that help realize better user privacy.

In Section 4, we provide, as a third contribution, a detailed discussion of the predominant paradigms of user-centric systems. We group the above identified technologies within the two significant paradigms of user-centric systems, namely relationship-focused and credential-focused systems. 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 current 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. This is followed by some conclusions in Section 5.

2. Properties for User Control in User Centric FIM

Realizing a user-centric identity system concerns several distinct properties. A key property of a user-centric FIM system is that of *user control* for which we provide a comprehensive and detailed analysis. While reasoning about the security and privacy properties of user control, we refer to the OECD principles [42]. The OECD guidelines are widely accepted and form the cornerstone of fair information practices and regulations designed

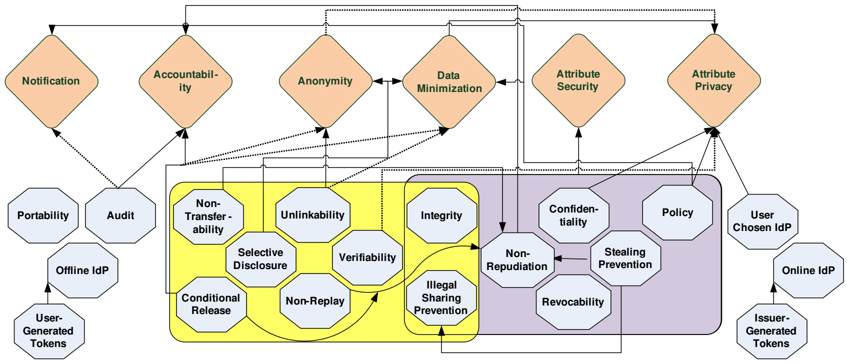

Figure 1. Taxonomy of User Control Properties of User-Centric Identity Management System

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* [17] 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.

Based on the above principles we elaborate on the user control aspect of a user-centric FIM system. 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 *high level properties* in that they are realized or composed on top of other properties, while others are *basic properties*, which provide the basis for other high level properties. The properties of our taxonomy related to user control are illustrated as nodes of the directed graph (more precisely, directed forest) in Figure 1. The high level properties at the top of the graph, denoted by shaded circles, tend to be more general and may depend on several basic properties. The basic properties, denoted by clear circles represent some fundamental properties which help achieve the other properties. Strong dependance or requirement of one property to achieve another is depicted as a solid arrow. On the other hand weak dependence is where one property simply enhances or helps achieve the second property and is denoted by a dotted arrow. These arrows however depend on the trust model assumptions which is elaborated further in Section 3.

The resulting properties of our taxonomy are based on a requirements-driven approach. We started with the basic security requirements that are expected from multiple FIM standards like Liberty ID-WSF and WS-*. Some examples include integrity, confidentiality and certain privacy properties. In addition, we extend the requirements based on established OECD principles [42] and the Laws of Identity [17]. Finally, we also added the requirements involving regulatory and business concerns which are crucial for practical deployments. For example, privacy and data minimization is to some extent driven by legislation, particularly in Europe. As a part of the exercise, we noticed that when developing novel mechanisms for achieving privacy and some critical properties in strong trust models (i.e. less assumption of trust on entities), achieving certain other properties becomes challenging. Based on all the above considerations, we developed a taxonomy of properties which can be extended to future needs.

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 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*. Our main focus is on user control properties which are elaborated as follows.

2.1. 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, the semantics of the properties would highlight which of the three they are relevant to.

Confidentiality. Confidentiality may be defined as the protection of sensitive information from unauthorized disclosure. This property applies to identity information and transactions in the system. This property 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. 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 IdP 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.

Revocability. 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.

Unlinkability. Unlinkability of two or more subjects or transactions—or generally, of items of interest—means that for the attacker the relations between the items of interest do not change based on his observations related to his a-priori knowledge on the relations. This means that the attacker, after having observed transactions, does not gain information on linking those transactions or linking subjects' accounts based on observed information. A typical attacker in our consideration is constituted by a collusion of multiple relying parties. Optionally, also IdPs are controlled by an attacker leading to a much stronger attacker model.

When considering that the attacker controls relying parties, then it is a prerequisite that the identity assertions being conveyed in the transaction do not establish linkability. For example, if multiple assertions contain a unique identifier referring to a particular user, then those transactions are trivially linkable by an attacker controlling the involved relying parties. As another example, consider an attribute certificate of a user that is provided to multiple relying parties. Then all those transactions would be linkable by an attacker controlling the relying parties due to the unique bit string provided in each of the transactions. We want to refer the reader to comprehensive work on terminology on privacy, anonymity, and unlinkability [45,44].

Ensuring unlinkability for an attacker that controls multiple relying parties is clearly necessary considering today's ownership structures where multiple companies are held by the same owners and mergers and acquisitions occur frequently. Unlinkability technically prevents (illegitimate) merging of profiles by linking them. If we take a more progressive approach, also IdPs could be possible points of failures due to external or internal attacks and thus should be seen as being controlled by the attacker.

Policy. Policy management, agreement, and enforcement 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 property *privacy policy* deals with the management of privacy policies defined in the system and 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 [1,20] 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.

Related concepts in privacy policies are obligations and restrictions. Obligations [2] 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, for example, to not share credentials with other users. 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.

Finally there are *access control policies* which are defined to authorize users to perform a set of actions on a set of resources. In FIM systems these resources can be the services provided by the relying party, and also user credentials or identity attributes.

User-Chosen IdP. The user-chosen IdP property 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.

Verifiability. We also define the verifiability property meaning that the user can verify that the IdP provides the correct identity data about the user and according to the user's intention. As such, a user giving her consent means that the user's view of the transaction 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.

Generated Tokens. 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*. In the case of user-generated tokens, the user should be able to construct tokens which can be verified as valid based on signed attributes present in other user credentials. In this case an *offline IdP* is sufficient as the IdP does not need to be contacted to generate the tokens. For other cases where the tokens are mainly constructed by just the IdP or the IdP together with the user, an *online IdP* is needed. Having an IdP offline may have better user privacy implications versus online IdP where the IdP may be required to be given some control of the users' transactions.

Illegal Sharing Prevention. 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.

Non-Transferability. The *non-transitivity* property addresses the impossibility for a recipient of identity information to reuse this identity information using the obtained security tokens. Note that the identity information itself can be reused in a typical setting as it becomes known by the recipient.

Non-Replay. Non-replay of messages of transactions helps in establishing stronger security guarantees within a system by preventing unauthorized parties to get authorized. Non-replay is one prerequisite for obtaining the non-repudiation property. The transactions providing for non-replay can potentially contain certain contextual elements of the transaction to guarantee freshness.

Non-Repudiation. The non-repudiation property of messages means that the sending of a non-repudiable message cannot be denied by its sender. 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.

Stealing Prevention. 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.

Selective Disclosure. Selective disclosure or release of identity information means that identity information can be released at a fine-granular level as controlled by the user. In this way a user can provide only the identity information that needs to be released for a service without having to leak additional information. Selective disclosure is a key property towards achieving both anonymity and data minimization.

Conditional Release. 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.

Audit. 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. Audit can be critical for compliance with legal requirements.

Portability. 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 of one user. This property may require intricate mechanisms depending on the identity management mechanisms and protocols being used.

2.2. High Level Properties

Accountability. 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 IdP. 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.

Notification. Notification is desired to enhance the control of the user so that she is able to receive and retrieve notifications regarding the usage of her credentials. This is also important when there is a security breach, and compromise of the user's identity information at an external entity, to which the user had provided these data. 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. The sixth and seventh OECD principles of openness and individual participation can potentially be satisfied using comprehensive notification mechanisms.

Anonymity. Anonymity in transactions deals with the subjects remaining anonymous within an 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 related to anonymity.

Note that 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.

Data Minimization. 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 appropriate 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 tocollection limitation. This principle is also reflected in the European Data Protection Directive 95/46/EC [23] and the national data protection laws within the European Union. It is to be noted that data minimization must not harm the attribute security. 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 [16] to have diverse and important implications for an identity management system.

Attribute Security. 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.

Attribute Privacy. 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 dependent properties very likely contribute towards attribute privacy in that they helps avoid the unnecessary release of (identifying) information. Data minimization also directly provides privacy. An orthogonal property essential for reaching attribute privacy is the support of privacy policy management, enforcement and agreement. 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.

2.3. Other User-Centric Aspects

Architecture and Deployment. Deployment consists of those activities that need to be performed with a software product after it has been released [25]. 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 [36] 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 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 IdP 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* may not make any assertions about the involvement of the human user, e.g., for approving every transaction. If the user is involved in providing the identity information to the service provider then this can be in two ways: 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.

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.

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 [21]. 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. Mechanisms

In this section we discuss (technical) mechanisms that can be used to obtain an identity management system with given properties from our taxonomy of Section 2. We refer to three different core mechanisms for user-centric identity federation and the associated trust models. These core mechanisms restrict the choice of complementary mechanisms to achieve certain properties of the resulting system.

For the further discussions we elaborate on the trust models for each of the three classes of systems. A trust model describes which parties need to be trusted and to what extent in order to achieve a particular property. For better readability, we summarize the trust requirements for each of the approaches to achieve a certain property in Table 1. Note that for some properties a dishonest party can be assumed as the user can detect deviations from the protocol, e.g., by inspecting the tokens she obtains. Successful completion of the protocol still requires that the party follows the protocol in such cases. We note that the intention of this section is not to give a complete survey of mechanisms, but to focus on the more interesting and prominently used ones.

Table 1. Comparison of the Three Classes of Mechanisms for User-Centric Identity Management.
Notation: r: Honest Relying Party; R: Semi-Honest Relying Party; $\underline{R}$: Adversarial Relying Party; p: Honest Identity Provider; P: Semi-Honest Identity Provider; $\underline{P}$: Adversarial Identity Provider;

Property	**Traditional Certificate System**	**FIM System**	**Anonymous Credential System**
Integrity	$\underline{R}$+P	$\underline{R}$+P	$\underline{R}$+P
Sharing Prevention	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$
Stealing Prevention	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$
Revocability	$\underline{R}$+P	$\underline{R}$+P	$\underline{R}$+P
Non-Transferability	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$
Verifiability	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$	$\underline{R}$+$\underline{P}$
Conditional Release	–	$\underline{R}$+p	$\underline{R}$+$\underline{P}$
Selective Disclosure	–	$\underline{R}$+p	$\underline{R}$+$\underline{P}$
Unlinkability	–	$\underline{R}$+p	$\underline{R}$+$\underline{P}$
Policy Enforcement	r+p	r+p	r+p
User-Generated Token	$\underline{R}$+$\underline{P}$	–	$\underline{R}$+$\underline{P}$
Issuer-Generated Token	$\underline{R}$+P	$\underline{R}$+P	$\underline{R}$+$\underline{P}$

3.1. Standard Certificates

Standard certificates, like X.509 certificates [30], allow—in conjunction with a private signing key—a user to prove that attributes have been issued to her. A certificate contains attributes and a public key signed by the IdP (the issuer of the certificate).

To assert the attributes of a certificate to a relying party, the user engages into a challenge-response protocol with the relying party. This protocol requires the certificate to be sent to the relying party and a signature being made with the private key. This reveals all attributes of the certificate to the recipient of the attributes as always the complete certificate is revealed. Technically, standard certificates are based on standard digital signature schemes such as RSA [49] or DSA [40]. Standards like X.509 [30,18] define the formats of the certificates.

Traditional-certificate-based technologies allow for constructing systems where a certificate is issued once and can be used by users arbitrarily often to reveal the attributes contained in the certificate. Thus, this technology allows for off-line IdPs. The tokens are generated by the user without involvement of the IdP, this making this method flexible with respect to this aspect. This technology is, for example, used in multiple ID-Card proposals.

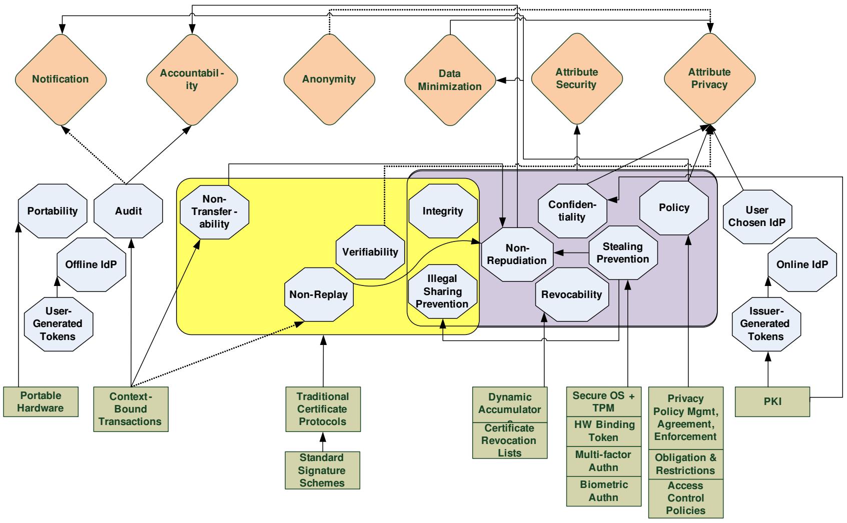

Figure 2. Mechanisms for Achieving User Control Properties in the Traditional Certificate System

3.1.1. Attribute Security

Integrity in such schemes is accounted for by the user attributes being included in the certificate that is signed by the IdP using standard signature schemes and the certificate being provided each time attributes are asserted to a relying party.

Confidentiality of attribute information is achieved by using encryption schemes in conjunction with public key infrastructure (PKI).

Stealing prevention for standard certificate systems targets at protecting the master private key as the certificates are made available to relying parties anyway. The following mechanisms can be used, also in a combined fashion: (1) Binding all certificates to one master private key of the user and mandating appropriate protection of this key, for example in a hardware token. (2) Keeping the master private key in a secure hardware token. This involves the hardware token in each transaction. Therefore, either the hardware token be portable (smartcard) or that the credentials be portable between multiple platforms controlled by the user. (3) Applying operating system mechanisms to prevent a user from sharing their key. (4) Multi-factor authentication makes it harder to share the token, e.g., if it is derived from biometrics of the user.

Illegal sharing prevention can be achieved by the same mechanisms as stealing prevention. Note that sharing prevention is of less importance in such systems as all transactions are linkable and consequently may contain identifying attributes of the involved users. Thus owners of certificates will have incentives not to share their certificates with other people. Illegal sharing prevention is of higher importance in the case of anonymous credentials as shown in Section 3.3.

Revocability can be achieved by the prominent mechanism of certificate revocation lists (CRLs) and the associated protocols [27]. This requires an additional protocol to be run in order to obtain the latest revocation list.

Non-transferability can be achieved by using a challenge-response protocol for proving ownership of a certificate or by adding information on the intended audience to the protocol. This context information renders the certificate useless outside the intended transaction context. Similarly, *non-replay* can be obtained using challenge-response protocols or by restricting information on the transaction context.

3.1.2. Attribute Privacy, Anonymity, and Data Minimization

Verifiability holds as a user can inspect the certificate and thus has control over the attribute information being revealed.

Conditional release cannot be realized in the setting in which the protocols operate, as an IdP is not involved in transactions. Technically, of course, protocols could be conceived that involve the IdP in a transaction to obtain the conditional release property, but by this we would leave the basic paradigm of the system.

Selective disclosure is not possible in the setting of using standard certificates as certificates always have to be revealed as a whole and no subset of their attributes can be revealed because of the properties of the standard signature schemes like RSA or DSA being employed.

Finally *unlinkability* cannot also be achieved in this setting. This is because, transactions done with multiple IdPs or multiple transactions with one IdP are linkable as of the same certificate bit string being provided in every transaction.

3.1.3. Accountability

Accountability is mainly built upon the non-repudiation and audit properties. *Non-repudiation* of transactions can be accomplished by a combination of various resultant properties and corresponding mechanisms as shown in Section 2.

3.1.4. Other Properties

The *user-generated tokens* property is achieved by construction as the IdP is not involved in a transaction. This, at the same time, gives the *offline-issuer property*, as the IdP can be offline during the user's transaction as long as revocation is accounted for.

Portability can be achieved by having both private key(s) and certificates on a portable hardware token.

3.1.5. Discussion

The main problems with the approach of using standard user-side certificates are the lack of overall privacy properties and thus the strong trust assumptions that we have to make on the relying parties. Assuming stronger trust in a relying party than it being ‘honest but curious’ for all its duties is quite unrealistic as in practice relying parties are often interested in getting hold of as much user data as possible.

3.2. Federated Identity Management with Online Issuing of Tokens

Current FIM protocols with short-term tokens address some of the weaknesses in terms of privacy in the abovementioned approach of standard certificates. In these relationship-centric FIM systems, the IdP does not issue long-term certificates to the users, but users only establish a relationship with the IdP. Based on this relationship, the IdP issues a short-lived federation token when the user engages in a transaction with a relying party. The federation token typically is a signed token relayed to the relying party over the user’s machine. This approach requires the IdP to be online at the time of each transaction with a relying party and also to be involved in the transaction.

See Table 1 for the trust requirements in the relying party and IdP for those FIM systems. We can observe that some privacy properties become achievable in such systems in contrast to the standard-certificate-based systems.

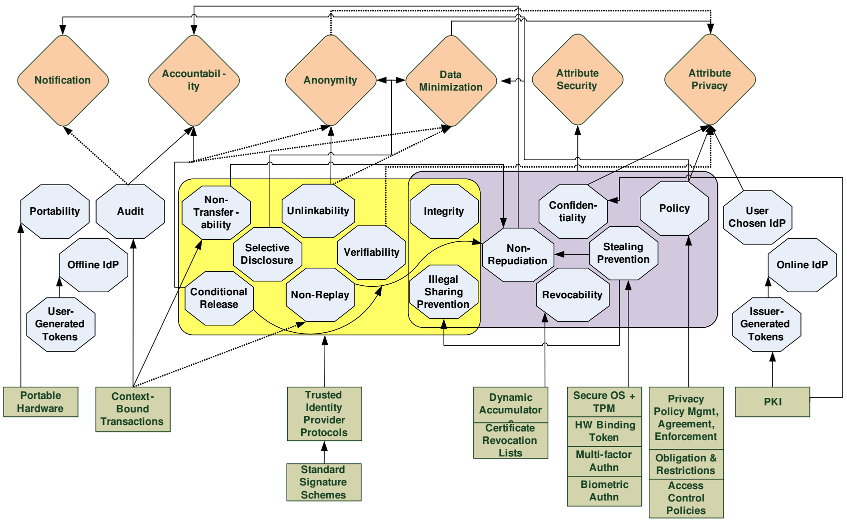

Figure 3. Mechanisms for Achieving User Control Properties in the FIM System

3.2.1. Attribute Security

Integrity is achieved by the IdP issuing a fresh token that is signed using a standard signature scheme like DSA [40] or RSA [49]. Being signed allows for directing the tokens to the relying party via the user while preventing the user from applying unauthorized changes to the identity information, thus retaining integrity of the conveyed identity information.

Confidentiality of identity information is mainly accounted for by using a secure channel to the relying party and to the IdP 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.

In the case of username/password authentication, it is quite hard to allow for *stealing protection* for the username/password tuple when considering attacks like key loggers, visual espionage, and others. In case of using authentication with the IdP that is based on a user's certificate and private key, the mechanisms from Section 3.1 can be employed, thus providing substantial protection against sharing.

In case the authentication of the user is by means of username/password tuples, it is generally hard to achieve *illegal sharing prevention* of their relationships with IdPs with others as passwords are easily sharable. Massive-scale sharing is easily detectable by the IdP, though, as it is involved in each transaction. The risk can be mitigated by operating system security mechanisms that prevent malware from obtaining passwords. In case of public key authentication, again the mechanisms from Section 3.1 can be utilized.

The *revocation* of identity information can be easily achieved by the IdP maintaining state of the revoked parties and eventually not issuing further tokens. Thus the paradigm of online-issued tokens allows for incorporating revocation features with very little effort and no changes to the flows.

Non-transferability can be achieved by the IdP including an audience field into the signed token with the semantics that only parties listed in the audience are intended recipients of the token. Having a very restricted audience, like only the relying party, solves the problem of non-authorized transfer of the token by the recipient to other parties.

Non-replay is harder to achieve in this model as the protocol does not have challenge-response characteristics. However, timestamps and state maintained by the relying party can help prevent replay of tokens by unauthorized parties.

3.2.2. Attribute Privacy

Understanding the semantics of the security tokens that a user receives from an IdP to forward to a relying party allows for *verifiability* of attribute correctness.

Unlinkability is achieved by having the IdP issue a fresh token for each transaction the user carries out. This approach is taken in the relationship-centric systems (see Section 4). 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 IdP. In case of identifying data being released, we have to fall back to a completely trusted relying party.

Selective release builds on the very same idea of the IdP issuing fresh tokens for every transaction containing precisely the identity information as required by the relying party. This property holds even if the IdP is dishonest as the user could detect this behavior and act accordingly.

For *conditional release*, the IdP can provide an encryption of data that is to be conditionally released under the key of a trusted decryption authority or the IdP guarantees to provide the identity information to be released once the agreed condition is met. Again, this is made possible by the weak trust model that assumes having a completely honest IdP.

3.2.3. 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 IdP, depending on what kind of setup 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.

3.2.4. Other Properties

The *issuer-generated tokens* property result from the inherent property of the system based on the IdP issuing a new token for each of a user's transactions. This property goes hand-in-hand with the *online IdP* property.

3.2.5. Discussion

There are several existing FIM systems which provide the above described attribute security, privacy, and accountability properties to varied degrees, in addition to the basic attribute provisioning functionality provided by them. A list of such systems each with a brief description is provided in Table 2.

Liberty Alliance [28]	The Liberty Alliance is a consortium of approximately 150 companies that develops specifications for federated identity management. It released the first version of its Liberty Web Services Framework in 2003 which allowed single sign-on and account linking between trusted partners.
Shibboleth [29]	Shibboleth is standards-based, open source middleware software which provides Web Single SignOn (SSO) across or within organizational boundaries. It allows sites to make informed authorization decisions for individual access of protected online resources in a privacy-preserving manner.
Microsoft CardSpace [38]	Windows CardSpace, formerly known as InfoCard, is a framework developed by Microsoft which securely stores pointers to digital identities of a person, and provides a unified interface for choosing the identity for a particular transaction, such as logging in to a website or accessing some web service.
OpenID [43]	OpenID is a decentralized identity system, in which any user's online identity is given by URL (such as for a blog or a home page) and can be verified by any server running the protocol.
Microsoft, IBM, and WS* [50]	In April 2002, Microsoft and IBM published a joint whitepaper outlining a roadmap for developing a set of Web service security specifications. Their first jointly-developed specification, WS-Security, offers a mechanism for attaching security tokens to messages, including tokens related to identity.

Table 2. Initiatives and projects of current FIM systems.

Despite the improvements in terms of privacy when compared to systems based on standard certificates, FIM systems are still not the best one can achieve with respect to privacy. The main drawbacks still are that the IdP needs to be completely trusted in order to preserve anonymity and unlinkability and also that the IdP needs to be always online and issue a signed token for each transaction of a user.

3.3. Anonymous Credential Systems

Anonymous credential systems allow users to obtain anonymous credentials and later use credentials to assert partial attribute information as contained in the credentials to relying parties. Anonymous credential systems allow for obtaining properties like anonymity and unlinkability in a stronger trust model than the one of FIM systems. More precisely, the IdPs and relying parties are assumed to be both controlled by the attacker, that is, they may be arbitrarily dishonest. This model only holds for transactions where the user does not need to provide identifying data to relying parties. See Table 1 for details regarding the required trust for achieving certain properties. This strong model makes sense in practical use cases where non-identifying identity data, like a proof of the user's age being greater than or equal to 18 years, or a proof of the user being citizen of a European Union member state are required by the relying party.

Though, many transactions in practice require users to reveal identifying data to a relying party in order to consume a service. In this case, user privacy can be compromised under this strong trust model as the identifying data allows that transactions can be linked by the attacker, and extensive user profiles can be built. In this case, we have to fall back to stronger trust assumptions, that is, a weaker trust model of fully trusted relying parties, and rely on appropriate enforcement of agreed privacy policies in order to account for user privacy. This is never avoidable in case identifying data are needed to be released to other parties.

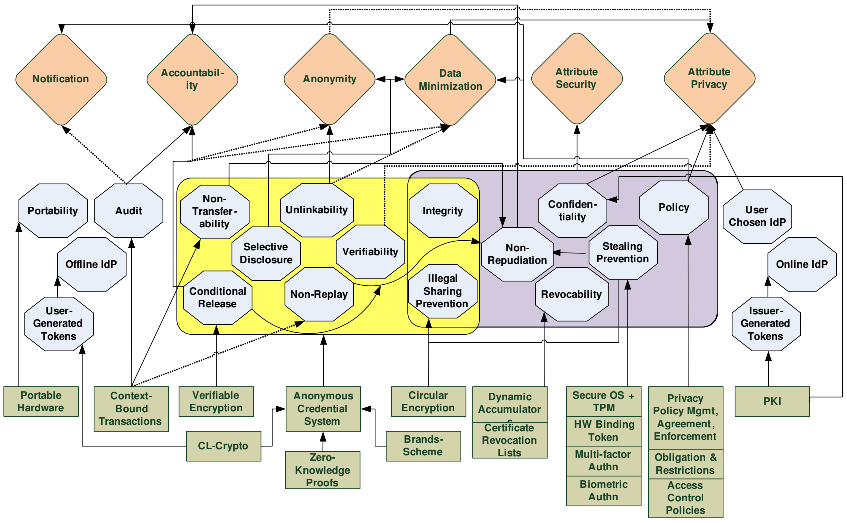

Figure 4. Mechanisms for Achieving User Control Properties in Anonymous Credential Systems

3.3.1. Attribute Security

In the setting of anonymous credential systems it is much harder to obtain integrity while at the same time providing for properties like unlinkability, anonymity, selective release, and conditional release. Achieving integrity and those other properties at the same time requires the orchestrated use of different (cryptographic) mechanisms as elaborated below. In particular, integrity cannot be achieved using traditional signature schemes like RSA [49] or DSA [40] as the unlinkability property is to be obtained as well as protecting the integrity of identity data with such a signature immediately implies that the IdP and relying party or different relying parties obtain the same signature bit string, thus violating the unlinkability property.

Thus, the basic idea is to deviate from those traditional signature schemes to achieve integrity and use special signature schemes and protocols. Those special signature schemes allow the user to break the linkability between the certificate (credential) she receives from the IdP and the token she sends to the 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 IdP and the relying party, that is, an adversary that can obtain all transaction transcripts of the parties it controls. A prominent example for such a scheme is the scheme by Brands [8] 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 [12,14], in addition, feature multi-show-unlinkable 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 in the architecture. We note that the more powerful systems require more computational effort for the execution of their protocols. Though, all of the systems are quite practical from this point of view.

*Verifiability*is easy to achieve when signature protocols are used as the user always generates the token to be sent to the relying party herself, thus gets to know the attribute information contained. Furthermore, credentials can be easily checked for appropriateness of attribute information when obtaining them. That is, the strong user control aspects of credential-based systems implicitly help us obtaining this property.

Revocability is much harder to achieve in the strong trust model than in the weak one due to the unlinkability of transactions. That is, a straightforward matching of bit representations of certificates or serial numbers of certificates with revocation lists cannot work. A mechanism that solves the problem can be realized by dynamic accumulators [13] together with appropriate zero-knowledge proofs. The resulting solution can be thought of as revocation lists in the setting of unlinkable transactions, that is, when using the special signature schemes. Whenever a user proves attributes using a private certificate, she makes a proof that a cryptographic value related to the certificate is not contained in the revocation list being used. This requires that both the user and the service provider obtain an updated "revocation list" as required. It is crucial to note that, although the user needs to obtain a sufficiently up-to-date revocation list, this need not involve the IdP, this could also be accounted for by other parties on behalf of the IdP, similarly to the setting of standard certificates. Particularly, the IdP need not issue new signatures for each transaction of a user, the private key of the IdP is only required whenever a certificate gets revoked.

Prominent mechanisms to achieve *theft protection* are similar to the ones in Section 3.1, but some new ideas come in. (1) Binding credentials to a master secret key of the user and mandating appropriate protection of this key. (2) Binding credentials to hardware, that is, storing the credentials or certain parts thereof in a hardware token such that they never leave the hardware. [9] (3) Binding credentials to hardware-bound credentials. This is, for example, described in detail in [10] for the Trusted Platform Module (TPM) that is deployed widely in today's computing platforms. (4) Operating system security mechanisms can provide a strong means of protecting private information from being obtained by an attacker. (5) Deriving the master private key from the user's biometrics, as put forth by Bhargav-Spantzel et. al. [4]. This requires that credentials be bound to this master key. (6) In general, multi-factor authentication for access to credentials and private key material makes it harder for an attacker to gain access to the credential.

If an attacker can obtain credentials, the obtained attribute information possibly already allow for certain levels of reusing the stolen identity. Nevertheless, the mechanisms that link the credential to a hard-to-steal private item effectively prevent credentials from being reused in protocols if they get stolen.

Illegal sharing prevention can be based on similar mechanisms than stealing prevention. For the binding of credentials to a master private key, strong incentives for the user not to share her master private key should be defined, e.g., by binding valuable secrets to the master key. See Camenisch and Lysyanskaya [11] for details on how to accomplish the binding of certificates to a master key.

One additional mechanism worth mentioning is the concept of all-or-nothing non-transferability implying that all of a user's credentials are shared once the user shares one of her credentials [13]. However, this mechanism may be disadvantageous for stealing prevention.

Non-transferability can be achieved by either one or a combination of the following mechanisms: (1) The properties of interactive zero-knowledge proof systems immediately lead to non-transferability as they require interaction of a user who possesses her master secret key and credentials with the relying party in order for the relying party to accept the proof as valid. (2) The same arguments hold in case the Fiat-Shamir heuristic [24] is used in the protocols and an initial challenge from the relying party is used. The challenge from the relying party provides the required level of "interactiveness" of the protocol. (3) Restricting a token to a limited audience, that is, a set of intended recipients of the token, can help to limit illegitimate transfer to any other party not in the audience.

Non-Replay can, for example, be obtained by having a challenge/response protocol for the zero-knowledge proofs for proving properties of credentials.

3.3.2. Anonymity

A basic prerequisite for any kind of anonymity is network layer anonymity. For the internet protocol [46] this can be accomplished by, for example, the JAP [48] or TOR [22,39] *anonymizers*. The use of an anoymizer allows a user to establish a communication channel to an IdP or relying party without revealing any information like IP address.

On the protocol level of identity federation protocols, anonymity is based on multiple other properties as outlined in Section 2. On top of anonymized communication, the *unlinkability* is accounted for by the use of the special signature schemes and their properties of obtaining such a signature is 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. Though, only the signature protocols of Camenisch and Lysyanskaya allow for multi-show unlinkability, that is, using a signature multiple times with the same or different parties.

Obtaining the *conditional release* property in the anonymous setting without involving the IdP in the transaction requires the specialized cryptographic mechanisms of *verifiable encryption*[15]. Verifiable encryption 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.

3.3.3. Accountability

Conditional release realized by verifiable encryption (see e.g., [15]) 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, neither at the time of the transaction, nor at the time of recovering the conditionally released information. The mechanisms is based on the user encrypting the identity data with the public key of an agreed decryption authority and providing a proof of the correctness of the ciphertext. Additionally, a decryption condition is cryptographically bound to the ciphertext. Both is provided to the relying party and thus gives the relying party assurance of the user's identity being recoverable by engaging with the decryption authority. A decryption authority will decrypt an encryption provided to it using the condition provided by the relying party. Only if the condition matches the original one, the decryption will work. Checking the correctness of the condition is handled out of band and depends strongly on the application scenario.

Achieving *accountability* in a setting with unlinkability, anonymity, and data minimization in a strong trust model with a dishonest IdP is much harder than in a weak trust model. Though, it is possible to achieve it with the mechanisms for conditional release. *Context-bound transactions* can be realized by using the Fiat-Shamir heuristic [24] for signing contextual information of a transaction and thus binding it to the transaction.

Non-repudiation requires, in an anonymous setting, conditional release such that a transaction of a user can be non-repudiable while no identifying information is released unconditionally at the time of the transaction, but only conditionally. The advanced cryptographic mechanisms used for conditional release help achieve non-repudiability in anonymous transactions.

3.3.4. Other Properties

User-generated tokens are possible by the basic workings of anonymous credential systems. *Portability* can be achieved by binding private keys and anonymous credentials to secure hardware tokens. This allows users to use those cryptographic entities on multiple machines, but not simultaneously.

3.3.5. Discussion

A key open issue with anonymous credential systems is the delegation issue as discussed in Section 4. Anonymous credential systems require changes in existing architectures and the infrastructure itself. Finally, there are no real-world deployments available which would prove useful in analyzing and understanding such systems.

3.4. General Properties

We discuss the properties that are not directly related to or governed by the identity federation system in the remainder of this section.

The *user-chosen IdP* property relies on an architecture that is open for IdPs to join and that allows relying parties to choose IdPs of their choice. Such an open system allows the user to choose the IdP they trust most for particular attributes and thus adds privacy to the system by accounting for another aspect of user control.

The *policy management, agreement, and enforcement* property can be accounted for by a rather wide range of mechanisms in the policy space. We will cover this space by some prominent examples only. In the privacy policy space, a system can be built upon the P3P language and protocol [20] for a party announcing their privacy policy. In the space of access control, multiple alternatives are available. XACML offers an extensible language for access control. The language can be extended to account for certain requirements of anonymous credential systems, though not all requirements can be cleanly accomplished. An interesting proposal for a policy language that—with extensions—applicable to anonymous credential systems is the one proposed by Bonatti and Samarati [7]. This work actually offers a complete trust negotation framework, thus goes beyond what standard access control solutions can offer. For the space of privacy obligations, applied research has been done resulting in a proposal for handling privacy obligations in a general way [3]. However, a formal semantics is still missing in this work.

A key aspect of attribute security is the definition and enforcement of the policy that defines about the issuance of identity tokens. This encompasses checking the identity of users before issuing identity tokens. For example, it may be required by the issuance policy to check in-person the user's passport and drivers license before a particular IdP issues a certificate asserting the user's age.

Audit requires audit logs to be created with audit policies reflecting legal requirements. Automated audit log analysis can be in place to detect non compliance. The auditing system could support a privacy-controlled sharing of identity attributes and checking the harmonization of privacy policies in the federation environment. Policy harmonizations mechanisms make it possible to determine whether or not the transfer of identity attributes from one entity to another violate the privacy policies stated by the former. As such, secure audit-trail is required for tracking and reporting activity around confidential data.

3.5. Performance Considerations

The classes of identity management systems outlined above are based on quite different mechanisms to achieve their functionality leading to different performance characteristics. For example, the classes of protocols require different numbers of message exchanges with other parties. All the protocols require processing of messages, typically using XML for representing parts of messages. The number of required cryptographic operations greatly differs between different systems and ranges from a little to a substantial fraction of the runtime depending on what kind of identity management system we consider. Generally, better privacy requires more computation and results in an increase in message length.

For traditional certificate systems, both issuing of a certificate and using it to provide attributes to relying parties requires standard cryptography like generation and validation of digital signatures. FIM systems that convey federation messages via the user to the relying parties require one traditional digital signature to be generated and validated for each transaction. Though, FIM solutions require additional messages on the network due to contacting the IdP within the transaction. Anonymous credential systems have a larger overhead in terms of computation, but they are highly practical. For example, the idemix system requires only three multi-base exponentiations on the user side and one on the relying party side for a basic transaction, thus the system being within the efficiency of a standard certificate system. Additional features such as conditional release require a handful of additional exponentiations, though. Thus, the overall runtime of an anonymous credential transaction depends on the features used. Standard arithmetic techniques for optimization bring the performance of a basic credential transaction even closer to standard-certificate-based protocols. For example, a multi-base exponentiation can be computed much more efficiently than naively computing the product of the powers. An additional performance improvement results from the ability to precompute a substantial part of the user-side computations due to the zero-knowledge proofs being computed.

4. Beyond the Boundaries

Where Section 3 covers how the different types of underlying technologies achieve the user-centric properties of a FIM system, in this section, we move on to logical distinction of the existing paradigms and their limitations. In Section 4.1, 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 4.2 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 4.1. Such a system would be a perfect solution for user centricity, however, still remain within the limitations of the paradigm of user-centric identity management itself.

In Section 4.3, 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 propose that one key aspect for this solution will be efficient and flexible delegation of identity information and rights associated with them.

4.1. 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 3.

A *relationship-focused* system (Table 3, middle column) is characterized by the FIM system only managing relations to IdPs and collaboration partners. We call this design focus *relationship focused*. In such a system, the user's client queries an IdP, 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 [41], Liberty [35], or WS-Federation [32] tokens. Such a *security token* is usually a statement about a user's identity or attributes that is simply signed by the IdP. The *FIM system* described in Section 3.2 falls into this category.

In a *credential-focused system* (Table 3, 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 [8] or *idemix* [12]). Clearly the standard certificate systems (Section 3.1) and anonymous credential systems (Section 3.3) fall into this category. Since we showed that anonymous credential systems provide a significantly better solution than standard certificate-based systems with respect to the key user-centric properties, we focus on anonymous credential systems for the discussion related to credential-focused systems.

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. [37]. 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 IdP while retaining the certification by the IdP. 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.

Note that the use of either of the paradigms, relationship-focused or credential-focused systems, does not affect the user interface concepts – both can build upon the same concepts, yet use different underlying protocols. However, zero-footprint FIM protocols which allow only for user interfaces being provided via a webpage and not a local wallet may not provide consistent user experience throughout multiple IdPs and relying parties. We think that non-zero-footprint solutions provide better user experience as it is the same over all identity-related interactions of the user.

4.1.1. Relationship-Focused Systems

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 IdP 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

Table 3. 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

issued (*correctness*, *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 [41,33]). Also, they only need to rely on well-known public-key cryptography.

Disadvantages. By construction, relationship-focused FIM systems require the IdP to be online during transactions. This renders the IdP a single point of failure for these systems and imposes requirements of high uptime and quality of service on the IdPs (*availability*). Consequently, IdPs are costly to deploy and operate (*cost effective*, *efficiency*). Moreover, typically in such systems users have to securely manage a high-entropy password for each IdP which creates the well known manageability issues for average users.

As the IdP 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 (*data minimization*, *anonymity*).

Sharing and theft are still possible in relationship-focused systems.[2] 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 IdP 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[3] 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*),

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

[3] WS-Federation, SAML, and Liberty have in common that they allow an IdP 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.

that is, a principal acting as service provider misusing the credentials received in a valid protocol run.

4.1.2. Credential-Focused Systems

Advantages. By construction, the IdP 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 IdPs 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*).

Disadvantages. What could be seen as a general disadvantage of credential-centric systems is the need for users to manage their credentials themselves. 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 IdP 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 [12]. Credential-focused systems naturally generate a higher workload on a user's client and require a rich client to be executed.

4.2. 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.

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

4.2.2. *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 for 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 therefore conclude that a credential-focused system gives us a good starting point for achieving a universal user-centric FIM system.

4.3. *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 current approaches. If we consider a spectrum of user control with one end being completely user controlled (provided by credential focused) and the other relying on IdP-controlled transactions (provided by relationship focused), then a major disadvantage in both approaches is that they essentially achieve the extreme ends. It is desirable to achieve some middle ground. 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 the boundaries of current approaches and allow a user to leverage other advantages.

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.

4.3.1. 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 a 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 on the case where 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 [34]. 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 [34].

4.3.2. 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 not straightforward. 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 can be implemented based on a delegation policy defined by the user and given to the IdP, thus substantial extensions in the policy domain are required. 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.

To summarize, there are no hard-to-overcome technical obstacles that would prevent delegation from being implemented in relationship-centric systems. Mainly, major policy extensions but also new protocols are required in order to implement delegation.

4.3.3. 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 rather complex. In particular, the issue of delegation becomes much more complicated than in relationship-focused systems.

In general, we find two variants of delegation in credential-focused systems. One requires the presence of the IdP 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 (mainly *data minimization*, *anonymity*, *integrity*, *sharing prevention*), while adding additional functionality such as k-times use delegations. Overall, this first approach to delegation in credential-focused systems has a similar technical complexity as its relationship-centric pendant.

The second class allows a user to create a delegation credential by herself without involving the IdP in the delegation. Exactly this requirement of not involving the IdP accounts for the difficulties in the realization as this cannot be done by purely policy-based mechanisms. Achieving this 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 [19]. There exists no (efficient) solution to this problem, yet.

In general, we perceive it as extremely 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 (cryptographic) tools for moving beyond the boundaries of this paradigm. We furthermore think that the policy-based approach for delegation in credential-focused systems is applicable in practice until better mechanisms have been found by cryptographic research.

5. Conclusion

We contributed to the notion of user-centric identity management by i) a taxonomy unifying today's notions; ii) a discussion on mechanisms useful for accomplishing properties described in our taxonomy; and iii) a discussion of the differences of the two predominant paradigms. 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.

References

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

[2] Claudio Bettini, Sushil Jajodia, X. Sean Wang, and Duminda Wijesekera. Provisions and obligations in policy rule management. *J. Netw. Syst. Manage.*, 11(3):351–372, 2003.

[3] Claudio Bettini, Sushil Jajodia, X. Sean Wang, and Duminda Wijesekera. Provisions and obligations in policy rule management. *J. Netw. Syst. Manage.*, 11(3):351–372, 2003.

[4] Abhilasha Bhargav-Spantzel, Anna Squicciarini, and Elisa Bertino. Privacy preserving multi-factor authentication with biometrics. In *DIM '06: Proceedings of the second ACM workshop on Digital identity management*, pages 63–72, New York, NY, USA, 2006. ACM Press.

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

[6] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. Technical Report 96-17, 28, 1996.

[7] Piero Bonatti, Sabrina De Capitani di Vimercati, and Pierangela Samarati. An algebra for composing access control policies. *ACM Trans. Inf. Syst. Secur.*, 5(1):1–35, 2002.

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

[9] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, New York, NY, USA, 2004. ACM Press.

[10] Jan Camenisch. 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.

[11] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. Technical Report Research Report RZ 3295, IBM Research Division, November 2000.

[12] Jan Camenisch and Anna Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In Birgit Pfitzmann, editor,*Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer Verlag, 2001.

[13] Jan Camenisch and Anna Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, volume 2442 of *LNCS*, pages 61–76. Springer Verlag, 2002.

[14] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology — CRYPTO 2004*, LNCS. Springer Verlag, 2004.

[15] Jan Camenisch and Victor Shoup. Practical verifiable encryption and decryption of discrete logarithms. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, LNCS, 2003.

[16] Jan Camenisch, Dieter Sommer, and Roger Zimmermann. A general certification framework with applications to privacy-enhancing certificate infrastructures. In*Proceedings of the 21st IFIP International Information Security Conference*, 2006.

[17] Kim Cameron. Laws of identity, 5/12/2005.

[18] CCITT (Consultative Committee on International Telegraphy and Telephony). *Recommendation* $X.509$*: The Directory—Authentication Framework*, 1988.

[19] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. Cryptology ePrint Archive, Report 2006/184, 2006. `http://eprint.iacr.org/`.

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

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

[22] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, August 2004.

[23] 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.

[24] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology — CRYPTO '86*, volume 263 of *LNCS*, pages 186–194. Springer Verlag, 1987.

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

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

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

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

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

[30] ISO. Annex C – the RSA public key cryptosystem, in ISO/IEC 9594-8:1989(E). Part of ISO/IEC X.509 Draft Standard, 1989. no note.

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

[32] Chris Kaler and Anthony Nadalin. Web services federation language, 2003.

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

[34] Ninghui Li, William H. Winsborough, and John C. Mitchell. Distributed credential chain discovery in trust management: extended abstract. In *ACM Conference on Computer and Communications Security*, pages 156–165, 2001.

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

[36] Chris Lüer and André van der Hoek. Jploy: User-centric deployment support in a component platform.

[37] Anna Lysyanskaya, Ron Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In Howard Heys and Carlisle Adams, editors, *Selected Areas in Cryptography*, volume 1758 of *LNCS*. Springer Verlag, 1999.

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

[39] Steven J. Murdoch. Hot or not: revealing hidden services by their clock skew. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pages 27–36, New York, NY, USA, 2006. ACM Press.

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

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

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

[43] OpenID. `http://openid.net`.

[44] Andreas Pfitzmann and Marit Hansen. Anonymity, unlinkability, unobservability, pseudonymity, and identity management Ű- a consolidated proposal for terminology, May 2006.

[45] Andreas Pfitzmann and Marit Köhntopp. Anonymity, unobservability, and pseudonymity – a proposal for terminology.

[46] J. Postel. DoD standard Internet Protocol. RFC 760, January 1980. Obsoleted by RFC 791, updated by RFC 777.

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

[48] JAP Anon Proxy. Anonymity and privacy, 2006. `http://anon.inf.tu-dresden.de`.

[49] Ronald Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[50] IBM Microsoft RSA VeriSign. Web Services Federation Language (WS-Federation). version 1.0. July 8 2003. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnglobspec/html/ws-federation.asp.