Privacy in an Ambient World (PAW) ⋆

Using licenses and private computing as PET

Kathy Cartrysse[1], Ricardo Corin[2], Marnix Dekker[3], Sandro Etalle[2], Jaap-Henk Hoepman[4], Gabriele Lenzini[2], Jan v.d. Lubbe[1], Jan Verschuren[5], and Thijs Veugen[3]

[1] EWI, Delft University of Technology
[2] EWI, University of Twente
[3] TNO Telecom, Delft
[4] Dept. of Computer Science, University of Nijmegen
[5] TNO ITSEF BV, Delft

Abstract In the vision of an ambient intelligent world, innumerable small interconnected devices will surround us and support us in our daily tasks and while at leisure. To do so, these devices need to know and exchange our personal preferences. Moreover, without any built-in countermeasures these devices are more than able to collect much more private information. This paper presents the goals and aims of the Privacy in an Ambient World (PAW) project. The purpose of PAW is to build a privacy protecting architecture for the future ambient world. This architecture is based on two foundations. To control, and to empower the user to control, the dissemination of personal preferences a licensing system will be developed. This licensing system is similar, but not equal, to licensing schemes used in digital rights management. To limit unwanted and surreptitious private information collection, private computing schemes for ambient systems will be developed. Both approaches are detailed in this paper.

1 Introduction

In the near-future ambient world, interconnected intelligent devices will surround us, at home or while travelling. These devices and their local networks will also be connected to the outside world through broadband and/or wireless networks. Numerous services to support us in our personal life will be provided through these ambient devices, and over the connection to the outside world.

To adapt to our personal life style, and to offer the right service at the right time in the right place, such services will rely on the use of private data. In particular user profiles will have to be kept and exchanged. The ambient devices may either learn about us and our personal preferences either through explicit

⋆ Id: pet-position.tex,v 1.17 2004/01/25 14:41:25 jhh Exp .
Research partially funded through the Dutch IOP GenCom programme.
Contact author: Jaap-Henk Hoepman (`jhh@cs.kun.nl`), University of Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, the Netherlands.

customisation by their owners, or by unobtrusively observing our behaviour. In both cases, these devices will learn much about our private lives.

This scenario poses a direct threat to our privacy in a variety of ways. The current approach towards protecting our privacy is by hiding our identity (e.g., by using pseudo-identities). However, hiding our identity does not mean our privacy is completely protected. Our actions, even when they are anonymous, can thwart good privacy protection, because of the possibility that various pieces of information can be linked together. Carefully considering the concept of privacy, we see that privacy protection can be divided into four different aspects.

1. Protecting a person's identity (e.g., our name).
2. Protecting an identity's personal data (e.g., our preferences).
3. Protecting the actions undertaken by an identity (e.g., our activities).
4. Protecting the instructions or tasks scheduled by an identity (e.g., software and mobile code executed on our behalf).

To provide full privacy, adequate solutions must be found for each of these categories.

The first category, protecting our identity is mainly about protection during identification actions, and has been the focus of most of the research on privacy (see Section 2). Not in all cases it is necessary to perform identification using our name. Sometimes a role or a pseudonym will be sufficient, for example.

Controlling our own personal data involves more than simply hiding our identity. Even when using pseudonyms, we need to provide many personal data items to other persons or entities in a system, e.g., to let the system behave according to our preferences. Once we reveal that information, we have no control over it anymore and other entities are able to obtain personal information that can affect our privacy. Solutions to this problem provide privacy in the second category.

The third and fourth categories are concerned with our privacy in our home and in extensions of our home environment in the outside world. Both are best illustrated by an example. Suppose you are staying in a hotel, and you want to perform some activities using some of the devices in the hotel room and using some data stored back home. To do this, your home environment must be extended to this hotel room maintaining privacy to prevent the hotel from collecting information about your activities. This is an example of privacy in the third category.

Due to advances in software technology, reasoning about privacy needs to be taken one step further and must also consider the protection of instructions or tasks scheduled by an identity (i.e., the fourth category). An example of this is mobile code, aka agents. The idea behind agents is that they can travel over the network and be executed at various nodes. This in stark contrast with the traditional approach where only static data is transmitted over the network. Mobile software leads to many privacy and security risks. Consider the case in which a program for reserving a hotel room is sent over the network. The program may be executed at computers owned by hotels and it may be in the interest of a hotel to obtain some insight into the contents of the program so that it can adjust its offer according to the user's criteria. This is of course a privacy risk, moreover the hotel may eventually offer a higher price than it would have if it had not known the user's criteria to make a booking.

The objective of the PAW (Privacy in an Ambient World)[1] project, that recently started on a grant obtained form the Dutch ministry of Economic Affairs, is to develop a privacy protecting architecture that can provide full privacy of the user in an ambient world. As said before, full privacy can only be achieved by providing adequate solutions for each of the four categories described for the home-environment. Therefore the main objective of the project is to find solutions in those categories that have not been addressed yet in research (i.e., categories 2, 3 and 4).

In this paper we elaborate on the approach to be taken by the PAW project to construct such an architecture. First, in Section 2 we summarise the state-of-the-art of privacy protection, relevant to the PAW project. Our approach to protecting privacy is described and motivated in Section 3. It is based on two principles.

- To control, and to empower the user to control, the dissemination of personal data a licensing system will be developed. This licensing system is similar, but not equal, to licensing schemes used in digital rights management.
- To limit unwanted and surreptitious private information collection, private computing schemes for ambient systems will be developed.

Technical details of our work are presented in Section 4, and Section 5 concludes this paper.

2 State of the art

The privacy goals discussed in the Introduction have been partially addressed in the literature, mainly in the following fields:

1. confidential communications,
2. database management,
3. mobile systems, and
4. ubiquitous computing and ambient systems.

We will discuss the state-of-the-art research in each these fields separately, in so far as it concerns the goals and approach of the PAW project.

Privacy in confidential communications. In this field privacy is prevalently interpreted as a capability of not revealing, while communicating, confidential information. Information leakage must not happen either through a direct disclosing or through indirect tracing (e.g., leaving traces that may ultimately be traced back to the information). In this sense, confidential information includes

[1] Additional information and future developments on the PAW project can be obtained from our project web site `http://www.cs.kun.nl/paw` .

both user's identity and personal data. The former has been studied within anonymity [Cha85, Cha92, SS96, Aba99, Shm02] and pseudo-identities preserving techniques [LRSW99]. The latter is more related with secrecy, widely studied in [Mea96, Sch98, AG97, THG98, Aba99, AD01]. In some approaches, secrecy and anonymity are commonly defined as information hiding properties, as [HO02, HO03] show. On the other hand [HO03] show how secrecy and some standard notion of anonymity subtly differ from each other.

Privacy in database management In this category, privacy is intended mainly as a guarantee that unauthorised disclosure of sensitive data will not take place e.g., by inference attacks or statistical queries. Therefore the main privacy goal is to protect users' personal data.

In [And01] Anderson discusses the problems of multilateral security and inference control [Den82]. Briefly the first problem concerns the definition of fitting access controls, w.r.t. a set of data, in order to prevent sensitive information from flowing towards unauthorised entities. The second problem, known since 1960's, studies statistical security, that is how to protect sensitive data from being disclosed by statistical queries involving it. A brief introduction about recent results, trends and references about inference control in statistical databases can be found in [DF02]. Inference control problems arise also in multilevel databases [JM95], where data are classified into confidentiality privacy levels e.g., high (private) and low (non-private). No inference problem exists if a user cannot infer any high information from a sequence of queries that each involves low level data only.

In [HILM02] the problem of guaranteeing privacy in a model of application service providers (ASP) is studied. In ASP "applications as services" are supplied to Internet costumers by some provider. Precisely [HILM02] focuses on "databases as a service", where a databases provider offers its services in storing data and running queries in the behalf of a customer. For the customer might not trust completely in its provider, he does not want the provider could perform its own queries on the database. "Database as service" problem is also considered in [HIM02].

In [GIKM98] a solution to the algorithmic problem of *Symmetric Private Information Retrieval* (in short SPIR) is studied. The SPIR problem involves both the goal of users' identity and data privacy. In fact it aims to satisfy the two following conditions: (a) to avoid that the data base manager could know the identity of the user, (b) to avoid that the user could retrieve more information than just the information required. Hence in this problem privacy is also used as protecting the users' identity. Similar to SPIR is the problem of "negotiating privacy", studied in [JLS02]. Negotiated privacy arises in any context where the goal of data collection is to detect and reveal "exceptional" conditions, while keeping routine events completely hidden. The referred scenario involves a set of data owners who consign their data to a database manager that, in turn, is allowed only to query previously-negotiated properties while prevented to access any other users' personal data. Negotiated privacy differs from SPIR schemes [GIKM98]. In the latter the manager cannot retrieve data from a database without the collaboration of an user, while in the former no query may take place at all until the negotiated conditions become true.

Privacy in mobile systems In this category privacy refers to a scenario where mobile agents may run on different hosting machines. Here different privacy goals arise. Firstly, privacy is preserved when no malicious host may learn secrets while executing mobile agents, as studied in [ST98], so it refers mainly to task and activity protection, but also data protection is involved. Secondly privacy may protect an host from malicious actions by the agents, and it refers to data or resource protection. In [Edw96] Edwards suggest the use of *polices*, in order to allows users (or better, their applications) to control collaboration in a potentially chaotic environment. Policies provides hosts with the ability to regulate access to their information and personal space and to govern how their applications will respond to events.

In [SBM03] a subset of Ambient Calculus [CG00b] is used to model a system composed by entities which are organised in a hierarchical structure of inclusion. For example, entities may represent either physical locations (e.g., an office, a workstation, a laptop) or a logical locations (e.g., a context, an agent). Inclusion represents either a physical (e.g., a workstation is in an office) or logical (e.g., an agent runs in a context) inclusion among entities. A process — in this fragment of the Ambient Calculus — represents an instantaneous picture of a system where entities may find themselves located into other ones by following a tree-based, spatial or logical inclusion relation. The security of the whole system is ruled by *security policies* supervising the evolution of the system, formalised as formulas of a decidable subset of Ambient Logic [CG00a].

Many articles have been written about code confidentiality, and most of them define confidentiality for protecting code such that it is impossible to determine the code's content. Hohl [Hoh98] described a mechanism called "time limited black box security", where the idea is to obfuscate the source code such that it takes more time to understand the code than the programmed time limit. A more cryptographic method is presented in [ST98] where Sander and Tschudin encrypt functions that can be executed in its encrypted form. This method works for polynomials and rational functions. Sander et al. [SYM99] extended the results to all functions computable by circuits of logarithmic depth and further generalised to arbitrary functions, provided they can be represented by a polynomial-size circuit. As far back as 1990, Abadi and Feigenbaum [AF90] described a method that provides confidentiality for circuit evaluation. A disadvantage of this method is that many interactions are required to provide confidentiality. Many other solutions have been published to provide secure circuit evaluation, but none of them is very practical and efficient. Loureiro et all. described how functions can be hidden using coding theory [LM99]. Several more practical methods have been proposed, but they are all based on either trusted hardware located at the host [Yee99] or on the presence of a trusted third party or oblivious third party [ACJG01].

Privacy in ubiquitous and ambient computing In this last category, defining and enforcing privacy is still very much an open question. Ubiquitous computing, also known as ambient systems, is a general term to describe a world where invisible and comprehensive networks control and interact with public and private life of users. Clearly, in such a world all the goals of privacy are involved. Few works exist on how to define privacy policies, how to preserve them and how to detect privacy violations in an ambient world.

In [Edw96] Edwards introduces the problem using *polices* in order to allows users' applications to control their activities when those ones run in a collaborative and potentially chaotic environment. Policies provide users with the ability to regulate access to their information and personal space and to govern how their applications will respond to events. The authors show how control access mechanisms suffice for modelling most common policies in the area of awareness and coordination (e.g., anonymity, pseudonymity) both in their static and dynamic (i.e., they may change as application run) version.

Another recent approach for describing privacy policies is the Platform for Privacy Preferences (P3P)[2] [Cra02]. P3P is a standard of the World Wide Web Consortium (W3C) that specifies how to express privacy policies in Web sites and browsers. P3P is intended to standardise the process in which Web servers and Web users can agree on how a user's personal information should be handled. Also P3P-enabled Web sites make such privacy policies be available in a standard, machine-readable format. Some experiences on the usage and testing of P3P are discussed by Hogben *et al.* [HJW02], and by Cranor *et al.* [CBK03, CAG02].

In [LHJ$^+$03, LDM02] Leder *et al.* discuss that in an ambient world, secrecy and anonymity can only cover two extreme cases of privacy. In fact, in the many situations of everyday life people wants or needs to share his information with others. The general question is how to share personal information with the right people and at the right level of detail. The authors call it *everyday privacy*. Central to their model of privacy is the ability adjust the precision of dynamic contextual information disclosed by a user to other people. For example a large spatial precision may disclose your position only w.r.t. the town you are but not the street, building, room we you indeed are. Control over precision provides control over the information density of a given disclosure. Further information may be found in [Lan03, Lan02, LMD03, LMDB03].

3 The PAW approach

PAW is a project that looks at privacy in a very broad sense. All privacy aspects are taken into account, not only confidentiality of data. The objective of PAW is to develop a privacy protecting architecture that can provide full privacy of the user in an ambient world. As said before, full privacy can only be achieved by providing adequate solutions for each of the four categories described in the introduction.

[2] See also P3P Project website `http://www.w3.org/P3P/`.

PAW's approach to building a privacy enhancing architecture addressing these categories is based on two principles.

detection To control, and to empower the user to control, the dissemination of personal data a licensing system will be developed. This licensing system is similar, but not equal, to licensing schemes used in digital rights management.

prevention To limit unwanted and surreptitious private information collection, private computing schemes for ambient systems will be developed.

Moreover, these detection and prevention techniques are combined into a single architecture that is verified and validated separately.

The use of a licensing scheme for privacy protection has been suggested in the past[3] but has never been an object of serious academic study. The idea originates from the observation that, at least from a theoretical perspective, the objectives of a digital rights management (DRM) system and a privacy enhancing technology are remarkably similar. Namely, both strive to control the use and dissemination of data after the original owner released that data. Licensing schemes have been proposed and investigated many times as the core of digital rights management schemes (see Sect. 4.1). It seems only natural then to study their application as a privacy enhancing technology as well.

Licensing based privacy protection is also a natural successor to schemes like P3P [Cra02]. P3P's main weakness is that privacy preferences of the user are not enforced in any way, and only depend on the trustworthiness of the owner of the website. The use of licenses extends such an approach with the ability to automatically check for privacy violations: any data item for which no corresponding license can be presented constitutes a possible privacy violation.

In fact, in a context like P3P, and similarly in an ambient context, maintainers of websites and producers of ambient devices have an interest in handling privacy correctly. Users have a broad choice of websites to visit. Like consumer electronics today, users will have a broad choice of ambient devices to choose from as well. Privacy protection is one of the selling points for such devices. If the privacy protection is based on a licensing scheme, producers have more than mere good faith to convince their consumers: by subjecting themselves to license reviews, they can boost their trustworthiness in a reliable manner, and gain a competitive advantage.

Private computing has been studied before, for instance in the precursor to this project, the PISA project [PIS03], in which some of the current members participated. Here we focus on two aspects. First of all, we continue to develop a theoretical model, such that the possibilities and impossibilities of privacy protection in an ambient world become fully known. This will help us in providing practical solutions to protect privacy. Secondly, existing cryptographic algorithms (e.g., for signing messages) are converted in such a way that they do not leak private information (e.g., the private signing key) in adverse environments.

[3] Dan Geer and others, June 25, 2002, on the cryptography mailing list (`cryptography@metzdowd.com`).

4 Project components

The PAW project aims to combine a licensing approach with private computing techniques in a single privacy protecting architecture. The licensing approach actually consists of two parts: defining a licensing language together with its semantics, and developing licensing protocols implementing and enforcing those semantics. Moreover, the PAW project aims to more formally verify and validate the architecture developed. These four parts of the project (licensing semantics, licensing algorithms, private computing and verification & validation) are discussed in the following sections.

4.1 Licensing semantics

In order to protect and control the dissemination of explicit personal data like user profiles, we are going to use a licensing system. The underlying idea is that the use and the storage of explicit personal data are illegitimate unless authorised by a specific license. The main functions of licenses are the following.

- To describe the authorised actions on the data.
- To specify terms and conditions for using the data.
- To describe the information of the data, e.g. to identify the data creator, distributor and user or consumer.
- To ensure the integrity of the data.

To specify licenses we need an appropriate language, which must be comprehensive, generic and precise. In the literature there exist no suitable proposals. In fact, no one has ever investigated the use of licensing languages for privacy protection.

The licensing languages that can be found in the literature are the Digital Right Languages (DRL), that are developed to describe the conditions of use of digital assets such as music and video. In fact, the last few years have witnessed a proliferation of DRLs — usually based on XML — like XrML[4] and ODRL[5]. These languages however are not suitable for protecting personal data. Indeed, they leave a number of issues unresolved, which we will address in our research, and which we briefly summarise here.

- DRLs are static. One of the challenges the language we aim at has to address is that it has to be dynamic: personal data can be filtered, projected, joined, etcetera. In one word, personal data can be transformed. Today's DRLs are static, and cannot cope with such a dynamic scenario.
- Lack of a formal semantics. DRLs rely on the intuition behind the syntactic expressions. The interpretation of a license can be vague or even inconsistent. The languages proposed are sophisticated and complex in syntax, but poor in genuine semantics.

[4] `www.xrml.org`
[5] `www.odrl.net`

– Lack of support for formally describing the environment of the license. In modern business models, the environment is a crucial factor in deciding whether a certain license can be employed or not.
– DRLs are not supported by design methodologies that allow designers to study the consequences of their design during deployment. The design tools associated with a design methodology would encourage designers to as what-if questions, to which the tools would provide the answers. Currently it is the harsh reality that reports that the system has been hacked or worse yet: the hack may go unnoticed for some time.

Within the PAW project we are going to define a language for describing licenses. Each of the above points represents a research question for the PAW project. In particular the first two points raise a number of issues at all levels of interest.

– At the language level, there is the need to project, split and recombine data. Likewise, we must be able to transform the licenses so that they can follow the evolution of the data. To do this, we need a formal language - the PAW licensing language - that allows for algebraic manipulation.
– The semantics of the language has to be formal, and computable: we aim at an architecture in which we can automatically check the validity of an action involving the use of personal data.
– At the same time, the semantics of the language has to be able to describe the usual conditions of use of personal data, which are often difficult to capture in a formal framework.
– At the methodological level we see a paradigm shift from static to dynamic licensing. Designers must now ensure that such a dynamic process would always adhere to the general terms and conditions agreed with the data owner.

To address these research problems we propose to design a generic framework in which licenses can be modelled, and where new licenses can be created or calculated from existing licenses according to precisely defined rules. The licensing language will be integrated in the privacy protecting architecture.

4.2 Licensing algorithms

Complementing the licence semantics, we need secure methods and algorithms to perform the following basic operations and tasks.

– Create/issue a license
– Store a license
– Link a licence to a particular data item
– Protect the integrity of the licence
– Transmit a license
– Verify the validity of a licence
– Check whether a certain action on a data item is allowed, given the license
– Revoke a licence

These operations correspond to the most basic, static, use of licenses.

In our system, licenses are dynamic. Combining data items requires the creation of a combined license (based on the original licenses), guided by the semantics of the join. Processing parts of data-items may require the splitting of the corresponding licenses, one for each part. This is especially true when splitting is performed to transmit part of a data item to another principal. Regarding this transfer of licenses, this requires the holder of the license to create a new license on behalf of the original data owner for the new principal. This should of course only be possible if the original license allowed the holder of the license to create such sublicenses. Moreover, if the holder of the license is allowed to create such sublicense, he should be able to do so without the cooperation of the original data owner (both to reduce the load on the data owner, but also to prevent the data owner from restricting the terms of the license by not cooperating).

In short, this means that at least the following more complex operations need to be securely implemented as well.

- Combine licenses (and their data items)
- Split licenses (and their data items)
- Project licenses (and their data items)

For the sublicensing algorithms we will use the delegation protocols (see [HSK99]) as a starting point, as well as the ticketing concepts developed for the Kerberos system (see [SNS88]) and protocols used in the digital rights management world. In any case, these algorithms should be efficiently computable and not produce overly wieldy licenses. Moreover, we aim to develop off-line algorithms that do not require participation of the original creator of the licenses to perform the necessary operations on the license. Naturally, when a certain operation is not performed by the license, the operation should be impossible to be performed on it.

The licensing algorithms will be embedded into the privacy protecting architecture making use of the private computing platform developed as well (see Section 4.3).

4.3 Private computing

The categories of protecting actions or protecting set tasks (as discussed in the introduction) can be called private computing. In the former the user is physically present where the action is taking place, while in the latter the user is not present where the computations are taking place. The difference between these two categories is whether the software performing the action is mobile or not.

A remaining part of privacy is the user's intention. It may be possible that it is public knowledge who a user is, what data he owns and what he wants to achieve performing a certain action or setting a task. However, the user may require to keep his/her strategy (how to achieve the objective) private. In the example where a user wishes to make a hotel room reservation, it may be that his name and address together with the objective of making a reservation is information the user wishes to share with the hotel, but he is not willing to share how he decides which room to take (based on rate, view, bath or shower, etc…). The protection of someone's strategy is an essential part of the categories 3 and 4. Especially when the code is executed at a untrustworthy host, this is a difficult challenge.

Two main questions must be answered in this area of private computing.

– The first question is whether it is in theory possible to protect mobile software against privacy and security attacks while these programs are executed at untrustworthy hosts operated by the user (category 3) or operated by an unknown user (category 4). In case the answer to this question is that full protection is not possible, this may have serious implications for the success of applications that are based on mobile software techniques such as mobile software agents.
– The second question is whether the conventional cryptographic algorithms are applicable in this mobile environment. This is unlikely, so new algorithms must be developed in order to be able to provide full privacy in this environment.

The approach to answer the first question is to model the environment. The main difference between this model and conventional software models is that the execution environment cannot be trusted. It may be that the execution environment executes the code correctly, such that security is guaranteed, but if it reads the unprotected code, it may be able to determine the content and privacy is compromised. Important in this model are the locations where an attacker may be present and what the attacker's capabilities are. Some progress in this area has already been made by some of the authors [CL04]. However, many questions are still unanswered. In [CL04] a definition for perfect secrecy is given, but an encryption scheme that provides that is still unavailable. Several aspects and attacks are taken into account, but these need to be extended. A second important aspect in this model is the definition of privacy. Using this model it may then be possible to use information theory to define the theoretical boundaries of providing privacy and security.

Several solutions to the second question can be found in the literature. Not all conventional cryptographic techniques can be applied directly to protect mobile software, as these techniques are usually based on the assumption that confidential data can be left unprotected during execution because the execution environment is trusted [CLY02]. A good example is the digital signing of messages. Signing a message means that a computation must be done using a private key. The private key is information that should never be available to any other party, trusted or not [CL02]. Hence, conventional cryptographic techniques must be transformed such that they are applicable in a mobile software environment.

Within the PAW project, a realistic model will be built, such that both attacker and privacy are modelled correctly. Using this model theoretical boundaries of security and privacy may be found. This is then followed by research in the area of converting conventional cryptographic algorithms into algorithms that can be used in a mobile software environment. The developed algorithms will be incorporated in the privacy protecting architecture.

4.4 Validation and verification

Within PAW, several protocols for the exchange of privacy-sensitive data in different scenarios will be developed. For the formal verification of our communication protocols we aim to extend the results achieved in security protocol verification (Millen and Shmatikov [CE02, MS01], CSP/FDR systems [Ros95]) to build tools that verify the correctness of a protocol with respect to privacy. We aim for verification tools that show clearly which parts of a so-called "Privacy Disclaimer" are satisfied by the protocol. Moreover, we want to model attacks (forged licenses, illegitimate operations) by the data-manager or an independent attacker and verify that the system shows who violated the "Privacy Disclaimer" with respect to the submitted data. In this way we endeavour 3 important results:

1. Proof to a data-manager that in fact, using these protocols, they are sure as to not violate specific privacy regulations adapted by them or their government.
2. Guidelines for a government or a service provider as to what kind of privacy-regulations can be implemented and by which protocols.
3. Proof to a data-subject that the data-manager indeed implements the privacy regulations as expressed in the disclaimer.

Building on previous work within the PISA project [PIS03], licensing of privacy-sensitive data will be implemented in a Job-market demonstration. Privacy sensitive data is exchanged between job-seeker-agents and employer-agents using communication protocols and licenses attached to the exchanged data. To completely analyse such a system we aim to integrate the protocol verification tools with tools that verify all PETs applied to this particular system. Finally in order to fulfil the market's and government's demand for a clear insight in privacy issues and more importantly how privacy enhancing technologies can be successfully implemented, we aim to set up a framework that can evaluate applications in any ambient scenario.

We think that in the coming years it is paramount that a service-provider can check how and which privacy-regulations can be implemented in his system, and that a controlling authority on behalf of users of the service can control that the policy is correctly implemented. Very likely we will see a differentiation of privacy policies and their implementations around the different sectors of the market. Thus new dynamic test-beds that evaluate complete systems, using different assurance levels as in the Common Criteria, can provide the necessary trust in these new technologies for all participating parties.

5 Concluding Remarks

We have presented the PAW approach to building a privacy enhancing architecture. The main components of this approach are the definition of a licensing language, the development of secure licensing protocols, the integration of the licensing system into a private computing architecture, and the separate verification and validation of this architecture. In the coming years the PAW project will further elaborate and implement the ideas outlined in this paper.

References

[Aba99] ABADI, M. Secrecy by Typing in Security Protocols. *Journal of the ACM* **46**, 5 (1999), 749–786.

[AF90] ABADI, M., AND FEIGENBAUM, J. Secure circuit evaluation. *Journal of Cryptology*, 2 (1990), 5–21.

[AG97] ABADI, M., AND GORDON, A. D. Reasoning about Cryptographic Protocols in the Spi Calculus. In *Proc. 8th Int. Concurrency Theory Conf. (CONCUR)* (1997), vol. 1243 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 59–73.

[AD01] ADI, K., AND DEBBABI, M. Abstract interpretation for proving secrecy properties in security protocols. In *Electronic Notes in Theoretical Computer Science* (2001), vol. 55, Elsevier Science.

[ACJG01] ALGESHEIMER, J., C.CACHIN, J.CAMENISCH, AND G.KARJOTH. Cryptographic security for mobile code. *Proceedings 2001 IEEE Symposium on Security and Privacy, IEEE* (2001), 2–11.

[And01] ANDERSON, R. *Security Engineering: A Guide to Building Dependable Distributed Systems.* John Wiley & Sons, 2001, ch. 8: Multilateral Security, pp. 161–186.

[CG00a] CARDELLI, L., AND GORDON, A. Anytime, Anywhere Modal Logics of Mobile Ambients. In *Proc. of the 27th Ann. ACM Symp. on Principles of Programming Languages (POPL)* (Boston, MA, USA, 2000), ACM Press, New York, NY, USA, pp. 365 – 377.

[CG00b] CARDELLI, L., AND GORDON, A. D. Mobile ambients. *Theoretical Computer Science* **240** (2000), 177–213.

[CL02] CARTRYSSE, K., AND LUBBE, J. VAN DER. An agent digital signature in an untrusted environment. *Proc. of the 2nd Int. Workshop on Security in Mobile Multiagent Systems* (2002), 12–17.

[CL04] CARTRYSSE, K., AND LUBBE, J. VAN DER. Mobile code: an information theoretic approach. *Submitted to IEEE Int. Symp. on Information Theory* (2004).

[CLY02] CARTRYSSE, K., LUBBE, J. VAN DER, AND YOUSSOUF, A. Privacy protection software design, deliverable 12. Tech. rep., PISA-project, September 2002.

[Cha85] CHAUM, D. Security without identification: Transaction systems to make big brother obsolete. *Comm. ACM* **28**, 10 (1985), 1030–1044.

[Cha92] CHAUM, D. Achieving electronic privacy. *Scientific American* (1992), 96–101.

[CE02] CORIN, R., AND ETALLE, S. An improved constraint-based system for the verification of security protocols. In *Int. Static Analysis Symp. (SAS), Madrid, Spain* (2002), M. Hermenegildo and G. Puebla (Eds.), Springer-Verlag, Berlin.

[Cra02] CRANOR, L. F. *Web Privacy with P3P.* O'Reilly & Associates, 2002. ISBN 0-59600-371-4.

[CAG02] CRANOR, L. F., ARJULA, M., AND GUDURU, P. Use of a P3P user agent by early adopters. In *Proc. of the ACM Workshop on Privacy in the Electronic Society* (Washington, DC, 2002).

[CBK03] CRANOR, L. F., BYERS, S., AND KORMANN, D. An analysis of P3P deployment on commercial, government, and children's web sites as of May 2003. Tech. rep., Federal Trade Commision Workshop on Technologies for Protecting Personal Information, 2003.

[Den82] DENNING, D. E. *Cryptography and Data Security*. Addison-Wesley, 1982.

[DF02] DOMINGO-FERRER, J. Advances in inference control in statistical databases: An overview. In *Inference Control in Statistical Databases: From Theory to Practise* (2002), vol. 2316 of *Lecture Notes in Computer Science*, Springer Verlag, pp. 1–7.

[Edw96] EDWARDS, W. K. Policies and roles in collaborative applications. In *Proc. of the 1996 ACM Conference on Computer Supported Cooperative Work (CSCW)* (Boston, MA, 1996), ACM, ACM Press, pp. 11–20.

[GIKM98] GERTNER, Y., ISHAI, Y., KUSHILEVITZ, E., AND MALKIN, T. Protecting data privacy in private information retrieval schemes. In *Proc. of the 30th Annual ACM Symposium on Theory of Computing (STOC)* (Dallas, TX, 1998), ACM Press, pp. 151–160.

[HILM02] HACIGUMUS, H., IYER, B. R., LI, C., AND MEHROTRA, S. Executing SQL over encrypted data in the database service provider model. In *Proc. of the ACM SIGMOD/PODS Conference* (Madison, WI, 2002), pp. 216–227.

[HIM02] HACIGUMUS, H., IYER, B. R., AND MEHROTRA, S. Providing database as a service. In *Proc. of the 18th International Conference on Data Engineering (ICDE)* (San Jose, CA, 2002), IEEE Computer Society, pp. 29–40.

[HO02] HALPERN, J. Y., AND O'NEILL, K. Secrecy in multiagent systems. In *Proc. of the 15th IEEE Computer Security Foundations Workshop* (2002), pp. 32–46.

[HO03] HALPERN, J. Y., AND O'NEILL, K. Anonymity and information hiding in multiagent systems. In *Proc. of the 16th IEEE Computer Security Foundations Workshop* (2003), pp. 75–88.

[HSK99] HELME, A., AND STABELL-KULØ, T. Offline delegation. In *8th USENIX Sec. Symp.* (Washington, D.C., USA, 1999), USENIX, pp. 25–33.

[HJW02] HOGBEN, G., JACKSON, T., AND WILIKEN, M. A fully compliant research implementation of the P3P standard for privacy protection: Experiences and recommendations. In *Proc. of 7th European Symposium on Research in Computer Security (ESORICS)* (2002).

[Hoh98] HOHL, F. Time limited blackbox security: Protecting mobile agents from malicious hosts. *Mobile agents and security, Lecture notes in computer science* (1998), 92–113.

[JM95] JAJODIA, S., AND MEADOWS, C. Inference problems in multilevel secure database management systems. In *Information Security: an integrated collection of essays*, M. Abrams, S. Jajodia, and H. Podell (Eds.). IEEE Computer Society Press, 1995, pp. 570–584.

[JLS02] JARECKI, S., LINCOLN, P., AND SHMATIKOV, V. Negotiated privacy. In *Proc. of the International Symposium on Software Security (ISSS)*, vol. 2609 of *Lecture Notes in Computer Science*. Springer–Verlag, 2002, pp. 96–111.

[Lan02] LANGHEINRICH, M. When trust does not compute - the role of trust in ubiquitous computing. In *Proc. of the UBICOM 2003 Workshop on Privacy* (Götemborg, Sweden, 2002).

[Lan03] LANGHEINRICH, M. Privacy invasions in ubiquitous computing. In *Proc. of the UBICOM 2002 Workshop on Privacy* (Washington, DC, 2003).

[LDM02] LEDERER, S., DEY, A. K., AND MANKOFF, J. Towards everyday privacy in ubiquitous computing environments. In *Proc. of the UBICOMP 2002 Workshop on Socially-informed Design of Privacy-enhancing Solutions* (2002).

[LHJ+03] LEDERER, S., HONG, J. I., JIANG, X., DEY, A. K., LANDAY, J. A., AND MANKOFF, J. Towards everyday privacy for ubiquitous computing. Technical Report UCB-CSD-03-1283, Computer Science Division, University of California, Berkeley, 2003.

[LMD03] LEDERER, S., MANKOFF, J., AND DEY, A. K. Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing. In *Proc. of the ACM Conference on Human Factors in Computing Systems (CHI)* (Ft. Lauderdale, FL, USA, 2003), pp. 724–725.

[LMDB03] LEDERER, S., MANKOFF, J., DEY, A. K., AND BECKMANN, C. P. Managing personal information disclosure in ubiquitous computing environments. Technical Report UCB-CSD-03-1257, Computer Science Division, University of California, Berkeley, 2003.

[LM99] LOUREIRO, S., AND MOLVA, R. Privacy for mobile code. In *Proceedings of the distributed object security workshop of OOPSLA* (1999), pp. 184–99.

[LRSW99] LYSYANSKAYA, A., RIVEST, A., SAHAI, R., AND WOLF, A. Pseudonym systems. In *Selected Areas in Cryptography (SAC)* (1999), Springer–Verlag, pp. 184–99.

[Mea96] MEADOWS, C. A. The NRL protocol analyzer: an overview. *Journal of Logic Programming* **26**, 2 (1996), 113–131.

[MS01] MILLEN, J., AND SHMATIKOV, V. Constraint Solving for Bounded-Process Cryptographic Protocol Analysis. In *Proc. of the 8th ACM Conference on Computer and Communication Security* (2001), pp. 166–175.

[PIS03] PISA CONSORTIUM. *Handbook of Privacy and Privacy-Enhancing Technologies, the case of intelligent software agents.* College Bescherming Persoonsgegevens, The Hague, 2003.

[Ros95] ROSCOE, A. W. Modelling and verifying key-exchange protocols using CSP and FDR. In *CSFW: Proceedings of The 8th Computer Security Foundations Workshop* (1995), IEEE Computer Society Press, pp. 98–107.

[ST98] SANDER, T., AND TSCHUDIN, C. F. Protecting mobile agents against malicious hosts. In *Mobile Agent Security* (1998), vol. 1419 of *Lecture Notes in Computer Science*, pp. 44–60.

[SYM99] SANDER, T., YOUNG, A., AND MOTI, Y. Non-interactive cryptocomputing for (NC_1). *40th Ann. Symp. on Foundations of Computer Science (FOCS)* (1999), 554–66.

[Sch98] SCHNEIDER, S. Verifying Authentication Protocols in CSP. *IEEE Transaction on Sofware Engineering* **24**, 8 (1998), 743–758.

[SS96] SCHNEIDER, S., AND SIDIROPOULOS, A. CSP and Anonymity. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, vol. 1146 of *Lecture Notes in Computer Science*. Springer-Verlag, 1996, pp. 198–218.

[SBM03] SCOTT, D., BERESFORD, A., AND MYCROFT, A. Spatial security policies for mobile agents in a sentient computing environment. In *Proc. of the Fundamental Approaches to Software Engineering (FASE)*, M. Pezzé (Ed.), vol. 2621 of *Lecture Notes in Computer Science*. Springer–Verlag, 2003, pp. 102–117.

[Shm02] SHMATIKOV, V. Probabilistic analysis of anonymity. In *Proc. of the IEEE Computer Security Foundations Workshop (CSFW)* (2002), pp. 119–128.

[SNS88] STEINER, J. G., NEUMANN, B. G., AND SCHILLER, J. I. KERBEROS: An authentication system for open network systems. In *Proc. Winter 1988 Usenix Conference* (1988), pp. 191–201.

[THG98] THAYER, J., HERZOG, J., AND GUTTMAN, J. Strand spaces: Why is a security protocol correct? In *Proc. of the 19th IEEE Computer Society Symposium on Research in Security and Privacy* (1998), IEEE computer Society.

[Yee99] YEE, B. A sanctuary for mobile agents. *Secure Internet Programming, Lecture notes in computer science* **1603** (1999), 261–73.