Trust Enhanced Ubiquitous Payment without Too Much Privacy Loss

Jean-Marc Seigneur
Distributed Systems Group
Trinity College Dublin
Ireland
+353 1 608 1543
Jean-Marc.Seigneur@cs.tcd.ie

Christian Damsgaard Jensen
Technical University of Denmark
DK-2800 Lyngby
Denmark
+45 4525 3724
Christian.Jensen@imm.dtu.dk

ABSTRACT

Computational models of trust have been proposed for use in ubicomp environments for deciding whether to allow customers to pay with an e-purse or not. In order to build trust in a customer, a means to link transactions using the same e-purse is required. Roughly, trust is a result of knowledge. As the number of transactions increases, the resulting increase in knowledge about the user of the e-purse threatens privacy due to global profiling. We present a scheme (and its prototype) that mitigates this loss of privacy without forbidding the use of trust for smoothing payment by giving the opportunity to the user to divide trust (i.e. transactions) according to context (e.g. location, user's current activity or subset of shops).

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues – *Privacy*; K.4.4 [**Computers and Society**]: Electronic Commerce – *Cybercash, digital cash;* K.6.5 [**Management of Computing and Information Systems**]: Security and Protection – *Authentication*

Keywords

Ubicomp, trust, privacy, payment, context.

1. INTRODUCTION

A ubiquitous computing environment, also called "*ubicomp*" [31] environment or pervasive computing environment, is a space enhanced with functions that unobtrusively support the human inhabiting the space due to its embedded computing and communicating elements. In this paper, we especially focus on one of these functions, namely payment. Electronic coins (e.g. ecash™-type coins) have been used for on-line payment on the Internet and can also be used off-line embedded into handheld devices or smart cards. Computational models of trust have been proposed to be coupled with ecash [7, 26] in ubicomp environments for deciding whether to allow customers to pay with an e-purse or not [5]. Although the latter method may decrease the risk for the vendor to accept invalid coins, this method significantly undermines the privacy-protecting aspect of using ecash coins. From a user's point of view, the nice property of ecash coins is that "a payment transaction leaves no trace about the identity of the user" and "there is no way the bank will be able to link coins that belong to the same user" [26]. However, this anonymity is in conflict with the need to link transactions with a particular e-purse, in order to gather experience/knowledge [17] required to build trust in a particular customer. In the previously mentioned proposal [5], an e-purse would be recognised mainly due to one public key (*PuK*) / private key (*PrK*) pair stored on the e-purse and a transaction would be linked to this e-purse due to digital signature verification. As the number of transactions increases, the increased knowledge about the user of the e-purse threatens privacy due to global profiling. Even if there is no first-hand link between the user's real-world identity and the PuK (e.g. an e-purse is anonymously given to a user), the unique link between all transactions acts as a comprehensive source for data-mining and analysis that can lead to reveal the real-identity of the user. We argue that, even though the vendor who accepts electronic coins for goods have the biggest risk and that the addition of trust can reduce their risk, the privacy-protection aspect of ecash should not collapse so easily due to the addition of trust. In this paper, we describe a scheme (and its prototype) that mitigates this loss of privacy without forbidding the use of trust for smoothing payment by giving the opportunity to the user to divide trust (i.e. transactions) according to context. In fact, the PuK of the e-purse acts as a *pseudonym* for the user. The ordinary definition of a pseudonym is "a fictitious name used when the person performs a particular social role"[1]. In our system, a *pseudonym* is a *public key (PuK) / private key (PrK)* pair. However, instead of providing one unique pseudonym per e-purse, pseudonyms can be created at will *to "speak for"* [1] the same e-purse. In doing so, different profiles may be used in different contexts. As it is hard to link these profiles, it is much more difficult to create an accurate profile of the owner of the e-purse. Depending on context, the right pseudonym is selected. Our prototype provides divisions according to location, the current user's main activity or subset of shops according to user's will. We envision that in pervasive computing environments people would carry a device:

- providing and managing different pseudonyms according to *privacy disclosure policies (PDP)* based on context, trust and user's input

SAC'04, March14-17, 2004, Nicosia, Cyprus

Copyright 2004 ACM 1-58113-812-1/03/04...$5.00

[1] Definition from WordNet Dictionary: http://www.hyperdictionary.com/search.aspx?define=pseudonym

- providing ecash e-purse functionalities coupled with trust-based security

Our e-purse proposal requires that the vendor has an ecash system with adjunct components:

- making decision whether to accept ecash payment according to *payment acceptance policies (PAP)* based on context (e.g. type of good to be bought) and trustworthiness in the *external pseudonym* (i.e. its PrK is not owned) and vendor's input
- dynamically managing trustworthiness of pseudonyms

The remainder of this paper is organized as follows: Section 2 examines a scenario; then, we discuss our approach in Section 3; the prototype is presented in Section 4; finally, Section 5 describes related work.

2. SCENARIO

We consider a mCommerce scenario, where anonymous digital cash resides in a purse on the customer's mobile phone. The anonymous digital cash can be used for payment of small amounts, e.g., public transportation, snacks or groceries at the local corner shop. Associated with every purse is a unique identifier that cannot be traced back to the customer and which the customer can change at will, e.g., different identifiers may be used with different merchants. This identifier allows the merchant to recognise returning customers, without violating customer privacy. Because of the inherent problems of double spending in anonymous offline digital cash, merchants may only accept small amounts from previously unknown customers, but if the digital cash is redeemed by his bank larger amounts may subsequently be accepted. If the customer uses the same virtual identifier in all shops, the local council of commerce will eventually be able to establish a full spending profile for all customers, which they may use for direct marketing or for credit approval. This would be a violation of the customer's privacy. In this scenario, trust is used to reduce the inherent problem of double spending in anonymous digital cash systems, while virtual identities preserve the privacy of customers.

Our solution provides a way to initiate a communication channel with an external pseudonym in proximity where signed messages – called *claims* – can be exchanged with another entity. A *trust-based security framework* (*TSF*) is used for trust decision-making and management. *Bootstrapping* is the process done to initiate the communication channel (e.g. short-range wireless). A high-level view of our solution is depicted in Figure 1.

3. RATIONALES OF OUR APPROACH

This section starts by explaining what we mean by trust and privacy. Then, we justify our approach: why pseudonyms are used and in what sense trust and pervasive computing are of assistance for pseudonym management.

3.1 Trust-based Security Overview

Others have detailed how trust can be formalized as a computational concept [2, 17, 23]. *Trust* in this paper is the human notion of trust, as it is used in the SECURE project [5, 27]. Trust is a means to cope with uncertainty, to engage in an action in spite of the probability of a harmful outcome. This subsection gives an overview of the basic components of a trust-based security framework. The framework should expose a decision-making component that is called when a requested entity has to decide what action should be taken due to a request made by another entity. In order to take this decision, two sub-components are used:

- a trust engine that can dynamically assess the trustworthiness of the requesting entity based on the three sources of trust: observation, recommendation and reputation
- a risk engine that can dynamically evaluate the risk involved in the interaction and choose the action that would maintain the appropriate cost/benefit

In the background, another component is in charge of gathering evidence (e.g. recommendations, comparisons between expected outcomes of the chosen actions and real outcomes…) This evidence is used to update risk and trust information. Thus, trust and risk follow a managed life-cycle. In the remainder of the paper, we use TSF in its broad sense: any TSF can be used (even though the TSF being developed in the SECURE project is an example of an advanced TSF).

3.2 Privacy

Cooley's definition of privacy – the "right to enjoy life and be let alone" [8] – has our preference. Current state of the art of pervasive computing does not properly address security and privacy [6, 21]. For sure, beside the linkability of payment transactions occurring in our scenario, other privacy issues arise in ubicomp environments, e.g. illegitimate monitoring of users. In fact, the pervasiveness of a plethora of invisible sentient sensors makes it harder to solve these other issues. Indeed, Langheinrich [20, 21] said these issues are almost impossible to solve technologically, especially when determined attackers are in action. In our system, we assume that no external scheme is used to recognize the users; the only scheme used is based on our

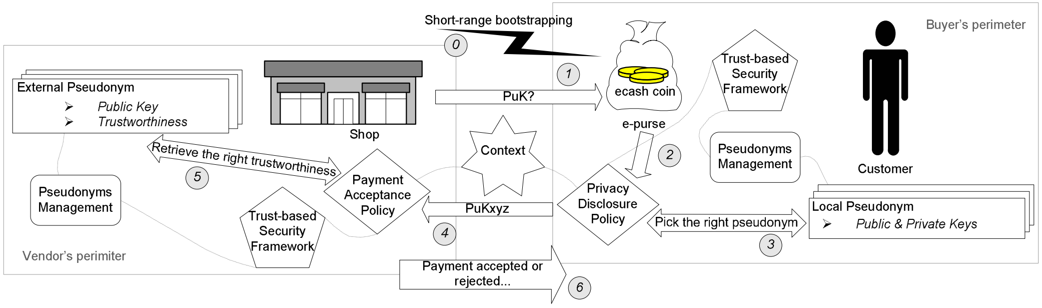

Figure 1. High-level view

pseudonyms. The same kind of assumption is done for other privacy-protecting mechanisms in ubiquitous computing environments. For example, ecash in its common form (i.e. without adjunction of trust) would also not defeat vision recognition of customers of both video shops if the video club holding shares the same centralized recognition database. Another example is the assumption made for the privacy protection mechanisms developed for Gaia smart spaces: "We also assume that the spaces supporting our privacy system would not contain surveillance cameras or voice recognition devices, otherwise, users will have to take additional physical precautions to protect their privacy, like wearing masks or staying silent!" [3].

Langheinrich has proposed six useful principles for guiding the design of privacy protecting mechanisms in ubicomp [21]. We briefly explain how they are applied in our system:

1. *Notice*: Intuitive interfaces are provided to help users to retrieve information on past transactions (see Subsection 4.3); the "model of switching identities" [14] or "faces" [22] based on context is said to be easily understood by users, so users should understand the concept of switching pseudonyms.

2. *Choice and Consent*: Users are in control of their PDP (detailed in Subsection 4.1); trust (and transactions) can be divided according to the user's choice.

3. *Anonymity and Pseudonymity*: This paper is entirely in line with this principle because a key goal of our system is to provide and dynamically manage multiple pseudonyms to make linkability harder.

4. *Proximity and Locality*: PDP can be based on location, and bootstrapping should be done in proximity; it is possible to only link transactions occurring in a restricted area instead of the whole planet.

5. *Adequate Security*: Many constraints (detailed in Subsection 4.2) have been placed on the bootstrapping process to make it more secure; we use asymmetric cryptography for authentication and the same trick as in PGP-like [32] system could be used to get confidentiality by first signing the claim's content, then encrypting both claim's content and signature with a symmetric session key and finally sending encrypted content and session key encrypted with the receiver's PuK.

6. *Access and Recourse*: Ecash provides a means for either the payer or the payee to reveal for which good the payment transaction was made and for which amount; in our prototype, the user can retrieve which transactions happened according to context but access to second use of transactions by the vendor is beyond the scope of the paper.

3.3 The Necessity of Pseudonyms

Before retrieving trust from the TSF, interacting entities must be recognized. In this paper, the approach for communication between pseudonyms is to send claims, i.e. digitally signed messages. The sender pseudonym is called a *claimant* and is recognised by the target e-purse on the basis of correctly signed claims. The ability to recognise another entity seems to be sufficient to establish trust in that entity based on past experience. The minimum requirement is a local reference for the formation of trust, which is in turn managed by other components in the TSF. According to the privacy protection principle of "*collection limitation*" [21], data collection should be strictly restricted to mandatory required data for the purpose of the collection. Our requirement is to retrieve the trustworthiness in customers and not their real-world identity. This is one reason why pseudonymity is necessary. Recalling the process of trust makes apparent the fact that privacy is at stake in trust-based systems. In order to be able to use trust in one entity, the first step is to retrieve the level of trust – result of knowledge [17] and evidence analysis – in this entity. Thus, trust, after all, relies on profiling; the more information the better, in order to guess likely behaviour. That is why it is important that we provide pseudonyms, i.e. levels of indirection between trust and real-world identity. In trust-based systems, there must be a mechanism that can dissociate users from their actions [19]. These systems are fuelled with information that aims at building more and more accurate profiles along the time. Any link with the real end-user would change this information into really sensitive personally identifiable information (PII).

When using pseudonyms, a means must be present to prevent users from taking advantage of the fact that they can create as many identities as they wish [12]. A TSF ensures that trade-off with cost and benefit is respected due to its risk analysis component. So, there is no need for the underlying assumption that a potential countermeasure against fraud is to be able to retrieve the real-world identity of the user and to sue this person. For example, the latter assumption is implicitly apparent in ecash: "if a user is able to manipulate its device such that some coins are used more than once, the protocols are such that the identity of the double spender can be computed" [26]. In contrast, the TSF is in charge of ensuring that fraud cannot happen: we do not rely on the ability to sue the real-world identity, but rather that the appropriate net cost/benefit is maintained.

It is worth noting that pseudonyms used for only one transaction and anonymity do not provide linkability between interactions. Trust is built by linking interactions over time and recommendations between entities.

"*Pseudonymous Identification*" [18], that we provide in our solution, appears to be the right solution for protecting privacy in trust-based systems and achieving the right level of privacy and trust. We underline that users should be in control of which key is created on their e-purse. The e-purse should be anonymously issued to the user without keeping a link with the real-world identity of the user.

3.4 Why Pseudonym Management is Eased by Trust and Pervasive Computing

Trust, as with privacy, is dynamic and evolving interaction after interaction. The intrinsic property of trust to evolve autonomously also improves the capability of our system to adapt automatically to context and to auto-configure [28]. Privacy is a constant interaction where information flows between parties [15, 24]. Privacy expectations vary [4, 15] and depend on context [18]. The advantage of pervasive computing environments is that computing entities are context-aware – environmental information that is part

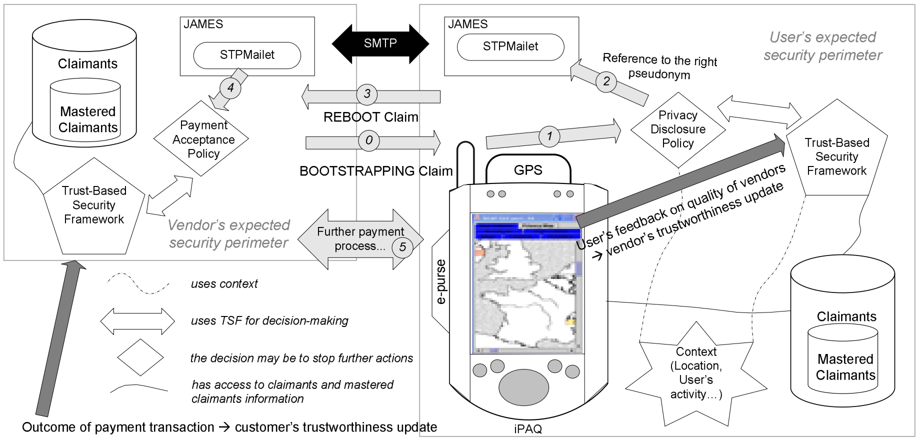
Claimants
Mastered Claimants
JAMES
STPMailet
SMTP
JAMES
STPMailet
Reference to the right pseudonym
User's expected security perimeter
4
3
REBOOT Claim
2
Payment Acceptance Policy
0
BOOTSTRAPPING Claim
1
Privacy Disclosure Policy
Trust-Based Security Framework
GPS
Trust-Based Security Framework
User's feedback on quality of vendors → vendor's trustworthiness update
Vendor's expected security perimeter
Further payment process...
5
e-purse
uses context
uses TSF for decision-making
the decision may be to stop further actions
has access to claimants and mastered claimants information
Claimants
Mastered Claimants
Context (Location, User's activity…)
Outcome of payment transaction → customer's trustworthiness update
iPAQ

Figure 2. Implementation diagram

of an application's operating environment can be sensed by the application [9]. So, privacy policies are improved and can be based on context [11, 14, 20, 22]. Our prototype allows the user to define PDP dependent on context as explained in Subsection 4.1. There is a strong requirement for our solution to be unobtrusive, or at least convenient to use, because, in addition to being one of the core features of pervasive computing, the cost of privacy protection cannot be too high [21]. There is always a cost-benefit analysis before deciding to invest in security features. If we draw a parallel with current online pseudonym services based on disposable email addresses, the two main flaws are:

- the time that must be spent by users creating and managing pseudonyms
- the difficulty of bootstrapping in special scenarios (e.g. no computers are at hand at time of bootstrapping, different and complex email addresses are difficult to remember and exchange orally)

In addition to easing management due to trust and context-awareness, the second flaw is solved if we assume that users have always at hand a means (e.g. a smart device) that allows them to conveniently and securely bootstrap communication with other entities (as explained in Subsection 4.2). The latter assumption is indeed supported by pervasive computing.

4. THE PROTOTYPE

In this section, we detail the implementation of our system (depicted in Figure 2). The first subsection focuses on policies. Then, the bootstrapping process between two principals is described. After that, we focus on how to ease user's understanding of the system with intuitive interfaces. The end of the section goes over further implementation aspects.

4.1 Policies

In this paper, we intend neither to focus on one particular TSF nor to present an advanced formal TSF for payment decisions. In contrast, others [5] work towards the creation of such formal TSF. We rather introduce how to protect privacy even when a TSF is used for payment acceptance.

The first type of policy provided is the Payment Acceptance Policy (PAP). In a TSF, scenarios where economic models can be used ease the risk analysis due to meaningfulness of cost/benefit comparison in terms of money. So, let us assume that the trustworthiness of people for being good payers is managed by a TSF and represented by a balance in Euro. The main abbreviations used in our trust-based decision making are:

- *b,d,u*: belief, disbelief, uncertainty; a trust value is a triple (b,d,u) where b,d,u are all in Euro
- *BA*: each claimant has a kind of bank account composed of three fields: BAb, BAd, BAu; all in Euro; BA(BAb, BAd, BAu)

When the outcome of each transaction is known, the trustworthiness in the claimant is updated according to the trust value associated with the outcome. For vendors, depending on how many electronic coins were valid, a simple method to calculate the trust value can be: *b=total value of valid coins; d=total value of invalid coins; u=0*.

The new claimant's BA can simply be updated as follows: *NewBA=(OldBAb+b,OldBAd+d,OldBAu+u)*. A simple PAP may be to test that the claimant has a positive BA by using the main sources of trust (direct observations, recommendations and reputation): *if (BAb > (BAd + (BAu/2))) then accept; else reject*. At the time of this test, the TSF may retrieve recommendations from other entities about the trustworthiness of the sending entity. A recommendation consists of the claimant's BA on the recommender side. How to retrieve and combine recommendations with the local claimant's BA depends on the TSF used. The TSF should also take into account that some entities are more or less good recommenders. The SECURE project gives an example on how to implement recommendation mechanisms in a TSF. Similarly, for customers, the trustworthiness in vendors for selling high-quality products and services can be based on the same simple process. If Alice decides to buy a DVD player in a video club, she might want to find out if this video shop is known to sell good quality products. In this case, the method to calculate the trust value is slightly modified to take into account that sometimes users are not certain about the quality of the product. First the users give a *quality feedback vector (QFV)* on the outcome of the transaction (e.g. good or bad product or service) composed of three percentages: *(QFVb=percentage that the outcome was beneficial, QFVd= percentage that the outcome was harmful; QFVu=percentage of* *uncertainty on the benefice/loss assessment of the outcome)*. Then, the trust value is calculated given the *amount of the transaction (AT)* in Euro: *b=QFVb*AT; d=QFVd*AT; u=QFVu*AT*.

The first part of the Privacy Disclosure Policy (PDP) can take advantage of the fact that vendors are more or less trustworthy. While walking in the street, the customer's e-purse could receive a claim from a nearby vendor. A threshold based on the trustworthiness in the vendor could be set in order not to disclose a pseudonym to untrustworthy claimants. A simple example could be: *if((VendorBAb-(VendorBAd+(VendorBAu/2)))>100€) then continue PDP; else no reply*. Thereafter, the model of "*privacy boundary"* [16] describes what kinds of context the remainder of PDP could be based on (e.g. activity or location). By basing PDP on activities, the common definition of pseudonym (in Section 1) is really close to what is done in our system: the pseudonym is selected when the person performs a particular social role. However, the default PDP implementation is location specific. The user can set up squared privacy areas starting from the home location. It is also possible to change the size of the squares. If the area is increased, the resulting pseudonym is potentially linked to more trust, but privacy is loosened. If the area is decreased privacy is tightened, but less trust may be retrieved. For example, users can reduce the area to 50m which would allow them to use different pseudonyms in different shops (or 500m to allow different pseudonyms in different parts of town). The user may also select another mode, called *One-Time*, which creates a new pseudonym each time a new claimant is met.

4.2 The Bootstrapping Process

Bootstrapping is initiated by the claimant in order to avoid tracking of users broadcasting their pseudonyms. It is the choice of *user-driven location tracking*, which is known for having a higher level of privacy [30]. The initiation should be based on short-range wireless technology that does not convey obvious identification clues for tracking. For now, our prototype simulates broadcast programmatically. In doing so, different pseudonyms can be exhibited depending on the location of bootstrapping. In future work, richer context may be used to choose the *mastered claimant* – it is a local pseudonym thus its PrK is owned – according to PDP. In fact, a claim contains special keywords used at our application level to apply further bootstrapping or payment actions:

- *BOOTSTRAPPING* means that the PDP is applied and returns either:
    - the *mastered claimant* to be associated with the new claimant (e.g. if there is already a pseudonym for the current privacy area),
    - the fact that a new pseudonym must be created
    - or that no bootstrapping should be done (or redone) with this claimant (e.g. the *Stealth Mode* option of the GUI is ticked or the trustworthiness is under a threshold)

  The eventual pseudonym is then sent back to the claimant as a claim containing the REBOOT keyword. The list of mastered claimants and claimants is updated accordingly in the GUI.
- *REBOOT* means that the claim contains the pseudonym of a new claimant which replied to a previous BOOTSTRAPPING claim. The list of mastered claimants and claimants is updated accordingly in the GUI. The PAP is applied on the vendor's side when this claim is received. (The description of the complete protocol for integrating ecash payment with a TSF is beyond the scope of the paper.)

4.3 Intuitive Maps

In order to ease pseudonym management, we provide two maps. The map represents Europe and can be zoomed in and out. For now, the user's location is changed by moving a pink circle on the map; a GPS module would dynamically change the position of this circle. The user's home is displayed as a green rectangle. The first map is the *privacy map*, where privacy areas covered by pseudonyms are displayed. By clicking on each zone, information about claimants (i.e. vendors) bootstrapped with the pseudonym associated with the zone and their associated information (e.g. BA or content of past transactions) can be displayed. The goal is also that the user, by using queries or clicking on the map, can easily retrieve any information about any claimants or mastered claimant (time of bootstrapping, trustworthiness, claimants associated to mastered claimants, textual information entered by the user…) as well as set of entities (e.g. all claimants bootstrapped in a specific area). In Figure 3, the zones covered by the user's pseudonyms are represented in the GUI as rosy squares.

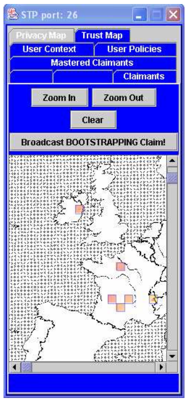

Figure 3. Privacy map

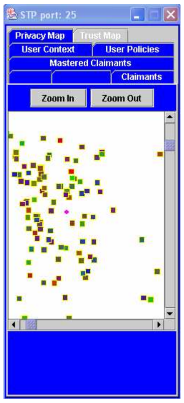

Figure 4. Trust map

The second map is called the *trust map*, where all vendors bootstrapped so far are displayed as small rectangles. The trick is to change their color according to their trustworthiness, which we argue helps the user to quickly understand the trustworthiness of vendors (as depicted in Figure 4) as well as areas. We use Java opaque sRGB color with the specified red, green, and blue values in the range (0-255). Green(0,255,0) means full trust. Red(255,0,0) means full distrust. Blue(0,0,255) means full uncertainty. In doing so, we argue to obtain the metaphor of red/green traffic lights and the cold aspect of blue grasping the feeling of uncertainty. Black(0,0,0) is a special case, which means that no trust information is known on the pseudonym. The three basic colors are combined according to the following algorithm:

*RectangleColor((BAd/(BAb+BAd+BAu))*255; (BAb/(BAb+BAd+BAu))*255; (BAu/(BAb+BAd+BAu))*255).*

4.4 Further Implementation Aspects

For now, the prototype, based on an iPAQ (even though the simulation of the system currently runs on a laptop) and assumed to run a TSF (e.g. the SECURE TSF), uses two main sources of context: location and activity of the user. The distance in km from the user's home in X,Y coordinates is simulated but the device could be equipped with a GPS. The current activity of the user consists of its textual profile (e.g. *on holidays*). The area covered by a pseudonym can be set. For example, Alice may have two pseudonyms: *PuKAlice1* used in the area of her native town and associated with the main activity *Holidays*; *PuKAlice2* used in another town associated with the main activity *Working Life*. All textual information on user's activity is simulated.

The prototype is Java-based (J2SE). Any symmetric key algorithm and secure hash available in Java can be used. Due to the requirements of using SMTP in the bigger parent project of this work, claims are currently exchanged over SMTP. The design of our systems allows the use of other communication channels that would be more appropriate (as discussed in Subsection 4.2). The claims are Java objects serialized in multipart MIME messages. At the heart of our system is a modified mail server named *STP* (for *Secure Trustworthy Payment*) and based on the Java Apache Mail Enterprise Server (JAMES) [13]. JAMES provides an SMTP server and a Java API, called the mailet API, to write Java code to process incoming email messages. A specific mailet, called the *STPMailet*, has been created and is launched when JAMES starts. This mailet processes any emails sent to the JAMES server. A Java GUI is linked with the Java objects instantiated by the STPMailet. Figure 2 depicts the different components in our implementation.

5. RELATED WORK

Others [12, 18] have presented how pseudonyms can be used for privacy protection and shown that different levels of pseudonymity and configurations exist. Their work is valuable to choose the right type of configuration and pseudonymity. Complete privacy management framework for ubicomp is needed [16, 20-22], and this encompasses our system. Our system could be extended by reusing techniques aiming at controlling how privacy information (in our case payment transactions) is used after disclosure. For example, we may try to rely on the fact that collectors of private information would respect the privacy policy that was specified by the user at time of collection [20] or to attach these policies to private data [16]. Previous work on identity management in ubicomp environments [14, 22] demonstrated that the model of switching identities according to context is appealing and meaningful for users. Different TSFs have been used for sharing personal information in ubicomp environments [11, 29]. However, to our knowledge, this work is the first allowing for the use of a TSF as part of pseudonym disclosure policies. A potential attack on systems, where pseudonyms can be created at will and trust is used, is the Sybil attack [10]. We may mitigate the latter attack by keeping a positive net cost/benefit due the TSF (as said in Section 3.3). The security mechanisms developed for Gaia smart spaces [6] aim at preserving privacy in ubicomp environments and indeed use context information. They focus on preserving privacy location due to a circuit of routers called Mist [6]. Their environment, where administrators are present to write policies and agreements on the infrastructure are possible (e.g. the hierarchy of Mist routers) differs from the environment of our scenario, which seems to be more ad-hoc, with no administrator per se, where the TSF is in charge of trust evolution. The idea behind self-profiling [25] is that users are in control of their profile. At will, the user can create pseudonymous identities, who do not reveal their real identity, but the certification authorities, who manage the mapping between identities and real users, can still know who is behind the identity. They propose temporary identities for protecting privacy. Our system extends the concept to other forms of context whilst focusing on trust enhanced payments. Another related work, although this one also focuses on recommendation, is the OpenPrivacy platform [19]. The user can create many pseudonyms linked with specific information. Other privacy user interfaces [22] may be added to our solution to further ease pseudonym management.

6. CONCLUSION

Even though the vendor who accepts electronic coins for goods have the biggest risk and the addition of trust can reduce their risk, the privacy-protection aspect of ecash should not collapse easily due to the addition of trust. We argue that the proposal to use only one pseudonym per e-purse does not fulfil the above requirement. Instead, we propose and demonstrate a system where pseudonyms can be selected according to context, trust and user's choice. As it is hard to link these pseudonyms, it is much more difficult to create an accurate profile of the owner of the e-purse and eventually reveal the real-world identity of the user. We present arguments for the necessity of pseudonyms in trust-based systems. However, the cost of privacy protection cannot be too high. So, we explain why pseudonym management is eased by trust and pervasive computing, especially due to context-awareness. We further improve convenience of management with intuitive maps. The importance of privacy protection during bootstrapping is underlined. We detail how payment acceptance policies and privacy disclosure policies can take into account trustworthiness of external pseudonyms. Our solution achieves a trade-off: the benefit of adjunct trust is obtained without too much privacy loss. The study of the feasibility of negotiating the *right* trade-off is our next objective.

7. ACKNOWLEDGMENTS

This work is sponsored by the European Union, which funds the IST-2001-32486 SECURE project.

8. REFERENCES

[1] M. Abadi, "On SDSI's Linked Local Name Spaces", in Journal of Computer Security, vol. 6(1), pp. 3-21, 1998.

[2] A. Abdul-Rahman and S. Hailes, "A Distributed Trust Model", in Proceedings of the 1997 New Security Paradigms Workshop, pp. 48-60, ACM, 1997.

[3] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing Through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments", in the International Conference of Distributed Computing Systems (ICDCS 2002), Vienna, Austria, pp. 65-74, 2002.

[4] B. D. Brunk, "Understanding the Privacy Space", in First Monday, vol. 7, no. 10, Library of the University of Illinois, Chicago, 2002, http://www.firstmonday.org/issues/issue7_10/brunk/index.html.

[5] V. Cahill, E. Gray, J.-M. Seigneur, C. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. d. M. Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen, "Using Trust for Secure Collaboration in Uncertain Environments", in Pervasive Computing July-September 2003, vol. 2(3), IEEE, 2003.

[6] R. Campbell, J. Al-Muhtadi, P. Naldurg, G. Sampermane, and M. D. Mickunas, "Towards Security and Privacy for Pervasive Computing", in Proceedings of the International Symposium on Software Security, Keio University, Tokyo, Japan, 2002.

[7] D. Chaum, "Achieving Electronic Privacy", in Scientific American, vol. August, pp. 96-100, 1992.

[8] T. M. Cooley, "A Treatise on the Law of Torts", Callaghan, Chicago, 1888.

[9] A. K. Dey, "Understanding and Using Context", in Personal and Ubiquitous Computing Journal, vol. 5 (1), pp. 4-7, 2001.

[10] J. R. Douceur, "The Sybil Attack", in Proceedings of the 1st International Workshop on Peer-to-Peer Systems, 2002, http://research.microsoft.com/sn/farsite/IPTPS2002.pdf.

[11] J. Goecks and E. Mynatt, "Enabling Privacy Management in Ubiquitous Computing Environments through Trust and Reputation Systems", in Proceedings of the 2002 Conference on Computer Supported Cooperative Work, ACM, 2002.

[12] R. Hes and J. Borking, "Privacy Enhancing Technologies: The Path to Anonymity", ISBN 90 74087 12 4, 2000, http://www.cbpweb.nl/downloads_av/AV11.PDF.

[13] JAMES, "JAMES: Java Apache Mail Enterprise Server", Website, http://james.apache.org.

[14] U. Jendricke, M. Kreutzer, and A. Zugenmaier, "Pervasive Privacy with Identity Management", in Proceedings of the Workshop on Security in Ubiquitous Computing, Ubicomp 2002.

[15] X. Jiang, J. I. Hong, and J. A. Landay, "Approximate Information Flows: Socially Based Modeling of Privacy in Ubiquitous Computing", in Proceedings of the 4th International Conference on Ubiquitous Computing (Ubicomp 2002), LNCS 2498, pp. 176-193, Springer-Verlag, Berlin Heidelberg, 2002.

[16] X. Jiang and J. A. Landay, "Modeling Privacy Control in Context-Aware Systems", in 1(3), pp. 59-63, IEEE Pervasive Computing, 2002.

[17] A. Jøsang, "The right type of trust for distributed systems", in Proceedings of the 1996 New Security Paradigms Workshop, ACM, 1996.

[18] A. Kobsa and J. Schreck, "Privacy through Pseudonymity in User-Adaptive Systems", in ACM Transactions on Internet Technology, vol. 3 (2), pp. 149-183, 2003.

[19] F. Labalme and K. Burton, "Enhancing the Internet with Reputations", 2001, www.openprivacy.org/papers/200103-white.html.

[20] M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments", in Proceedings of Ubicomp 2002.

[21] M. Langheinrich, "Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems", in Proceedings of Ubicomp 2001: Ubiquitous Computing: Third International Conference, LNCS 2201, pp. 273-291, Springer Verlag, Heidelberg, 2001.

[22] S. Lederer, C. Beckmann, A. K. Dey, and J. Mankoff, "Managing Personal Information Disclosure in Ubiquitous Computing Environments", Intel Research, IRB-TR-03-015, 2003, http://www.intel-research.net/Publications/Berkeley/070920030922_139.pdf.

[23] S. Marsh, "Formalising Trust as a Computational Concept", PhD Thesis, Department of Mathematics and Computer Science, University of Stirling, 1994, http://citeseer.nj.nec.com/marsh94formalising.html.

[24] E. M. Noam, "Privacy and Self-Regulation: Markets for Electronic Privacy", 1997, http://www.ntia.doc.gov/reports/privacy/selfreg1.htm#1B.

[25] S. Pearson, "Trusted Agents that Enhance User Privacy by Self- Profiling", Technical Report HPL-2002-196, Hewlett-Packard, 2002, http://www.hpl.hp.com/techreports/2002/HPL-2002-196.html.

[26] B. Schoenmakers, "Basic Security of the ecash Payment System", in Bart Preneel et al. (eds.) Computer Security and Industrial Cryptography: State of the Art and Evolution, ESAT Course, Leuven, Belgium, LNCS series, Springer-Verlag Berlin Heidelberg., 1997.

[27] SECURE, "Secure Environments for Collaboration among Ubiquitous Roaming Entities", Website, http://secure.dsg.cs.tcd.ie.

[28] J.-M. Seigneur, C. Damsgaard Jensen, S. Farrell, E. Gray, and Y. Chen, "Towards Security Auto-configuration for Smart Appliances", in Proceedings of the Smart Objects Conference, 2003, http://www.grenoble-soc.com/proceedings03/Pdf/45-Seigneur.pdf.

[29] B. Shand, N. Dimmock, and J. Bacon, "Trust for Ubiquitous, Transparent Collaboration", in Proceedings of the 1st IEEE Annual Conference on Pervasive Computing and Communications.

[30] P. Viswanathan, B. Gill, and R. Campbell, "Security Architecture in Gaia", Technical Report UIUCDCS-R-2001-2215 UILU-ENG-2001-1720, Universiy of Illinois at Urbana-Champaign, 2001, http://citeseer.nj.nec.com/viswanathan01security.html.

[31] M. Weiser, "The Computer for the 21st Century", Scientific American, 1991.

[32] P. R. Zimmermann, "The Official PGP User's Guide", ISBN 0-262-74017-6, MIT Press, 1995.