On Helping Individuals to Manage Privacy and Trust

Stephen Crane, Marco Casassa Mont, Siani Pearson
Trusted Systems Laboratory
HP Laboratories Bristol

HPL-2005-53
March 17, 2005*

trust, privacy, personal systems, obligations, trusted platforms, prime, policies

Being able to say with absolute certainty that another party can be trusted to handle personal information with today's technology is probably unrealistic. In this paper we explain an approach to establishing trust based on the status of a remote platform and an anticipated willingness of the other party to comply with prior negotiated obligations. Ongoing monitoring and notification, and the ability of the individual to form a simple record of past interaction, provides the individual with greater confidence in situations where they need to share personal sensitive information with organisations they would otherwise not be able to claim they trust. We describe the principles of our approach and architectures that support a practical implementation.

* Internal Accession Date Only Approved for External Publication
© Copyright 2005 Hewlett-Packard Development Company, L.P.

On Helping Individuals to Manage Privacy and Trust

Stephen Crane, Marco Casassa Mont, Siani Pearson

Trusted Systems Laboratory (TSL)
Hewlett Packard Laboratories
Filton Road, Stoke Gifford, Bristol UK BS34 8QZ
{stephen.crane, marco.casassa-mont, siani.pearson}@hp.com

Abstract. Being able to say with absolute certainty that another party can be trusted to handle personal information with today's technology is probably unrealistic. In this paper we explain an approach to establishing trust based on the status of a remote platform and an anticipated willingness of the other party to comply with prior negotiated obligations. Ongoing monitoring and notification, and the ability of the individual to form a simple record of past interaction, provides the individual with greater confidence in situations where they need to share personal sensitive information with organisations they would otherwise not be able to claim they trust. We describe the principles of our approach and architectures that support a practical implementation.

1 Introduction

Within PRIME[1][2][3] we have been investigating how Personal Identifying Information (PII) can be shared between individual and individual, and between individual and organization, in a way that reassures the individual, who is recognised as the owner of the PII, that their information will not be misused or abused. The one factor that underpins the ability to share with confidence is trust. In this paper we describe our work to date in establishing techniques to manage trust in another party at initial contact and throughout the duration of an interaction.

Although in this paper we use the sharing of PII as our reference scenario, the techniques we discuss are not limited to privacy situations. At any time when one

[1] PRIME: PRivacy and Identity Management for Europe. European RTD Integrated Project under the FP6/IST Programme. http://www.prime-project.eu.org/

[2] The PRIME project receives research funding from the Community's Sixth Framework Programme and the Swiss Federal Office for Education and Science. This work was supported by the IST (Information Society Technologies) PRIME project; however, it represents the view of the authors only. The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

[3] Note that the architectures and concepts described in this paper will not necessarily be incorporated into the PRIME architecture, whole or in part.

party needs to assess the trustworthiness of another we believe our approach can be used to good effect. As will be seen, our architecture effectively treats trust as an out-of-band process, so it should be able to integrate the process with any general information-sharing process instigated between two or more parties, or indeed simply when one party needs to determine the trustworthiness of another party.

Being able to say that another party can be trusted to handle personal information with today's technology is probably unrealistic. Unless we can 1) completely isolate the processing from the operator and 2) rely on the technology and implementation, we have to rely on some level of faith in the other party. Requirement 1) is unrealistic since in practice virtually every application is likely to involve some form of human intervention, including access to the information after the 'trusted' processing is complete. Requirement 2) is currently difficult to demonstrate.

Since in practice individuals have difficulty proving 'before the event' that a recipient is trustworthy and will uphold their wishes, the next best approach (as in real life) is to establish an alternative means of enforcement. A contract gives an individual a strong indication that a recipient intends to carry out the individual's wishes and provides a means to identify deviation from agreed actions. Of course, the contract is only useful if it is enforceable.

A deceitful recipient of PII will most likely always be able to circumvent controls. However, the concept of a contract is useful for a recipient who has every intention of behaving properly, and wishes to demonstrate so in order to be differentiated from other less scrupulous recipients. This approach simplifies the enforcement challenge.

Large corporate organizations, for the most part, have strong reputation brands (which itself can be a basis for trust) which they would like to protect, and so take steps to behave honourably and fairly. Often the later is enforced through third party legislation and codes of conduct. These are the organizations that are willing to demonstrate openness and be held accountable for their errors.

Trust is a combination of both social trust and technical trust. Both of these aspects of trust influence a user's overall trust assessment. Another way to look at trust is in terms of three components: technical, history and reputation. Some readers may consider history and reputation to be the same thing. However there is a subtle difference. History and reputation form a social assessment, each being based on past interaction with the recipient. In the case of history the assessment is made on past interactions that the user has had. Reputation includes interactions that other individuals have had. Reputation introduces a further complexity in that the user also has to judge the trustworthiness (or reliability) of the third party's assessment. The user must also be aware that the quality of a reputation indicator may vary between providers and therefore be ready to compensate accordingly. Reputation is clearly strongly influenced by social understanding, but history (as perceived by the user) is measurable as long as the user can articulate the conditions under which past performance has a bearing on future performance. It is this ability of the individual to collect and assess evidence related to past events that provides a means to form an opinion about trustworthiness in the absence of other more definitive trust indicators.

2 Our concept of trust

Background

Individuals want to be able to release personal information in the confident belief that it will only be used in the way the individual intended. Providing this assurance is the key to demonstrating trustworthiness. For most situations, the trust that individuals place in an organization is a mixture of technological trust (system trust) and social trust (human trust). In many situations it is possible to manage technical trust by minimising risks using threat/vulnerability models. Social trust – the trust we place in another human – on the other hand, is very much more difficult to understand, measure and control. Except for a handful of niche applications, technology and humans interact to affect outcome. On the whole, trust is limited to a belief that (say) an organization will fulfil a request. There is usually limited evidence to support this belief other than possibly a contract that is only enforceable in specific circumstances. One way to understand trust better is to consider the nature of the participants. On the one hand there is the deceitful recipient who, if sufficiently motivated, will be able to circumvent controls (not always technical). This is a difficult category to deal with unless we can separate system and human trust. Another category is the recipient who sets a high standard of business conduct and wishes to demonstrate this in order to provide differentiation from other less scrupulous recipients.

Organizations that have valued brand and reputation are keen to ‘show’ individuals that they can be trusted even if they cannot present indisputable facts that support their claim. Of course, even the best-intended organizations make unintentional mistakes. These organizations would welcome solutions that help them keep in check and reaffirm their own trust in their systems.

Our emphasis is on the individual as the consumer of a service. However, it should also be recognized that an individual can be a service provider too. Since trust is (in part at least) a multiparty experience, it is inevitable that any solution to the trust problem will involve both user-side and server-side technologies. In this paper we are concerned with establishing trust in the service provider and choose to ignore the trust that the individual (or another party) might have in the client system.

For further reading on aspects that have affected our understanding of trust see [CC03] and [KSG04].

Organisational Trustworthiness

Trust in an organization is built up over time, based in part on past interactions. Evidence that an organization is willing to commit to an intended action, possibly in the knowledge that not doing so will incur penalties, is a useful sign of good intentions.

Typically, an individual would either review or present the terms under which the interaction will take place (i.e. a policy or contract). Once accepted, these terms are binding to some degree. As required, the user reviews the interaction and compares outcome against the contract, particularly where the terms specify several points in the process where an assessment can be made (c.f. project milestones). This leads us to a process with clearly definable steps:

- Policy/contract comparison between user and organization
- Fulfilment (by an organization)
- Checking (by a user)
- Opinion forming (by an individual – essentially retention of evidence to aid trust evaluation during future interactions.)

3 Problems associated with disclosing PII

Nowadays, PII is more exposed to misuse than ever before. Even within a personal platform it cannot be considered safe. Spyware, viruses and the general lack of control that individuals have means the protecting PII is a challenging task, especially for those not skilled in security. Similarly when individuals release information to an organisation there is little to prevent its misuse or provide the individual with the ability to determine either beforehand or after the event how the PII will be used. Beyond expressing a request using the opt-in / opt-out check boxes, and checking the organisation’s privacy policy, there is usually little more the individual can do to protect themselves.

PII management options fall along a spectrum (Fig. 1). At one extreme there is the situation where a user adopts the approach of not releasing any personal identifying information at all. Instead, the user provides the recipient with information that has passed through some form of anonymiser[4].

At the other end of the spectrum is ‘unrestricted release’ of identifying information. This approach potentially exposes personal information to the greatest level of abuse, but is common practice nowadays for most commerce and services-based interactions.

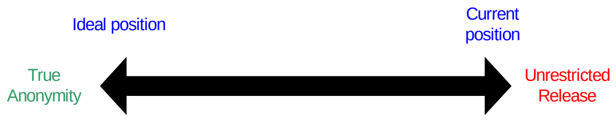

Fig. 1.

Anonymising approaches could be considered the ideal. However, whether the world of commerce is able and willing to adapt existing practices and procedure to the extent that some anonymising techniques demand is still unclear. Furthermore, it is doubtful that a completely anonymous approach is possible with many scenarios, e.g. healthcare and travel, where personal information simply must be divulged.

[4] Anomymisation is the de-personalisation of data.

Our approach to PII management, as outlined in this paper, is to provide the tools that allow PII to be control after it has been released. We still support the technique of anonymisation, and can imagine situation where the first step in an interaction is to minimise the release of PII. Consider, for example, an individual who requests advice about general medical care and (presumably happy with the advice) then asks for more specific information based on personal symptoms. In this situation the interaction may begin anonymously and progress through to partial or full release of identifying information depending on how the interaction develops.

4 Architecture

The proposed approach differs from existing approaches (e.g. P3P[5] [P3P]) by providing feedback to the individual and indeed involves an individual/client platform in the process of 'active' comparison and management. The process can be presented diagrammatically as shown in Fig. 2.

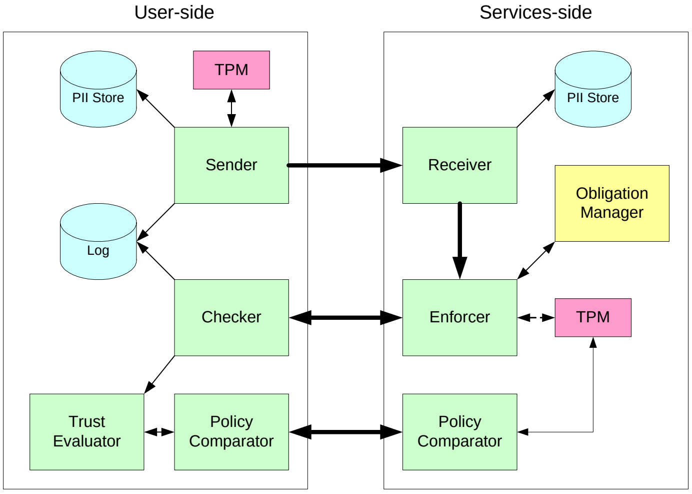

Fig. 2.

Policy exchange and negotiation

The policy, which contains the obligations, is initially exchanged between the Sender and Receiver, and negotiated by the Policy Comparators. PII is only released once a policy has been negotiated successfully. Policy negotiation involves the Sender presenting the policy as a list of requirements to the Receiver and obtaining

[5] Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/

back a list of those requirements that the Receiver can perform. The Policy Comparator then reports the outcome of the ‘negotiation’ to the Sender, which then decides whether to release the PII under the terms of the ‘negotiated policy’. The agreed policy is retained by the Receiver’s Policy Comparator.

Once an obligation is placed on the Receiver, a description of the obligation is passed to the Policy Enforcer.

Trust measurement

The Sender’s policy will specify exactly what trust conditions must exist before data can be released. For example, the Sender may wish to determine whether the Receiver has a functional TPM installed[6].

By way of an example of an obligation, suppose a policy states that data must be deleted by the Receiver after 30 days. The Obligation Manager instructs the Enforcer to perform the deletion and notifies the Checker. The Checker maintains a log of completed and outstanding obligations.

In addition, the Checker performs proactive obligation checking (through the Enforcer and Obligation Manager), and presents to the individual an aggregated and meaningful trust assessment via the Trust Evaluator.

In this architecture we show the use of a Trusted Platform (TP). The function of the TPM is to provide protected storage (user-side) and attestation (signing) of claims and actions (services-side).

A more sophisticated architecture

The architecture shown in Fig. 2, and the accompanying description, illustrate a very simple implementation of the processes we have discussed. In practice a more sophisticated architecture would be required. In the appendix we have provided a description of such an architecture. We have not described in any detail how this revised architecture achieves the stated goals, but it is fair to say that the principle of operation are very similar to the simpler version, and we hope that the naming of components will make their function obvious. See Fig. 5.

5 Obligations

The policies that the client and server negotiate and agree contain conditions that the server must fulfil which we call *obligations*. Obligations cover many aspects of the process of sharing PII. Here we are only interested in those obligations that relate to trust, which we call *trust obligations* but will refer to in this paper simply as obligations.

[6] We note that currently a TPM is state-of-the-art TP technology, but not widely deployed in servers.

Another way to think about obligations is as Service Level Agreements, or SLAs. An SLA can give an individual greater assurance about expected behaviour, especially when the SLA is bound to a server-side platform identity, for example bound to the trusted identity that a TPM provides. It can also help with the automated fulfilment of contractual obligations. Even using existing trusted technology it would still be possible to bind an SLA to a server platform using a public key for which the corresponding key pair was securely generated. See Fig. 3.

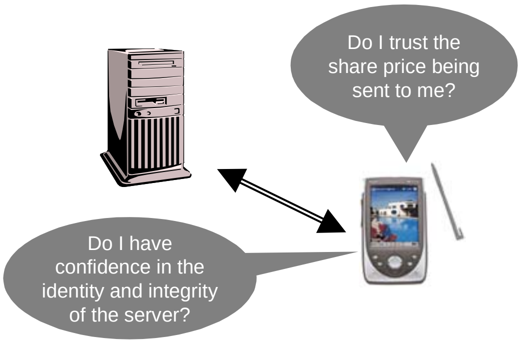
Do I trust the share price being sent to me?
Do I have confidence in the identity and integrity of the server?

Fig. 3.

In the example of an individual who is using a roaming appliance, e.g. mobile phones or wireless-enabled PDAs, to access an e-commerce application, the individual could be certain that the requested service is coming from the expected server.

Looking to the future, trusted platform will be able to provide evidence about the trustworthiness of the software being run and the operating environment. These *integrity metrics*, when related to the server-side, could help the individual assess the trustworthiness of the platform before authorising an action. See [Pea03-2] for a discussion about SLAs for profiling and [Pea03-3] for a more detailed description of what is possible using integrity reporting features of a TPM.

In this paper our objective is not to discuss how obligations that have been established by the individual or are managed on the server-side. We are interested in how obligations are used to convince the individual that the faith they placed in the organisations is justified.

Returning to obligations, the individual begins by stating their conditions. These can be simple or complex (see appendix for a description of possible obligations). In this paper we choose to demonstrate our architecture using a simple obligation, i.e. PII must be deleted after a stated number of days. From this point forward in our discussion we will use the terms *client* and *server* to represent the individual and organisation respectively, except where it makes sense to differentiate. The client states his obligations, to which the server responds by either accepting the obligations or offering revised/alternative obligations. The latter will initiate a negotiation between client and server. (Of course the server could simply refuse the obligation, in which case the negotiation takes on a different theme, and may result in the client declining to share PII with the server because the server fails to meet the client's minimum trust threshold.)

Assuming that the obligations are accepted, the server will instantiate processes to ensure that the obligations are fulfilled. As this paper concentrates on the client side we will not explain in details how this process is achieved, suffice to say that in addition to providing enforcement mechanisms, the server will also provide the client with notification of the status of an obligation. Exactly how this happens is the subject of the remainder of this paper.

To learn more about obligations and server-side enforcement mechanisms please refer to the related publication, [MCM04-1] and [MCM04-2].

Checking the status of an obligation

The server provides the client with notification of the status of obligations so that the client can be satisfied that the server is ‘behaving properly’. Notifications come in two forms: *solicited* and *unsolicited*. Unsolicited notifications are messages sent by the server to the client that are automatically generated when the status of an obligations changes. Taking the earlier example, on the stated day the server should delete the client’s PII and notify the client that deletion has taken place. Solicited notifications are the reverse, where the client generates a spontaneous request for the status of an obligation, which is passed to the server. The server will respond as before.

When the client initially communicates an obligation to the server the client also retains a local copy of the obligation for future reference. The local copy will be used to cross-reference status notifications received from the server that are raised as obligations are fulfilled. Notifications enable the client to check the status of an obligation against its own expectation.

Referring back to Fig.1, this checking process is performed by the *Checker*. The Checker receives notifications from the server’s *Enforcer*, which it records in the *Log* for future reference.

The client monitors the status of obligations on an on-going basis, from the point the obligations become active (normally once the obligation has been accepted by the server and PII exchanged) until the obligation is fulfilled and the obligation has expired. In practice a PII may have many obligations associated with it, and all must be fulfilled before the server’s responsibility for the PII ends. Conversely, in certain situations an obligation may never expire. Consider an obligate that states PII must never be shared with another party. This obligation will exist so long as the server possesses the PII, which could be forever.

At any time the client can check the status of an obligation. The report received back from the server will typically be a Boolean value, and this can be simply communicated to the individual. However, a more likely situation is where the individual asks questions like “Are any obligations that I’ve issued overdue?” or “For this server I’m about to share PII with, are there any outstanding obligations that may affect my trust in the server?” These are more demanding questions to answer. In the case of the first questions, the client must have up to date knowledge of all obligations issued (or be able to obtain one quickly). This may be straightforward to obtain where the information is available locally. The client simply checks current data against expiry data en looks for notifications received. But an obligation of the form "Notify me every time my PII is accessed" is more difficult since the client will be looking for a positive indication from the server ("nil responses don't count"), so the client will need to interrogate the server.

Communicating trust

The status of an obligation forms the basis of the trust indicator. The trust indicator is computed by the client's *Trust Evaluator*. The Trust Evaluator presents a 'simple to understand' indication of the trust status of all outstanding obligations. At the highest level this is a single value indication. In our initial prototype we have chosen to use the traditional traffic light (Red/Amber/Green) indicator, but other means of indication would probably be just as effective. The objective is to quickly alert the individual to any potential problems, and a red light is intended to do just this. Having alerted the individual, the individual is likely to want to know exactly where the problem lies. To help the individual understand why an alert has been raised we provide the facility to interrogate the trust status and reveal first which PII and/or server is affected and then which specific obligation is causing concern. Based on this information the individual can make an informed decision on how to resolve the matter with the server. Where the alert is raised as a result of a query by the individual relating to the anticipated release of PII, the alert will help the individual judge whether to continue with the release, negotiate specific obligations or take another action. Fig. 4 shows our Trust Evaluator in an early prototype.

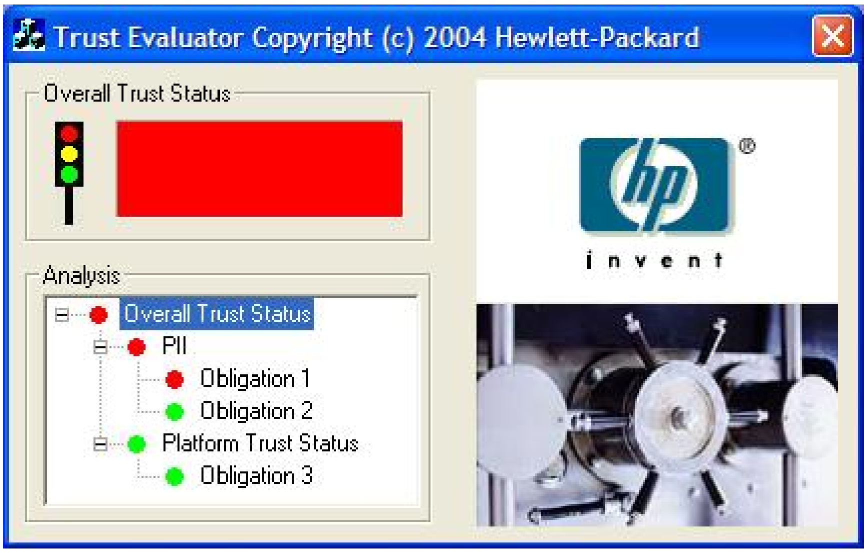

Fig. 4.

6 Obligations in practice

The setting and monitoring of obligations is a clear candidate for automation. Few individuals, except perhaps those that are paranoid about security (or simply interested in it!), will have the time and energy to actively monitor the status of their past interactions. Equally they will be challenged to define meaningful obligations and more so when asked to negotiate them. For a majority of individuals help will be required.

Options we have considered that could help the individual include the provision templates, the involvement of a trusted third party (who may simply provide the templates) and peer-to-peer monitoring.

Obligation templates

Templates will record the obligations that are considered most suitable to particular types of PII and situations. For example, individuals feel that some PII is more sensitive that others. A date of birth is normally more strongly protected than a telephone number. An individual interacting with their government may be less concerned about sharing information than they would with a retailer. Context plays an important part in establishing trust.

Individuals can obviously create their own templates, either beforehand or ‘on the fly’ at the time they share their PII. More likely, they will look for advice from a third party whom they trust to provide reliable and robust templates that have been designed for specific situations and ideally pre-negotiated with likely servers. One could imagine individuals referring to public service organisations like Which?[7], professional organisations (government, financial institution, employer) or even friend and colleagues.

Establishing the template is only the first step. Interpreting the status is another complication for the individual. As before, the individual may simply ask for help from someone they trust. Either status reports would be passed to the trusted party for them to evaluate, or the third party could be called upon to help assess the aggregated output of the Trust Evaluator.

Reputation

A third option is where a group of individuals pool their resources and agree to share templates and the status of obligations. This brings forth two advantages: 1) the peer group can help one another to understand the significance of an outstanding obligation, and 2) they start to build a reputation service where they share opinions about the trustworthiness of servers they have interacted with.

The idea of a 'home grown' reputation service is potentially particularly a very interesting development. Traditionally reputation services have struggled to provide recommendations that can be evaluated against a common criterion. For example, an individual 'scoring' a server has limited value unless others understand (and probably agree with) the underlying scoring rules. Similarly, scoring is influenced by context and personal averseness to risk. We believe our approach has merit because the basis for a trust assessment is clearly defined. Context is set by the stated application that the template is suitable for, and the evaluation is against specific obligations which, again, can be easily re-interpreted by another individual. Of course the opportunity to cheat still remains. Either the server or the client could collude in order to affect the shared evaluation, giving a false impression of a server. However, given that the organisation operating the server has entered into the process with stated good intentions, and we have already explained that our approach is not intended to deal with dishonest organisations, this weakness may not be so significant.

The approach we are proposing differs from traditional reputation-based systems and *webs of trust* in that assurance and reputation are based on the fulfilment of the individual's expectations, on an on-going basis. We do not rely on how other people interpret how their expectations have been fulfilled. Thus we offer a direct measurement of the trust experienced rather than an indirect one.

Obligations for Trusted Platforms

One specific obligation relates to the presence on the server-side of a trusted platform (TP). As briefly mentioned earlier, a TP, for example a platform that hosts a Trusted Platform Module (TPM), has a bearing on the client's perception of trustworthiness. It may be viewed that an organisation that chooses to use TP

[7] Which? is an independent source of expert advice. http://www.which.net/

technologies is demonstrating respect and willingness to abide by any agreement between the two parties.

The client may specify that the server must be a TP. This requirement would be described as an obligation in which the exact form of TP technology used could be negotiated. A criticism of the current 'first generation' TPMs is that it can sometimes be difficult to conclude much about the trustworthiness of the owner or user of a TP from a TPM alone. However, the presence of a TPM does say something about the potential capabilities of the platform which may be useful at an applications level.

The TPM could be used to provide the client user with a signed acknowledgement/confirmation, similar to a signed contract, thereby providing non-repudiation achieved with the help of TPM-controlled signatures

Checking the signature may involve the Trusted Third Party (TTP), possibly the same TTP that endorsed the TPM. The TTP may also play a role in resolving disputes that arise between an individual and an organisation.

7 Related work

The foundation for this work was established in 2003 with the development of a model for a Personal Trust Assistant (referred to in publication as an Intimate Trust Advisor) [CC03]. The concept was to provide individuals with trusted technology that would allow the individual to determine the trustworthiness of their surrounding environment and the entities with whom they interact.

As mentioned earlier, the work of World Wide Web Consortium's Platform for Privacy Preferences (P3P) that defines a policy language for privacy is relevant to our work. P3P is emerging as an industry standard providing a simple, automated way for users to gain more control over the use of personal information on Web sites they visit

Similarly, AT&T's Privacy Bird[8] which reads privacy policies written in the standard format specified by P3P.

We believe that our solution extends the P3P model by creating an *active feedback loop* that enables the individual to play a more active part in understanding how their PII will be used.

8 Future work

Dynamic obligation negotiation

So far we have explained that obligations are negotiated and agreed prior to the release of PII by the client to the server. Some obligations are simple to define and easily monitored. Other obligations are more complex and long lived. Over time obligations that initially seemed appropriate for a given situation could become less so, to the extent that they become irrelevant and need to be replaced, superseded or

[8] For more information about AT&T's Privacy Bird see http://privacybird.com/

redefined. Our current architecture doesn’t allow for the renegotiation of obligations, but this would be a logical extension if the need arises.

9 Conclusions

In this paper we have explain an approach to determining the trust an individual has in an organisation using a technique of direct assessment. We have concentrated on the process required to run on the client-side platform rather than the supporting server-side processes. Our approach assumes that the organisation that (in our case) is receiving the personal information is essentially honest and believes there is merit in demonstrating a respect for the individual’s privacy. We have identified some of the problems associated with disclosing PII without first establishing trust, and illustrated a working solution. We introduced the concept of obligations as a way for an individual to express how their PII should be managed. These obligations also provided the individual with a means for establishing deviations from an agreed policy.

Trusted platforms were introduced to strengthen technical control and underpin the integrity of claims that the organisation / server-side makes relating to ability to conform to the agreed policy.

Finally we identified future work that we intend to pursue.

10 About PRIME

PRIME (Privacy and Identity Management for Europe) is the name of a 4-year project, conducted within the EU 6th Framework Programme, which was launched on 1st March, 2004. Its objective is the research and development of solutions to empower individuals in managing their privacy in cyberspace.

PRIME is performing research in the related areas of ontologies, authorisation and trust model, cryptographic mechanisms, secure and privacy-enhancing end-to-end communications, technologies that enable trust in privacy-enhancing IDM solutions, and in assurance through formal evaluations and seals.

References

[CC03] Cofta, Piotr; Crane, Stephen; Towards the Intimate Trust Advisor; First International Conference on Trust Management; May 2003.

[Pea03-1] Pearson, Siani; et al; Trusted Computing Platforms: TCPA Technology in Context; Prentice Hall; ISBN: 0-13-009220-7; 2003.

[TAO04] Trusted Computing Group (TCG) Architecture Overview. Available for download from the TCG website (https://www.trustedcomputinggroup.org/home) at https://www.trustedcomputinggroup.org/downloads/TCG_1_0_Architecture_Overview.pdf

[P3P] Platform for Privacy Preferences (P3P) Project. http://www.w3.org/P3P/

[MCM04-1] Casassa Mont, Marco; Dealing with Privacy Obligations: Important Aspects and Technical Approaches; TrustBus 2004; http://www.hpl.hp.com/personal/Marco_Casassa_Mont/Documents/Documents.htm

[MCM04-2 Casassa Mont, Marco; Dealing with Privacy Obligations in Enterprises; ISSE 2004; http://www.hpl.hp.com/personal/Marco_Casassa_Mont/Documents/Documents.htm

[Pea03-2] S. Pearson, "A Trusted Method for Self-Profiling in e-Commerce", Trust, Reputation and Security: Theories and Practice, R. Falcone et al. (eds), LNAI 2631, pp. 177-193, Springer-Verlag, Berlin, 2003.

[Pea03-3] S. Pearson, "Privacy-Enhancing Uses of Trusted Platform Technology", Proc. CCCT 2003, vol. 3, ed. H.-W. Chu and J. Ferrer, pp.116-121, IIIS, Florida, July 2003.

[KSG04] Tim Kindberg, Abigail Sellen, and Erik Geelhoed. Security and trust in mobile interactions: A study of user' perceptions and reasoning. Technical Report HPL-2004-113, HP Laboratories, 2004.

Appendix

Types of obligations

This section is not intended to provide an exhaustive list of possible trust obligations, but rather give their flavour.

In general, obligation fit into one of three categories: Short-term/transactions; long-term and on-going. Of these three categories, the group of obligations that are of most interest as far as trust management is concerned are the transactional obligations. In this category we include obligations that are influenced by events and changing circumstances or conditions, though the satisfaction of any category of obligation has a bearing on the trust the individual has in the organisation.

Examples of (trust) obligations include:

- Platform trust status, including presence of a TPM which is correctly endorsed and operating as intended. In addition, the unique identity of the TPM may be requested and cross-referenced against known trusted platforms.
- The server's signed acceptance of the policy containing the obligations. (In practice it is more likely that each individual obligation will be signed. This makes monitoring and analysis much easier, and improves the efficiency of the negotiation process.)
- The timely fulfilment of any obligation that relates to PII, eg. Delete PII after 30 days, notify client each time PII accessed.
- The occurrence of an event known to influence the trust status of the server platform, e.g. TPM failure or suspected attack.

A more sophisticated architecture

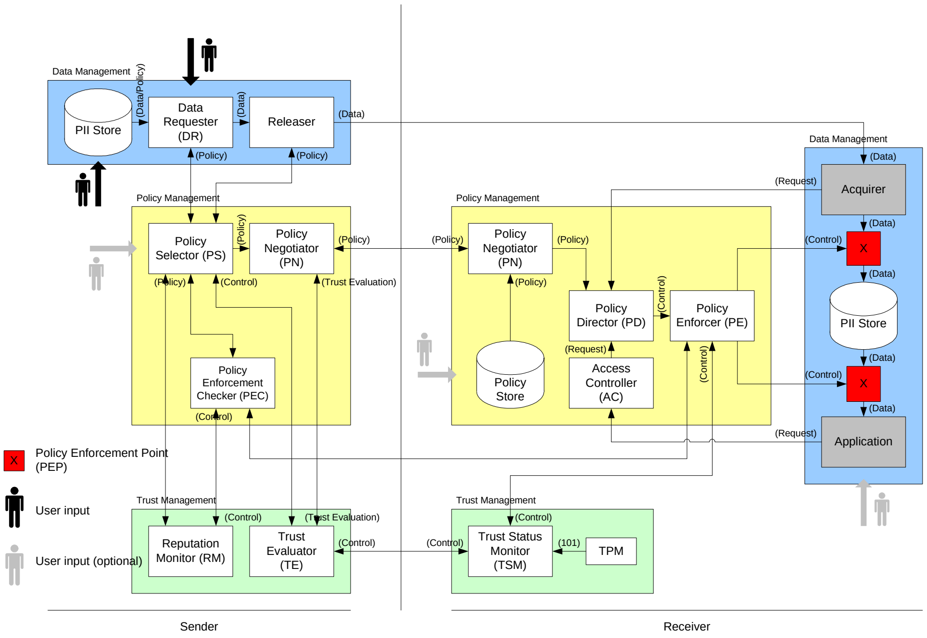

Fig. 5.