Secure identity management for pseudo-anonymous service access

Technical Report Number 546

June 2004

Michael Hitchens
(Department of Computing, Macquarie University)

Judy Kay and Bob Kummerfeld
(School of Information Technologies, University of Sydney)

ISBN 1 86487 637 9
Secure identity management for pseudo-anonymous service access

Michael Hitchens *, Judy Kay # and Bob Kummerfeld #

* Department of Computing,
Macquarie University, Sydney Australia, 2109
# School of Information Technologies
University of Sydney, Australia 2006.
michaelh@ics.mq.edu.au, {judy,bob}@it.usyd.edu.au

Abstract: In this paper, we propose an architecture that enables service providers to give personalised service to users, while minimising the risks to the users’ privacy. We describe how security concerns have been at the foundation of the design of two critical parts of such systems: a mechanism that supports users in defining authenticated personas that can be pseudonymous; and mechanisms for users to share such personas with service providers. A trust-based approach is used to make decisions about accepting personas. We illustrate the issues and approach with an extended scenario that is used to illustrate the design of the architecture and the detailed processes in managing authenticated personas.

1. Introduction

One of the important current developments in the role of computing in modern society is the emergence of pervasive computing. This technology will potentially deliver a world where a multitude of computing devices (such as sensors, embedded processors and traditional computers) are seamlessly integrated into the everyday environment. The result will be a marked increase in the amount of information and the range of services that depend upon computing support. The provision of these will often depend upon the provider obtaining some information about the user. This information will enable the provider to decide whether or not to give the user access to such services and, possibly, to personalise those services. For their part users may be reluctant to disclose any more information than is absolutely necessary.

In more concrete terms, consider the ability of service providers (including both traditional and e-commerce retailers) to offer improved services based on characteristics of the customer. For example, loyalty program rewards may be offered based on information held in a PDA possessed by the customer, some services, such as access to cigarettes or alcohol, may be restricted based on the age of the customer as reported by their PDA or staff at a student help desk may be able to provide better assistance if the student’s knowledge and background are readily available to the staff member. This is an example of personalisation of service.
Conversely, customers will often wish to limit the amount of information that can be obtained by the retailer or other service provider. For example, a person purchasing alcohol may wish to prove their age without revealing their name or address. Or, to take another example, a student obtaining discounted travel may wish to prove they are a student without revealing their name or where they are enrolled. The provision of such services depends upon characteristics of the user other than their name (or other directly identifying characteristic). Indeed the user’s identity is irrelevant in the examples given. Such situations obviously involve considerations of privacy/and or anonymity.

If retailers are to provide the improved services based on information provided by the users, they want to be able to ensure the accuracy of such information. This is especially true if such information represents only a portion of the total information about the user. Providing only partial information requires a mechanism for defining the relevant subset of information.

Current research into privacy in pervasive systems has focussed primarily on privacy of location information \[1,2,3\] or on the control of information obtained from sensors and other pervasive infrastructure \[4\]. Providing services in the manner described above will require the provision of controlled levels of anonymity to allow users to provide sufficient, and no more, information to service providers (ie limited user models), and in a manner in which those service providers can be certain of the veracity of that information. This will require both personalisation of services based on user models as well as controlled levels of anonymity. These topics have been studied in relation to the world wide web \[5\], but have received less attention in the pervasive computing context.

In this paper we propose an architecture that can aid service providers in giving their users enhanced levels of service while minimising the risks to the users’ privacy. The information is communicated to the service providers on a pseudonymous basis, but one that allows the provider to have at least some trust in the accuracy of the information. The user’s privacy is protected by the pseudonymous form of the communication. The information itself is part of a user model about the user. User modelling is an area of active research and provides a flexible and structured means of storing and conveying the necessary information. In Section 2 we discuss user models and anonymity in more detail. Section 3 discusses the aims of our proposals and gives a simple, but detailed, example of how a system based on our ideas would work in practice. In Section 4 we present our architecture from a high level perspective, together with an example of its use. In Section 5 we detail some of the required protocols. The underlying trust model of the architecture is discussed in Section 6. Section 7 presents our conclusions and plans for future work.

2. User Models and Anonymity

This work draws on two other research areas, user modelling and anonymity. The former provides a supporting foundation for structuring, storing and communicating the information about the users. The latter provides the necessary privacy for the users.
A user model has been defined [12] as a system’s set of beliefs about a user. A review of some representations for user models is in Kobsa [13]. A user model represents and manages the components of information about a user needed to personalise services in ubiquitous computing environments, such as those we have described. There are important constraints on the management of that user model because it involves information that may be private. If this information is to be deployed in a pervasive computing environment, it must be safe-guarded. This has been widely acknowledged by many, perhaps most, researchers in the area of pervasive computing since some form of personalisation is intrinsic to the core goals of the systems. This has led to establishment of principles for managing such information [14], research towards implementing support for them, such as [15].

In addition to the general protection of the private information in user models, there is a critical need to enable users to define useful user models with minimal effort. Once that is done, it is important to also help users to also define, also with minimal effort, subsets of the user model that can be released in different situations. We call these subsets personas. Finally, we need to ensure that the components of such personas can be correctly interpreted in the ubiquitous computing environment. This means that applications in that environment must share the vocabulary used to define the user model components comprising personas available in that environment. These are important, yet neglected areas of research that must be tackled with security as a foundation of the design. This complements quite comprehensive analysis of privacy support for personas [11] starting with the assumption that the user model and personas have already been suitably defined.

Anonymity has been an active research topic for some time, in relation to the World Wide Web [5], the internet [6,9], as a study in itself [7] and in relation to location privacy in pervasive computing [8]. Anonymity is based on the concept of one entity being able to interact with another without the second entity learning anything about the “true” identity of the first. In a more general terms “the nymity of a transaction can be defined to be the amount of information about the identity of the participants that is revealed” [6].

Various terms are used (with occasionally conflicting definitions) in discussing anonymity – such as pseudonymity and pseudo-anonymity. A pseudonym is a “name” (or other identifying piece of information) that the receiver knows is not a true name, but can be used to link together transactions.¹ This linkability property is important, as it allows one entity to tailor communication with another based on past behaviour, but without knowing anything about the true identity of the second identity.

Most research into anonymity has been directed at protecting the identity of the originator of a message – such as an e-mail message, http request or internet packet. The explicit aim has been to minimise the amount of information in a message so as to reduce the risk of inadvertent identity disclosure. The reason for this minimisation is that the chance that a hostile entity may be able to deduce identity from message contents increases as the amount of information in the message increases [9]. However, in a truly pervasive environment the user population is so large that such

¹ A related term, pseudo-anonymity, refers to the situation where an entity receives a name, but is unsure as to whether it is a pseudonym or a true name.
means of identification become progressively difficult. The advantages of personalisation may be seen to outweigh such dangers. This leads to our proposal, where the user deliberately provides certain information about themselves, but in a way that should preserve the privacy of their identity. The pseudonymous identity then becomes not just a trusted, but sterile, identity, but one that has some guaranteed characteristics. The challenge becomes one of how to provide this information in a secure manner while still protecting the user’s “true” identity.

3. Aims and Detailed Example

The intent of the design proposed here is to allow users to provide some information about themselves, while reducing the risks to their privacy. We explicitly expect that often users will not wish to give away identifying information, such as their name or address. This makes it harder for the service providers to verify the validity of the information that is provided. The aims of our design can be summarised as:

1. Allow users to provide subsets of information (ie personas) concerning themselves, in a form the service providers will understand.
2. Allow the creation of personas with minimal user effort
3. Ensure that the provision of information in the form of personas has minimal chance of breaching the user’s privacy, especially in terms of revealing their identity.
4. Provide a flexible and extensible means of storing and communicating the personas.
5. Provide a basis on which the service providers can have at least some level of trust in the accuracy of the provided information.

We illustrate these aims by use of an example scenario, which we will refer to through our description of an architectural approach to supporting these aspects of personalisation and anonymity in pervasive computing environments. Our example consists of a student, Jane, of the University of X who wishes to purchase some textbooks from the university bookshop.

As Jane walks into the University Bookshop, her PSD (Personal Server Device) beeps and asks if she wants to release her AnonEnrolment persona. This was defined by the University of X when she enrolled and loaded onto her PSD at that time. It has a list of her subjects for the coming semester and those for which she is pre-enrolled for the following semester. It does not include her name or other personally identifying information.

This illustrates a user’s view of the way that, with minimal effort, it is possible to release a persona. The personas that a user has available are held in their PSD.

The scenario also illustrates how an environment, like the bookshop, that is aware of relevant persona templates enables it to make sense of the information released about the user. A persona template (or simply template) is a definition of the structure of a persona. A template allows an entity receiving a persona to interpret the persona and identify the individual pieces of information of which it is composed. Identification of the appropriate template is included within a persona. Interpretation
of a template may also require additional knowledge to be shared and amongst its users, such as the definitions of the base types of which the template consists.

The AnonEnrolment template was defined by the University of X (so instances of it could be released to students) and communicated to the university bookshop.

Jane approves the release of the persona. It is delivered to the environment (ie., bookshop’s system). As she approaches the customer enquiry screens near the entrance, she sees a list of the textbooks and recommended readings for her current courses. There is an annotation against the French poetry book, indicating that the lecturer has just sent the recommendation for it today. (So it does not appear in the printed booklist in the class notes.) Although Jane is unaware of it, the terminal shows the French grammar book that was ordered in restricted numbers and is only available for sale to students enrolled in her course.

This illustrates the way that the parts of the user model in the persona affect some personalisation, providing just the details of books relevant to Jane and giving her details of restricted information. It also illustrates that the environment may have to establish the authenticity of the information in the persona. A persona is certified by some entity (normally a third party). In our vision, this certification takes the form of a signature on the persona and the provision of public key pair for the persona. The key pair allows the entity associated with the persona, in this case Jane, to prove her entity.

The next week she returns to the University Bookshop to buy a gift for her mother. Things start in exactly the same way, except that when the PSD asks to release the AnonEnrolment persona, she rejects this, instead activating her MumBook persona: this reflects her mother’s book preferences for motor mechanics and travel. Jane downloaded this from her mother’s server several months ago, under the pretext that she felt she had similar interests and would appreciate being able to use her mother’s book interest persona at the local library. This time, the bookshop system, provides pointers to the new and best-seller books in those areas. The bookshop has no means of linking the two visits to each other or to the “true” identity of Jane.

This stage of the scenario illustrates that a user may have several personas that are equally relevant to a particular context. This means we need to support sensible default personas and easy means to alter the choice of persona. It also introduces the interesting case where Jane may hold another person’s persona. If this is to be possible, we need to support such sharing if the owner is willing to allow that. Alternatively Jane can provide this persona without external authentication and it is then up to the bookshop to decide whether or not to accept it.

The above scenario is relatively simple. However it does illustrate several important practical problems. A university student may wish to prove that they are indeed a student to service providers that provide personalised service on that basis. As names and student numbers might be useful in recovering student results, students may well wish to not advertise such information. Service providers will often use past interactions to tailor future ones – this can be seen currently on the World Wide Web. As Jane, in our example above, is not herself interested in motor mechanics and travel, she may not wish to be plagued in future visits by information about such
books – again something that currently happens on the web where interaction, and therefore personalisation, tends to be unitary.

Another simplification in this example is that the bookshop is able to receive, and verify, the information about the user as a single, signed, piece of information. One of the strengths of user modelling is that the pieces of evidence which make up the user model may have been obtained at different times and from different sources (including the user themself). Thus the persona supplied to a service provider may actually consist of several pieces, each separately signed (possibly by different signatories). The service provider then has to decide how much trust to place in the persona, as a whole. This leads to the question of trust models, which will be discussed below.

4. System Architecture

Figure 1 is an overview of the architecture we propose to deal with issues such as those described above. It consists of users, service providers and authorising entities. In our example scenario, the users are students at the University of X, the service provider is the University Bookshop and the authorising entity is within the enrolment administration section of the University of X.
Essentially, an authorising entity (such as the University of X) provides a service to users by defining a range of authorised templates for its students and by guaranteeing the veracity of particular instances or pieces of information that will go towards constructing actual personas. We gave one example of these above, the template designed to define the student’s current year enrolment and the actual instance delivered to Jane by the University. We now describe how each entity interacts to support personalisation scenarios.

First, consider the structure of the authorising entity. It needs to create the authorised personas it will supply to individuals (like Jane) and the authorised persona templates it will give to service providers so that they will be able to interpret and make use of the personas of users. It has a *Persona Template Composer*. This allows authorised individuals to construct new templates, modify or delete existing templates and to define which users are to be given personas adhering to a template. In our example, this is the *AnonEnrolment* persona, which consists of the subjects enrolled in and a statement that the holder of that persona is an enrolled student. Continuing our example, scenario, the University of X must define the template as a list of all the courses that a student can enrol in. The template structure may be defined in some structured language, such as XML, or in manner specified by the user modelling software. Once defined, such persona templates are added to the authorising entity’s *Persona Template Database*. An authorising entity will normally be willing to supply personas conforming to a number of templates, so these need to be stored.

The *Persona Template Distributor* then uses the *PersonaTemplate Authoriser* to create an authenticated version that is released to providers and users, such as Jane and the bookshop in our scenario. The *PersonaTemplate Authoriser* does this by signing the template structure using the private key of the authorising entity. Service provider systems need the templates to decide which ones they will request and accept from users. Users’ PSDs need the templates to be able to understand such requests from service providers. The *Persona Managers* of users and service providers place the persona templates in their system’s *Persona Template Database*. It can be recovered from them as necessary.

When an authorising entity needs to provide a user with an authenticated persona, the *Persona Distributor* selects the appropriate template from the *Persona Template Database* and then consults the *User Information Database* to construct the partial user model for that person (ie, fill in the template to create an instance persona). The authorising entity then authenticates the persona (typically by signing it using the *PersonaTemplate Authorise* as for templates) and then the *Persona Distributor* delivers it to the person’s Personal Server Device. The authorising entity (using the *Persona Distributor*) may also provide a persona to a service provider’s system. In effect, the service providers are also users and may require personas if users wish to identify them. The main differences between service providers and users are:

- The systems of service providers will typically be more powerful and have more storage, not being personal wireless devices.
- Service providers will probably be less interested in anonymity. If the service provider has a physical presence – such as a shop front – anonymity has little appeal to them.
§ Service providers will normally advertise their existence by some form of broadcast message, thus initiating the exchange between user and service provider or there could be some form of service discovery protocol.

These differences make it useful to distinguish the two classes of system, rather than compose the architecture purely in terms of users and authorising entities.

The Persona Manager of the user (or service provider) first ensures that a copy of the appropriate template is available. If it is not, it is obtained from the authorising entity. The authenticated persona is then added to the User Model. It can now be used as a persona in its own right, or the information contained within it can be used to make further personas.

In the Jane scenario, this means that the authorising entity uses its persona template for the AnonEnrolment persona to create the particular partial user model for Jane. This partial model will consist of two sets of subjects, those of the current semester enrolment and one of the next semester’s pre-enrolment. The authorising entity needs to authenticate this information. This could be as a single authentication on the entirety of the information, authentication of each individual piece of information or separate authentication of subsets of the information. Authentication takes the form of creating an asymmetric key pair for the information, and signing the information and public key of the pair. Obviously grouping information together will improve efficiency, as there will be fewer keys to manage and less computation will be required to authenticate a complete persona. On the other hand using a finer granularity of authentication will provide more flexibility to the user in creating further personas from the individual pieces of information.

Once an authenticated persona arrives in the user’s PSD, the Persona Manager places all this information into his/her user model. This means that the user can employ the supplied persona as in the bookshop scenario. The user can also create additional unauthenticated personas. This means that the University effectively supports her definition of personas related to her university studies.

At this point, the service provider has details of the persona templates defining what user model personas it can expect from students. Correspondingly, a students-like Jane carries her PSD, containing her user model, which has been updated from the information in the AnonEnrolment persona. When Jane walks into the bookshop, her PSD is detected and the bookshop system sends the list of names of expected personas. The first one recognised by the PSD is selected as the default and it asks the user to approve its release to the bookshop system.

The user’s Persona Manager is the software that enables the user to manage their personas. It enables the user to obtain new personas and templates from the authorising entity (such as the university) and handles interaction with the service providers (such as the bookshops). In user modelling terms, the persona information obtained from the authorising entities, provides evidence for the user’s own user model.

The Persona Manager enables the user to use either signed personas or to create new personas, such as the one for her mother. Note that such personas will not be signed by an authorising entity. The service provider may choose to place less trust in such information in such personas. In our scenario, restricted books might not be listed for users with an unsigned persona of enrolment information.
The service provider can obtain personas (not just templates) from an authorising entity. This can be sent to the PSD’s of users to authenticate the service provider. This could either be as part of a broadcast in response to an explicit request from a user’s PSD. This gives an obvious symmetry of roles to the users and service providers (beyond the question of who initiates the communication). This allows our architecture to be used for user-to-user communication as long as one PSD takes responsibility for initiating communication. This would allow, for example, Jane to obtain the information about her mother’s reading preferences.

5. Detailed processes for managing authenticated personas

We now define the detailed steps involved in the three major classes of interaction: the persona provider releasing a persona template to a service provider or user; the release of a persona to a user; and the interaction between the user and service provider.

5.1 Obtaining new Persona Templates

We begin with the process for a user or service provider obtaining a new persona template from an authorising entity. This is illustrated in Figure 2.

![Figure 2. Obtaining New Persona Templates](image)

**Message 1 Alert**

This is an optional message, informing the end system (User PSD or Provider System) that a new template (or templates) is available. It gives the names, version numbers and creation times of the templates. These, together with authorising entity identity, should be sufficient to uniquely identify a template. Authoring entity identity and serial number could also be used for this purpose. The message includes a timestamp to prove its freshness and is signed using the public key of the authorising entity. Such a message would be periodically broadcast to service provider systems and user PSDs, as required.
Message 2 Request Template

The end system requests the authorising entity to provide one or more templates. This request may be qualified by template name, version number, time etc to obtain responses such as

1. A given version of a one or more named templates.
2. The most recent version of a set of templates – either all templates or those specified in the message.
3. The most recent version of a set of templates where there exists a version created after a given time – either all templates or those specified in the message.

The name/version of template could have come from an alert or from communication with another end system. This message also includes the identity of the end system. The authorising entity then decides whether it needs to authenticate the end system or whether it will simply release the requested templates. If authentication is not required the next two messages are omitted.

Message 3 Authentication Challenge and Message 4 Response

These two messages form a standard challenge-response pair. The challenge is based on the knowledge of the private key of an asymmetric key pair.

Message 5 Provide Templates

The authorising entity replies with the requested templates. The templates would have already been signed by the authorising entity as described above. The authorising entity should still sign this message, and include a timestamp, so that the recipient can be assured of the message’s freshness. This would prevent a hostile party replaying an earlier message, which may include genuine, but obsolete, templates. The end system’s persona manager examines templates and decides whether to include them in its template database.

<table>
<thead>
<tr>
<th>User PSD or Provider System</th>
<th>Authorising Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Request Persona</td>
<td>2. Return Persona</td>
</tr>
</tbody>
</table>

Figure 3. Obtaining a New Persona

5.2 Obtaining a new Persona

The procedure for obtaining a new persona is illustrated in Figure 3. While either service providers or users may require personas, we consider, as noted above, that users will be more interested in anonymity than service providers, and so users will make use of the request personas. For that reason, we frame the following in terms of users requesting personas – for users it can be read as including service providers.

These messages are encrypted, so as to prevent an eavesdropper linking identities and personas. We have omitted any description of secure communication setup, key management, etc. as existing technologies can easily be used for our purpose here. For simplicity, we can assume that the PSD and authorising entity can establish a
symmetric session key, which will be used to encrypt the request for the persona and the return of the persona itself.

**Message 1 Request Persona**

The user requests a new persona, and gives their identity so that the information to be included in the persona can be found in the authorising entity’s user information database. The request must include at least one of

$\$ A template name, and optional version number. If no version number is given, the latest version is used.

$\$ Information to be included in the template.

If information is given, but no template identified, the authorising entity may still decide to authenticate (ie. sign) the information, creating a persona without a template. In either case the authorising entity will check the supplied information against that in its database. If the two do not match, the authorising entity may choose which to use or simply abort the persona creation. All these decisions would be based on the policies of the organisation controlling the authorising entity. If the user supplies a template identifier, but no actual information, the authorising entity attempts to create the persona from the information in its user information database.

When a template is specified the authorising entity needs to check whether the user is allowed to have a persona created from the identified template before creating the persona. Access to templates is specified through the persona template composer, as described above. Between Message 1 and Message 2, the authorising entity may require an authentication process, for example by using challenge response. This has been omitted from the diagram for simplicity.

**Message 2 Return Persona**

After checking the relevant policies as described above and obtaining the necessary information the authorising entity creates the persona, including creating a new key pair for it, signs the persona (including the public key of the new key pair) and returns the persona and key pair to the user. Depending on the requirements of the technology used (such as the user model infrastructure) and the policies of its organisation, the authorising entity may sign the persona as a single unit, or as individual pieces of information (which would also mean creating a key pair for each piece of information). The persona is encrypted before sending to the PSD, as mentioned above, so that no eavesdropper can link the personas to the user’s actual identity or discover the private key(s) corresponding to that persona.

### 5.3 Interaction between User and Service Provider

The interaction between user and service provider is illustrated in Figure 4. This is where users identify themselves to the service provider by use of the anonymous (or more properly pseudo-anonymous) personas.

**Message 1 Provider ID**

This message more properly represents some form of service discovery process. For example, the user’s PSD could receive some form of beacon message from the provider’s system. The user would obtain from the service discovery process a persona for the service provider (and the unique template identifier for that persona)
and a list of persona templates that the service provider recognises. The templates would be uniquely identified as described above.

If required, the user’s PSD authenticates the provider based on its supplied persona. This could again involve extra authentication steps, such as challenge-response, and even a complete interaction of the kind illustrated in Figure 2, if the nominated template is unknown to the PSD and the appropriate authorising entity is currently accessible.

**Figure 4.** Interaction between User and Service Provider

**Message 2 Provide Persona**

A persona to supply to the service provider must then be selected. The selection could be managed in a number of ways. The user may have previously interacted with the service provider and the persona used last time could be used as a default. The PSD could prompt the user with a list of possible personas, with the list being composed on the template nominated in the service discovery step, personas frequently employed by the user, or other options as allowed by the persona manager software on the PSD (such as a preselected persona, based on the user anticipating interaction with this service provider). The PSD sends the persona to the service provider, along with the identification of the template to which the persona corresponds (if any).

**Message 3 Authentication Challenge**

After receiving and checking the supplied persona against the appropriate persona template the service provider will normally challenge the PSD to ensure that the persona use is valid (ie the PSD knows the corresponding private keys). Note that the user may provide a persona whose template did not appear in the list in Message 1. The user’s PSD may or may not specify a template (including authorising entity) for such a persona). The service provider could: abandon the exchange, authenticate anyway (assuming it can interpret the persona) or temporarily halt this exchange while it obtains the appropriate template from the authorising entity.

Regardless of how many key pairs are involved, the challenge and response should take two messages, as the service provider could challenge on all keys in a single (possibly lengthy) message. If multiple keys are involved, the authentication may have been only partially successful. It could also be that the user provides information with no corresponding key (perhaps it is information they entered into their own user model). The service provider will need to decide how much trust they
place in the persona and then decide exactly what service to provide. Our suggested trust model is described in the next section.

After a successful authentication the PSD and provider’s system interact, according to the service provided. If linkability between sessions with a single service provider is required this can be provided either by identifying common sessions by the public keys of the persona or by an identifying (eg., serial) number in the persona.

6. Trust Model

The basis of trust in our model depends on the possession of private keys of asymmetric key pairs. From a conceptual view, trust could be based on positing the existence of a public key infrastructure (PKI). This would give certificates which could verify key pairs, based on the signature on the certificate by a certificate authority (CA). The authorizing entities would act as CAs. In practice, a PKI that would encompass sufficient users, service providers and authorising entities to be useful does not exist. Entities within the system, especially users and service providers will be presented with certificates whose signer cannot be readily verified. We trust users and authorising entities not reveal the private keys. We therefore need to consider how the entities will determine the levels of trust they have for each other.

Trust between users and authorising entities is relatively straightforward. A user can register with an authorising entity via some off-line process. This will give an identity to the user for interacting with the authorizing entity. As part of that process, the user would obtain the public key of the authorizing entity (for storage on their PSD) and a key pair to authenticate their identity to the authorizing entity. Users and authorising entities can trust each other based on these keys. Note that this means that we trust users not to give away the private keys for either their identities for interacting with authorising entities or for the personas.

For trust between service providers and authorising entities the situation is somewhat more complicated. A service provider may register with an authorising entity, in which the case the above comments apply. However, it would be unreasonable to expect a service provider to register with all authorising entities. When a service provider receives (either directly or from a user) some information (such as a persona, element or template) from a previously unknown authorising entity it must assign a trust rating to that authorizing entity. The exact rating would depend upon the policies of the service provider, but would probably be low and may depend on the trust rating the service provider has for related authorising entities (if any are known). The trust rating would be adjusted over time as the information provided by that authorising entity proves more or less trustworthy. The level of trust in an authorising entity could also depend upon the trust placed in it by other, related, service providers. This would again depend upon the policies of individual service providers. Trust ratings in our model is rated as a number in the range [0,1]. A low value represents low trust, a high value high trust.

The most widely varying levels of trust will be between users and service providers, as the service providers must decide what trust to put on information that is provided on an anonymous basis. The various information in a persona may be
signed by a different authorising entities, some or all of which may be unknown to the service provider.

In deciding on the trust to place in a persona the service provider considers each element (or group of elements signed as a single unit) separately. Each element (1 to \(n\)) is assigned a weighting \(w_i\) of importance. It is up to the service provider to manage its own weightings. For calculating trust in a given persona the weightings are normalised to give each element a weighting of importance in the range 0 to 1. The authorising entity that signed each element has a trust rating \(t_i\) as described above. Trust, \(T\), in a persona can then be calculated as

\[
T = \frac{\sum_{i=1}^{n} w_i \times t_i}{\sum_{i=1}^{n} w_i}
\]

Each service provider must set its own threshold (or thresholds) for deciding whether or not to trust a persona based on this rating. A provider can decide, if the threshold is not met, to either simply reject a persona or to recalculate the trust rating only on parts signed by more trusted signers. This re-calculation may result in a value that exceeds the threshold. The service provider will then provide service to the user based only on the information in the reduced persona. Once accepted, subsequent behaviour of the user may be used to adjust trust ratings in the various signers of the persona elements. The same procedure can be used for establishing trust between users and for users determining whether to trust service providers.

7. Conclusion

We have presented an architecture that allows users to provide information to service providers in such way that:

1. users do not have to reveal their identity
2. service providers can establish a level of trust in the information

The service providers can use the information to provide personalised service.

The architecture presented meets our aims as stated in Section 3. The use of templates allows users and service providers to interpret personas. The employment of user modeling techniques to handle the personas that encapsulate the information gives a high degree of flexibility to the management of personas, such as in users constructing their own personas. The flexibility of the persona structure allows users to provide the minimal information necessary to obtain the personalized services. This minimizes the risk to the user’s privacy and hence protects their identity. The use of authorising entities to sign the information allows the service providers to establish trust in the information. There would be far fewer authorising entities than users in such a system, and they would be more visible, allowing service providers to monitor their trustworthiness.

It should be noted, that while we employ asymmetric key cryptography, we do not require a full PKI. For example, this means that we would not distribute certification
revocation lists. With the fine granularity of certification, they would be too large. User modeling techniques allow for the identification of outdated information.

Some issues do remain. While the processing and storage capacity of mobile devices has increased, the performance impact of the number of asymmetric key operations required to verify personas has yet to be tested. We are currently implementing a prototype based on this architecture, using Personis-lite [10] and will report on its performance in the near future. There are also additional issues in the trust model, such as the exact policies a service provider should employ in updating trust in authorising entities, which require further investigation.

References