

# An Approach for Secure Usability for Ambient Intelligence Environments

Sebastian Höhn and Rafael Accorsi  
University of Freiburg, Dept. of Telematics  
Friedrichstr. 50, 79098 Freiburg, Germany  
{hoehn, accorsi}@iig.uni-freiburg.de

Mathias Maier  
mathias.maier@gmx.net

## Abstract

Many transactions of our daily lives depend on computer systems. This means that newly deployed systems must be carefully designed in order to be correctly used by laypersons. The scenario we present encompasses a large number of sensors in a shopping environment. Since these environments are publicly accessible and the sensors are able to collect data without the users' notice and hence without their explicit consent, privacy issues play an important role. It is infeasible to prevent these highly dynamic systems from collecting data, we propose transparency for data collection, storage and processing as a possible solution.

Reliable data of the systems' behavior are an essential part of an architecture for transparency. To this end, secure logging mechanisms are of foremost importance. Furthermore, it is still an open question how this transparency can be presented to the users in a simple and intuitive fashion.

## 1 Introduction

Since computer systems became networked, dynamic and embedded in everyday objects the complexity of administration and usage exponentially increased. Together with this evolution, we are losing the ability to protect our privacy while interacting (perhaps even implicitly) with modern ubiquitous computing systems. We are obliged to deprive ourselves of the possibility to choose whether or not to disclose certain information (e.g. information about our whereabouts) as computer networks collectively and invisibly collect data with their various sensors.

At present the number of environments where ambient intelligence environments track humans and dynamically interact with their users is still small. Nevertheless these systems are entering our lives in many different areas. Several application scenarios show that all the technologies are at our hands and that they can readily be combined to provide new services

and additional assistance. Among the frequently cited scenarios are personal shopping assistants, home automation scenarios and assisted living for elderly or impaired people [12, 5, 17].

Providers of personal shopping assistants claim that customer acceptance of the involved technology is good [25], although they had to face calls for boycott from privacy activists and organizations when a hidden RFID-chip was discovered in the customer card [16]. This shows that systems of this new type have to be designed with special care for security and privacy in mind in order to be useful and usable to users without background knowledge without being enigmatic and therefore frustrating.

In this article we first present a scenario involving a personal shopping assistant based ambient intelligence technologies. We point out privacy issues which arise from collecting data invisibly and without the customers' consent and which counteract the otherwise great opportunities for the retailers' customer relationship management. We then propose the usage of *log views* to create more transparency as a means to lessen the customers' frustration by the ambient systems when they act in an unexpected way or when they violate the customers' privacy policies. Log views provide the customers with evidence which has by definition to be unaltered. To this end, we deliver the necessary technological tool of *secure logging*, which we describe in detail in chapter 4. We conclude with reasoning why transparency will lead to a win-win situation in the privacy domain for retailers and customers.

## 2 Scenario

To visualize the technologies and processes involved in our scenario we present a short introduction to the supermarket of the future. Some other scenarios are presented in [3].

Because Alice is frequently visiting this supermarket, she owns a combined customer and rebate card. During her visits to the store she is interested in the convenience provided by the innovative technological concepts. She is not really interested in the security and privacy enhancing technologies deployed in the super market. This behavior is common to 85% of the people using the new technologies [15].

Bob, on the other hand, cares a lot for his privacy preferences and thus, refuses to get the customer card. He is not interested in sharing his personal information for a small rebate.

When entering the shop both Alice and Bob get themselves a shopping cart with a personal shopping assistant attached to it. The shopping assistant is a computer equipped with a touchscreen, RFID and barcode scanners. It is linked to the backstore network via an integrated wireless network module, and the location of the cart can be tracked at any time to provide the customers navigation information within the store. As neither Alice nor Bob carry their own devices with them, they do not connect to the assistants open interface via near field communication. This interface enables the transmission of shopping lists or special offers directly to the shopping assistant. Near field communication interfaces are also used to replace customer cards with the users' devices and for a new mobile payment system [22].

Alice presents her customer card to the shopping assistant and is consequently provided with the shopping list and some personalized special offers. She has uploaded the shopping list to her personal page on the store's Internet portal.

Bob enters his shopping list directly into the assistant's interface. He will do this, because he is not familiar with the store and wants the shopping-cart to guide him to the products on the list as quickly as possible.

While the electronic shopping list is a nice feature for regular, and a help for one-time customers in finding products faster, it is a very powerful instrument for the retailer. The electronic shopping list enables the analysis of the products that each of the customers intended to buy when entering the store, and which products they bought in addition. Together with the possibility of being able to locate the customer in the store via the shopping cart, and the resulting possibility of personalized marketing with info-displays on the shelves, it is possible for the CRM-system to measure the effectiveness of different kinds of marketing techniques on any particular customer. Furthermore, it is possible to dynamically modify marketing strategies in the customers' direct surroundings in order to positively influence their inclination to buy.

Alice is standing in front of the dairy products now, and since she regularly buys the same brand of yogurt, the Electronic Shelf Labeling which is connected to the store's central pricing system updates the label and displays a discounted price for her. She happily takes a few cups of yogurt and scans them.

Bob observed Alice buying the seemingly cheap yogurt and decides to buy some cups of it, too. When he scans the yogurts with his shopping assistant however, that shows the regular (and therefore higher) price, as does the label on the shelf now.

Alice knows that she has earned the right to buy yogurt for a discounted price, and enjoys the effect of the labels "magically" presenting the price of the current special offer. Bob does not know this and is annoyed, because he has to pay more for the same product in the presumably same situation. Things happening without apparent reason are very welcome, if beneficial. However, it is also very frustrating if they show a discriminating nature [23]. It is strictly necessary to help customers understand why someone gets discounted prices on certain goods, while others do not.

Alice moves on to the frozen meat. The info terminals keep flashing advertisements for vegetarian products at her. They do that ever since she experimented with a vegetarian diet. It annoys her to be reminded of the bad experience now that she enjoys eating meat again.

Generating profiles of customers and classifying them into groups for marketing is sensible from the retailer's point of view. Yet it is a tricky business since it invades the customers' privacy and over-fitting is prohibitive. In this case, Alice was classified as a vegetarian on mistake.

Bob has finished his shopping. However, he finds that he does not want to buy the yogurt any more now that he knows he has to pay more than he thought at first. Being annoyed he doesn't bother to return to the shelf with the dairy products and puts his yogurt on the cheese shelf.

The shelves constantly scan the RFID-chips of the products to report shortages and

misplaced products. The computer system in the shop can relate a misplaced product to the customer standing in front of the shelf with his cart or if Bob has already scanned the product for the automated checkout procedure directly with his shopping cart. This could be abused by the retailer (for example to punish customers for undesirable actions). Although Bob thinks that he is anonymous, the system could have analyzed his movements in the shop and the items he bought over time. Based on these statistics the system could single out his individual patterns as matching to some customer who regularly misplaces products. The system cannot make a correct decision whether it actually was Bob, or whether he was unlucky to coincidentally match the movements of a customer notoriously misplacing products.

Bob proceeds to the checkout. He hands over the shopping assistant to the cashier. He is lucky that he does not have to place the contents of his shopping cart on the conveyor belt. There are random tests once in a while to ensure that all the contents was scanned by the customer and therefore registered with the shopping assistant. Before leaving the store he passes a gate which deactivates anti-theft protection.

Alice checks out, too. She pays with her cash card and takes the detour to the RFID deactivator. In order to erase the data from the RFID chips she has to place each individual tagged product on the deactivator and press a button.

It is not possible for the customers to directly perceive if an RFID chip is readable or which information it contains. Neither is this possible for anti-theft protection. While for the latter non-observability is essential, it is a problem in the former: Alice does not want to carry tagged products out of the shop. She has to trust the retailer that the deactivator works correctly and that it is not manipulated. She would lose her trust in it immediately, if she heard of any malfunction of the device. If she was able to perceive the state of the RFID, she would not need to trust anything or anyone.

### **3 Observability as a Basis for Usability**

The scenario presented in the previous chapter illustrates the characteristics of such highly dynamic systems in an informal way. In the following chapter we will take a closer look at the characteristics of the ambient intelligence environments:

1. These systems are *open*, i.e. participants are able to join and leave the systems at their will.
2. They are *context sensitive*, i.e. they continually adapt to the special requirements of the participants and the technical necessities.
3. They have *autonomous properties*, i.e. there is a vast amount of automated tasks in order to keep these systems in a sane state
4. They are *invisible*, i.e. the users are not directly involved in observing and controlling the systems.

If we summarize these properties and consider the challenges for privacy and usability, we realize that all of these contribute to a *loss of control* realized by the users.

Users cannot observe who is participating in the different stages of communication and to whom they transmit their information. Data are sent over different networks and services of a plethora of providers are combined in order to establish a highly dynamic system. In the supermarket scenario above not all the services will be provided by the owner of the shop. The owners certainly will add new services from specialized service providers, for example, to offer deferred payments. This shows that there exists no well defined area of trust within these systems.

The system's reaction to context makes the interactions between users and systems implicit. These implicit interaction models act like a two-edged sword: on the one hand, they facilitate the users the performance of their tasks and the achievement of their goals. On the other hand, they are not very well suited to support direct control by the users. It is a proactive interaction model that must correctly anticipate the users' intentions. But how can the users' desires correctly be anticipated? And how can the users *observe* what has been anticipated and what decisions have been made on their behalf? If this anticipation fails and the users can realize this failure they become annoyed and directly feel the loss of control.

The same is true for autonomous properties. They are necessary in order to enable such a complex technical infrastructure, but failure of these mechanisms will not be accepted by the users as well as the administrators, in this case. Perhaps the most important aspect for end users is the invisibility of many of the devices. RFID tags are integrated into the packing of the products but they cannot easily be recognized. Even if they are supposed to be marked one must trust the producers that they really marked their tags correctly.

Taking stock, it is essential to provide proper mechanisms in order to bring the user back into control. How can we achieve this? If we take into account the seven stages of action presented by Norman [18] we realize that it does not suffice to provide means of control, such as a kill function on the RFID tag. It is furthermore necessary to provide adequate observation mechanisms, too. People who do not trust the provider and hence wish to disable their tags will not trust a fully automated and hence invisible and unobservable deactivation process. They must be able to observe the outcome in a reliable and trustworthy manner.

Unfortunately it is not that simple. Besides the system and its single devices the users have context, too. Depending on their current situation they cannot operate special devices or receive special messages. This is most obvious in systems that are deployed in environments that require the users' full attention, for example, in cars or in the working environment. But it is of course also of great importance in the scenario presented above. If the users are carrying some goods they cannot operate their devices, if they are at the check out they will not be interested in goods they might receive at the bargain-basement. This shows that it is important to correctly adapt the user interfaces to the current states of the users and their current preferences, where these preferences cannot be seen as unalterable over time.

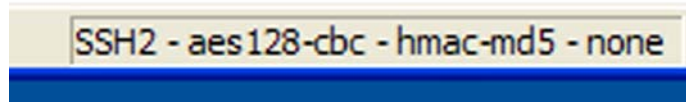


Figure 1: How can the security properties be evaluated?

Future paradigms for human computer interaction hence must take into account several types of context. Classify this context to tell apart the different states of the devices on the one and the states of the users on the other hand. Furthermore, they must proactively adapt the user interfaces to these states in order to present the information required in a very convenient and unobtrusive way. This means that user interfaces for ambient intelligence systems will be decoupled from the actual implementation and the services that make up a system must provide enough information to implement algorithms that can automatically propose the best way to present information and receive input from the users.

## 4 Unobtrusive Observability

In his seminal book “The psychology of everyday objects” Don Norman presented the seven stages of action [18]. These stages form the process of mostly any action people perform. An elemental part of this theory of action is a feedback loop. That means, people can not plan the accomplishment of very complex tasks in one single step. They, in fact, start performing one subtask they consider correct, observe the outcome of that subtask and then decide on the further steps. To this end, it is necessary that they are able to observe the system’s behavior at any time, because the system’s state might be an important parameter when planning the next actions.

When designing complex and highly dynamic systems it is very difficult to provide adequate levels of observability. The system’s state is most often described in technical terms, such as the type of encryption algorithm used or the length of an encryption key in bits. This seems adequate for the designers and programmers of an application, but it is impossible for laypersons (and often even experts) to decide whether, for example, the given algorithm with the given key length is an adequate choice in the current situation (cf. the interface depicted in Fig. 1).

This becomes even worse in the scenarios of Ambient Intelligence where we have *Mixed-Mode Systems*. This are systems that are build of heterogeneous components that vary in processing power, storage and energy supplies. This worsens the situation for implementing observable security, as additional parameters must be taken into account. Consider, for example, devices with slow processors or a small energy supply that cannot perform the strongest encryption at hand. What kind of encryption is the adequate one in this special case? What happens if we reduce the strength of the encryption in favor of speed? It is beyond the scope of this paper to discuss the issues related with the area of conflict between security and resource poor devices in general, but it is important to realize that for unskilled

users this is a problem way beyond their technical understanding.

From the usability point of view there is only one viable solution: system states must be abstracted from their concrete implementation and interpreted in a way the user can comprehend. For the encryption example above the actual estimate of security must be presented not the algorithms. This must be an adequate measure, for example, an estimate of the time an attacker needs to break the encryption. If this is set into relation with the maximum level of security the device can provide and the battery life time, users are able to adjust the device to their needs.

As a metaphor one can think about a speedometer of different system properties. For mobile devices battery lifetime under certain assumptions could be one of these properties, but also the aforementioned time needed to break an algorithm with the given key length or the time required to guess a password entered.

It is important to realize that all these quantifications that make up the “speedometer” are just estimates. Currently these estimates are calculated by experts if they plan and implement an application. For future systems that are by definition open and an ad hoc combination of services, this becomes infeasible. As it is impossible to predict the exact composition of a system at a given point in time it is impossible to specify the system states in advance. It is hence impossible to provide the correct algorithms and security mechanisms in advance. It is necessary to provide a collection of different mechanisms and algorithms for the user to choose from together with the estimates of their performance (regarding security as well as consumption of resources). If this information is accurately presented the users are in a position where they can successfully accomplish their tasks.

This does not necessarily mean that the user must decide about every mechanism and algorithm by itself. It is possible that the applications do proactively choose the adequate ones according to the users’ policies. These policies must contain the objectives of the users regarding the different measures. Only if it is impossible to find an adequate configuration that satisfies the different objective it is necessary to bother the user with the different settings of the system.

In the following chapter we will thus provide an approach for log views and secure logging. The secure recording of events within the system is the most fundamental building block for all approaches that require observability. Any visualization of data heavily depends on the quality of the data it needs to visualize. This is overly true in the context of the dynamic and flexible scenarios presented above.

## **5 Secure Logging and Log Views**

Our goal is not to avoid loss of control from the outset, as this would require effective mechanisms to prevent personal data from being collected and regulate the usage of such data, thereby anticipating that customers have complete knowledge about the systems or that their behavior is predictable. This is in conflict with the properties presented in Chapter 3.

Instead, our goal is to provide the customers with evidence regarding how their data has been collected and processed within the system, as schematically depicted in Figure 2. How

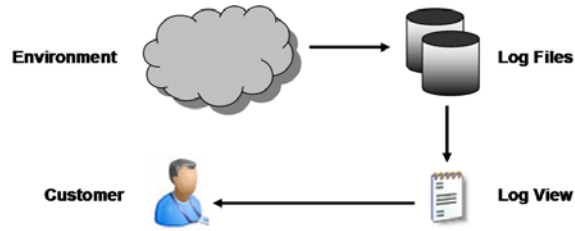


Figure 2: Schematic representation.

that information is presented to the users is discussed in Chapter 6.2. Here, ordinary events happening in the system – e.g. sensor measurements and interactions with RFID-tags – are recorded in (possibly distributed) log files. These log files exist for several reasons. On the one hand, they are indispensable when it comes to managing the systems and pinpointing potential conflicts and failures. On the other hand, they are containers of context data upon which personalization strategies such as individualized services and recommendations could be based upon. Finally, *log views* are the construct we employ to make the behavior of the system evident to customers. Similar to database views, log views encompass all the entries in the log files related to a particular customer. Log views are generated in the system and sent to the customer.

To realize the approach proposed in Figure 2, we must first ensure that collected log data shall not be maliciously modified, i.e. log entries remain authentic throughout the necessary time span. Second, it must be ensured that log views are designed in a way that customers are able to comprehend the information they contains. In this section, we present a secure logging mechanism, as it is a central building block for the generation of log views. For details, see [1, 2].

## 5.1 An Approach to Secure Remote Logging

Log data can only provide a sound basis for further services when it is authentic. We define authenticity as the simultaneous fulfillment of data *integrity* and *uniqueness*, as illustrated in Fig. 3. *Confidentiality* of log entries is necessary for privacy and is considered as an extra protection goal. A log service is labeled *secure* when integrity, uniqueness and confidentiality properties are fulfilled.

- Integrity states that log data faithfully reflects the state of the devices, i.e., the log data is accurate (entries have not been modified), complete (entries have not been deleted), and compact (entries have not been illegally added to the log file). Thus, log data is not modified, deleted, or appended during the transmission to, and storage at, the collector.
- Uniqueness states that log data shall not allow for parallel realities. Concretely, it is impossible to intercept log data sent from  $d_1$  to  $c_1$  and to resend it (possibly in

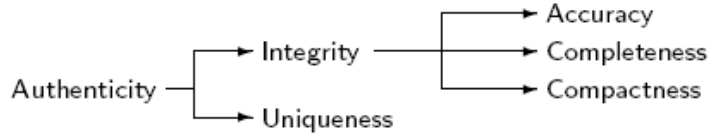


Figure 3: Authenticity property for secure logging.

modified form and claiming a different device identity) to  $\mathcal{O}_2$ . Log data must be uniquely tagged.

- Confidentiality states that log entries cannot be read by unauthorised individuals, for this would harm inner privacy. Note that confidentiality is also related to uniqueness, for log data transmitted in clear-text can be easily duplicated.

These properties are implemented with cryptographic techniques, which need to ensure *tamper evidence*, i.e., attempts to illicitly manipulate log data must be detectable to a verifier [14], and *forward integrity*, i.e., log data contains sufficient information to confirm or rebuke allegations of log data modification before the moment of the compromise [4].

The goal of the attacker is to gain access to private log data and, thus, to violate its integrity, uniqueness, and confidentiality. The threats posed by an attacker are described using an attacker model. While we are aware of recent ongoing research on formally characterizing attacker models for dynamic environments [7], we refrain from sticking to a particular model.

## 5.2 Overview of our Approach

Our approach to secure remote logging services in dynamic systems is based on and extends the techniques proposed in [20]. The idea is to devise a protocol to securely store log data. The protocol starts at the device: it is in charge of applying cryptographic techniques to ensure tamper evidence and forward integrity. When the device runs out of storage, it contacts the collector to request remote storage. The protocol ends with an irrefutable proof of possession of the collector. In detail, our protocol consists of the following main steps:

1. *initialization and construction of the log file*: the device is in charge of applying cryptographic techniques to safeguarding the integrity and uniqueness of its log file. To this end, it computes a secret random value  $pv_0$  and, for each entry, a proof value  $Z$  associated to that entry based on  $pv_0$ . The  $Z$ -values build a chain, so that the integrity of the whole chain can be checked by analysing its last link.
2. *mutual authentication of services*: apart from authentication, the device and the collector also agree on a secret value  $pv_0$  that will be used to ascertain authenticity of log messages.

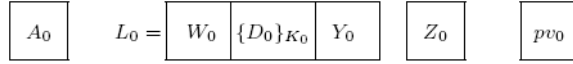


Figure 4: Initial log entry  $L_0$  and values  $A_0$ ,  $Z_0$ , and  $pv_0$ .

3. *acknowledgement of receipt from collector*: by receiving the chunk of log data, the collector computes the  $Z$ -value associated to each entry and sends the last  $Z$ -value signed together with a timestamp and protocol sequence number back to the device. The device then stores this unambiguous piece of information, as it demonstrates that the collector received the chunk of log data correctly and can be thus held accountable for attacks upon this data.

We describe the initialization, appending, and acknowledgement processes below; the authentication phase and the secure exchange of  $pv_0$  are taken for granted.

### 5.2.1 Initializing the Log File

Assuming that the device  $d$  successfully generates a secret value  $pv_0$ ,  $d$  creates the log file by inserting the first entry into it. We illustrate the initialization entry  $L_0$  in Fig. 4, where the fields stand for the following information (we remark that all entries have the same format):

- $W_0$  is a permission mask to regulate the access to the log entry  $L_0$ . According to [20], at the initialisation phase this entry type may be set to **LogfileInitializationType**.
- $\{D_0\}_{K_0}$  is the symmetrically encrypted log data for the entry  $L_0$  and  $K_0$  is a random session key. To provide the necessary security guarantees,  $D$  contains not only the event to be logged, but also a timestamp and a protocol identifier. (The former states the actuality of the entry, the latter avoids harmful protocol interactions between different protocol steps.)
- $Y_0$  is the first link of a hash-chain.<sup>1</sup>
- $A_0$  is the authentication key for the first entry. As we discuss in [1], this is the secret upon which the security of the whole protocol rests. In practice, since we do not assume that the device is tamper resistant, we suggest that this secret should be kept off-line.
- $L_0 = W_0, \{D_0\}_{K_0}, Y_0$  stands for the initial log entry.
- $Z_0$  is the message authentication code of  $L_0$  defined as  $MAC_{pv_0}(L_0)$ . This piece of information is used to compute the proof value associated to the whole chunk of log entries and, thence, can be used as a challenge against the collector. For this, it will *not* be send along with  $L_0$  to the collector.

---

<sup>1</sup>In the simplest form, a hash-chain  $Y$  is inductively defined as  $Y_1 = Hash(Y_0)$  and  $Y_n = Hash(Y_{n-1})$ .

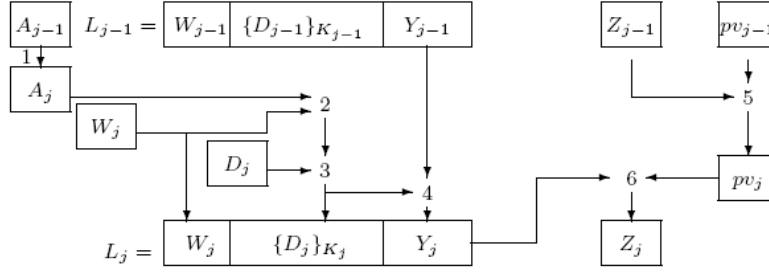


Figure 5: Adding an entry to the log file.

### 5.2.2 Appending Log Entries

After creating the log file, the device starts adding entries to the log file. This is schematically illustrated in Fig. 5, where the numbers correspond to the items below.

1.  $A_j = Hash(A_{j-1})$  denotes the authentication key of the  $j$ th log entry. The confidentiality of this information is for accomplishing the aforementioned security properties, as it is used to encrypt log entries. Thus, we assume that the computation of the new value irretrievably overwrites the previous value.
2.  $K_j = Hash(W_j, A_j)$  is the cryptographic key with which the  $j$ th log entry is encrypted. This key is based on the permission mask  $W_j$ , so that only authorized services gain access to the entry.
3.  $\{D_j\}_{K_j}$  is the encrypted log entry.
4.  $Y_j = Hash(Y_{j-1}, \{D_j\}_{K_j}, W_j)$  is the  $j$ th value of the hash-chain. Each link of the hash-chain is based on the corresponding encrypted value of the log data. This ensures that the chain can be verified without the knowledge of the actual log entry.
5.  $pv_j = Hash(Z_{j-1}, pv_{j-1})$  is the proof value associated with the  $j$ th entry.
6.  $Z_j = MAC_{pv_j}(Hash(L_j))$  is the authenticator of the  $j$ th log entry. Note that we compute the message authentication code for the whole entry instead of a field of it (in the case of [20], the hash-chain value  $Y$ ).

The log entry  $L_j$  generated by the device consists of the permission mask  $W_j$ , the encrypted log entry  $\{D_j\}_{K_j}$ , and the hash-chain value  $Y_j$ ; it is denoted by  $L_j = W_j, \{D_j\}_{K_j}, Y_j$ .

### 5.2.3 Acknowledgement Phase

Assuming that the authentication between the device and the collector succeeds and they agree on the proof value  $pv_0$ , the last phase of the protocol aims to provide irrefutable evidence regarding collector's possession of the chunk of log data sent by the device, as well as the chunk's integrity. To this end, the following steps are carried out:

- by receiving the chunk starting at  $L_j$  and ending at  $L_k$  (with  $j < k$ ), the collector computes for each entry  $L_i$  the corresponding  $pv_i$  and, thus,  $Z_i$  values.
- after  $k-j$  iterations, the collector obtains the authenticator  $Z_k = MAC_{pv_k}(Hash(L_k))$ .
- the signed proof value is sent to the device  $Sign(Z_k)_{K_c^{-1}}$ . This message includes a timestamp and protocol step identifier.
- the device then checks whether the authenticator matches with the authenticator computed during the second phase. If it does, the device frees the storage by deleting the chunk.

This completes the presentation of the secure logging mechanism upon which the generation of log vies is based. We refer to [1] for a detailed analysis of the secure properties of this mechanism, in particular the fulfillment of the authenticity properties presented in Chapter 5.1.

### 5.3 Generating Log Views

Log views are generated by querying possibly distributed log files for a particular individual. Suppose individual  $GM$  wants to be aware of all the events he or she has been engaged to in the system. To this end,  $GM$  requests the corresponding log view, i.e. a logical table consisting of the entries of the log file.

To generate the log this end, we employ the access control field  $W$  of each entry as an index. The individual –  $GM$  according to our running example – authenticates to the system and queries the log files for entries whose index, i.e.  $W = GM$ , matches to the authenticated individual.

The generation of view proceeds without further ado in case the events triggered can be related to an individual. Figure 6 depicts a view for individual  $GM$ .

At the moment, we consider only the case where individuals are identified at the moment of the event collection. However, should this relationship not be known in advance, techniques such as data mining or sensor fusion should be applied to make the relationship evident, at least up to some statistical probability. Our current research focuses on this issue.

## 6 Usable Security and Privacy for the future super market

How can we integrate these mechanisms to provide usable security and privacy for end users? Based on the classification of services according to the amount of private data they use we propose the integration of this information in contextualized prototypes for the end user.

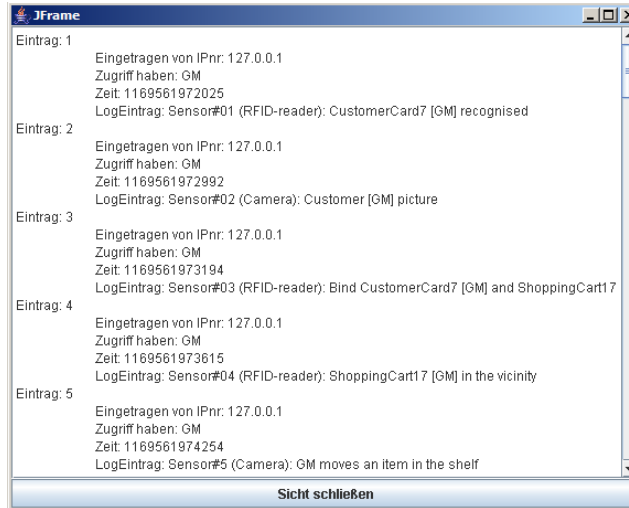


Figure 6: Example of a log view.

## 6.1 Individualized and personalized services

In [19] we presented a classification of services according to the need of input data necessary to provide the service. This classification is well suited for the economical perspective the provider takes. But it is very misleading if we take the users' perspective. It is important to realize that this static view we presented does not hold for the users moving around in the area controlled by the provider. The longer they interact with the system the more information aggregates with them. This information is not vanishing over time, that means that the users in this categorization will inevitably drift from universal ones to personalized ones. If they interact with the system long enough it will be possible for the provider to personally identify each of them.

For the users shopping in this super market it is important to know at which level they actually interact with the system. Before the first interaction they are certainly unidentified. But when will they be uniquely identified? We propose the implementation of a "anonymeter" or "anony-lights" in order to present the user this information. As a measurement we evaluate the securely logged data about the interaction with the system and try to find the user's profile. As long as there is no interaction with the system we can not aggregate more information than all the user profiles have. Thus the probability to get the user profile is as high as randomly picking one of the profiles:

$$P_{ident} = \frac{1}{\text{number of profiles}}$$

While the users interact with the system we get some of their attributes and can sort out some of the users that do not match with these properties. As a consequence the probability of uniquely identifying the user increases (i.e. the k-anonymity set shrinks [21]):

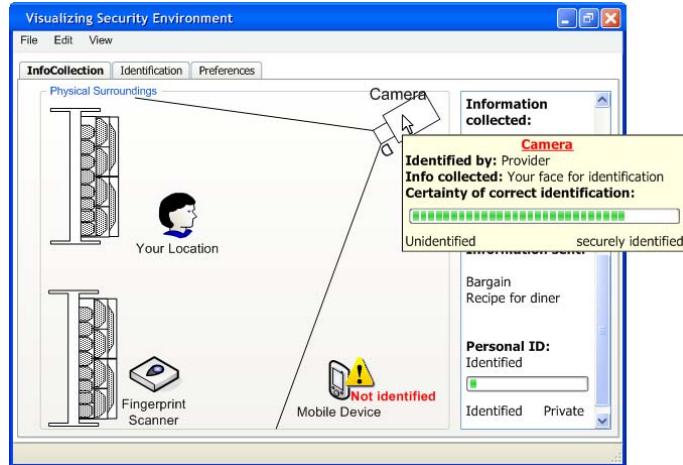


Figure 7: Mockup interface of the Context Descriptive Security Model.

$$P_{ident} = \frac{1}{\text{number of profiles} - \text{profiles sorted out}}$$

It is trivial to realize that the user is uniquely identified with 100% accuracy if all profiles except the one of the user can be sorted out.

## 6.2 Bringing the User Back Into Control

In [13] we propose the integration of security and privacy information into “Context Descriptive Security Models” of the users’ surroundings (see Fig. 7 for an example). These security models present the users’ surroundings in a map like interface. In a supermarket, for example, the shelves are shown, together with relevant sensors. This will help the user to realize where the different parts of the information system are placed and how their interplay is to be evaluated.

It is obvious that it is undesirable to walk around the ambient environments and continually look at the security model presented on the display of some mobile device. We rather see this as a mechanism to gain the users’ trust. They are in a position to visualize the system, there is no magic or hidden functionality. Once the users believe this, they do no longer need to ensure that this is really true. This effect is similar to the trust in open source software: the *option* to look through the code is sufficient to gain the users’ trust. They hope that someone will look through the code and verify its trustworthiness.

What are the sources of information for this visualization interface? We see several possible sources of information for this type of visualization. The selection of the most adequate ones is still an open research question.

According to the principles of social navigation [8] we integrate information from different sources into the map like representation of the users’ environment. Social navigation

in this context means, that decisions of other users are evaluated according to their outcome (with respect to the users' policy). This allows for the "visualization" of different paths and their effect on the map. In the real world this is like a worn path on a field of grass that leads us to the points of interest. Additionally, information gained from social navigation mechanisms can be used to calculate security measures like the anonymizer presented above. These are presented to the users in order to give them a view on the system's current state.

The foundations of secure logging mechanisms are the most basic building block. It will provide two possibilities to bring the user back into control again: On the one hand, the integration and evaluation of the interactions of other users with the system provide a great source of information. This evaluation is only possible if there exists enough reliable and trustworthy information about these interactions. On the other hand, the users have the possibility to reclaim their rights after they have been violated, if they are in a position to realize violations. To realize the infringement of users' rights and to reliably prove this infringement information written to the log file is essential.

Compared to the current state of the art in usability research the approach presented in this paper provides several advantages. There exist interfaces that are signaling the current state of the privacy mechanisms [6]. While these work well for rather well defined interaction models, like browsing the web, the integration of these interfaces into the context is essential. Considering the different approaches for engineering usable and secure systems [9, 10, 11] it is important to realize that these well defined processes of developing, testing and deploying systems are no longer feasible for the steadily growing and evolving systems of ambient intelligence. The special properties of usable security as they are for example presented in [24], make it inevitable that dedicated user interfaces for security and privacy are carefully investigated and designed.

## **7 Conclusion**

In this paper we have presented the scenario of the supermarket of the future. A plethora of sensors and computer systems enhance the shopping experience of the customers. We argue that it is inevitable to cope with the privacy, security and usability issues of these systems, because the pervasive nature of these systems does not allow for avoidance of these technologies. To this end, we present mechanisms to reliably and securely log the systems events and hence prove the systems' behavior. This brings the users into a position where they are able to realize possible infringements and they can actively control their privacy and security requirements. As an interface between the users and the system we propose "Context Descriptive Security Models". These allow for the intuitive presentation of measures of relevant security properties, such as anonymity or strength of confidentiality. They, furthermore, allow for the integration of information collected from other users and their evaluation of the outcome. This social navigation will help the users to find a configuration of their systems that really fits their needs.

## References

- [1] Accorsi, R.: *On the relationship of privacy and secure remote logging in dynamic systems*. In: Fischer-Hübner, S.; Rannenberg, K.; Yngström, L.; Lindskog, S.; editors, Proceedings of the 21st IFIP TC-11 International Security Conference: Security and Privacy in Dynamic Environments, International Federation for Information Processing. Springer-Verlag, 2006, 201, 329–339.
- [2] Accorsi, R.; Hohl, A.: *Delegating secure logging in pervasive computing systems*. In: Clark, J.; Paige, R.; Pollack, F.; Brooke, P.; editors, Proceedings of the 3rd International Conference on Security in Pervasive Computing, Lecture Notes in Computer Science. Springer Verlag, 2006, 3934, 58–72.
- [3] Alahuhta, P.; De Hert, P.; Delaitre, S.; Friedewald, M.; Gutwirth, S.; Lindner, R.; Maghiros, I.; Moscibroda, A.; Punie, Y.; Schreurs, W.; Verlinden, M.; Vildjiounaite, E.; Wright, D.: *Dark scenarios on ambient intelligence: Highlighting risks and vulnerabilities*. Technical report, SWAMI Deliverable D2 A report of the SWAMI consortium to the European Commission under contract 006507, November 2005.
- [4] Bellare, M.; Yee, B.: *Forward integrity for secure audit logs*. Technical report, University of California at San Diego, Dept. of Computer Science & Engineering, 1997.
- [5] Blache, F.; Chraïet, N.; Daroux, O.; Evennou, F.; Flury, T.; Privat, G.; Viboud, J.-P.: *Position-Based Interaction for Indoor Ambient Intelligence Environments*, Lecture Notes in Computer Science. Springer, 2003, 2875, 192–207.
- [6] Cranor, L.-F.; Guduru, P.; Arjula, M.: *User interfaces for privacy agents*. ACM Transactions on Computer-Human Interaction, 2006, 13(2), 135–178.
- [7] Creese, S.; Goldsmith, M.; Harrison, R.; Roscoe, B.; Whittaker, P.; Zakiuddin, I.: *Exploiting empirical engagement in authentication protocol design*. In: Hutter, D.; Ullmann, M., editors, Proceedings of the 2nd International Conference Security in Pervasive Computing, Lecture Notes in Computer Science. Springer-Verlag, 2005, 3450, 119–133.
- [8] DiGioia, P.; Dourish, P.: *Social navigation as a model for usable security*. In: SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security, 2005, 101–108.
- [9] Dix, A. J.; Finlay, J. E.; Abowd, G. D.; Beale, R.: *Human Computer Interaction*. Prentice Hall, 2nd edition, 1998.
- [10] Dourish, P.; Redmiles, D.: *An approach to usable security based on event monitoring and visualization*. In: Proceedings of the 2002 Workshop on New Security Paradigms. ACM Press, 2002, 75–81.

- [11] Flechais, I.; Sasse, M. A.: *Developing secure and usable software*. In: Workshop on Human-Computer Interaction and Security Systems. ACM, April 2003.
- [12] Friedewald, M.; Costa, O.D.; Punie, Y.; Alahuhta, P.; Heinonen, S.: *Perspectives of ambient intelligence in the home environment*. Telematics and Informatics, August 2005, 22(3).
- [13] Höhn, S.: *Bringing the user back into control*. In: S. Fischer-Hübner, S. Furnell, and C. Lambrinouidakis, editors, Trust, Privacy and Security in Digital Business, Lecture Notes in Computer Science, Springer, 2006, 4083, 114–122.
- [14] Itkis, G.: *Cryptographic tamper evidence*. In: Proceedings of the Conference on Computer and Communication Security, ACM Press, 2003, 355–364.
- [15] Kaiser, J.; Reichenbach, M.: *Evaluating security tools towards usable security*. In: Proceedings of the 17th IFIP World Computer Congress (WCC2002), August 2002.
- [16] Lüke, F.: *Jobgefährdend*. Die Zeit, 2006.
- [17] Nehmer, J.; Becker, M.; Karshmer, A.; Lamm, R.: *Living assistance systems: an ambient intelligence approach*. In: ICSE '06: Proceeding of the 28th international conference on Software engineering. ACM Press, 2006, 43–50.
- [18] Norman, D. A.: *The psychology of everyday things*. Harper Collins Publishers, 1988.
- [19] Sackmann, S.; Strüker, J.; Accorsi, R.: *Personalization in privacy-aware highly dynamic systems*. Communications of the ACM, September 2006, 49(9), 32–38.
- [20] Schneier, B.; Kelsey, J.: *Security audit logs to support computer forensics*. ACM Transactions on Information and System Security, May 1999, 2(2), 159–176.
- [21] Sweeney, L.: *k-anonymity: A model for protecting privacy*. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, May 2002, 10(5).
- [22] Valcourt, E.; Robert, J.-M.; Beaulieu, F.: *Investigating mobile payment: supporting technologies, methods, and use*. In: Proceedings of the IEEE International Conference on Wireless And Mobile Computing, Networking And Communications, 2005, 29–36.
- [23] Weiser, M.: *The world is not a desktop*. Interactions, 1994, 1(1), 7–8.
- [24] Whitten, A.; Tygar, J. D.: *Why Johnny can't encrypt: A usability evaluation of PGP 5.0*. In: 8th USENIX Security Symposium, 1999.
- [25] Wolfram, G.: *The future of retail: A visionary outlook*. Technical report, METRO Group Future Store Initiative, November 2005.