Safeguarding Personal Data using Rights Management in Distributed Applications

Adolf Hohl[1] and Alf Zugenmaier[2]

[1] University of Freiburg, adolf.hohl@iig.uni-freiburg.de,
[2] DoCoMo Euro-Labs, zugenmaier@docomolab-euro.com

Abstract. Privacy includes the right to determine the use of personal information after it has been released. Some compliance solutions have been proposed already. However, they are usually monolithic systems operating only within one database system or requiring a customized infrastructure. This paper explores the possibility to use an off-the-shelf document rights management platform to enable enforcement of usage policies. First experiences from a building a demonstration application are encouraging.

1 Introduction

Fears of users about misuse of their personal data can prevent acceptance of new services and technologies. This is especially the case, when software acting on behalf of the user can autonomously release sensitive information to communicating partners or services. Nearly everybody has had experience of misused personal information in the Internet such as unwanted advertisements and spam. This is only the tip of the iceberg. More serious abuse of the information may involve selling it to rating agencies, resulting in unwanted "personalization" of prices, interest rates, denial of credit, etc.

Therefore, it is essential that services handle their users' personal data with care and can communicate this fact to the service users. If it is not possible to ensure this, fear of misuse and privacy concerns remain with the user.

In this paper, we address the problem of giving service users more control over their data after they are transmitted, e.g., during the use of a service or an application. As a running example, we make use of a sales scenario in which a person is interested in buying a car. To improve the service quality to the user the car dealer can combine his car offering with a suitable car insurance and provide a finance offering.

We assume a very simple policy to avoid misuse: use the data only for providing an offer and delete the data afterwards. This policy is also called oblivion[1, 2].

Part of this work was funded by the DFG / Gottlieb Daimler and Carl Benz Foundation. We would like to thank Caspar Bowden of Microsoft EMEA for his comments on an earlier version of this paper.

The contribution of this paper is to show the feasibility and explore the practical problems of using rights management for privacy aware managing of personal data as proposed by Korba and Kenny [3].

The paper is structured as follows: The next section discusses an overview of the approach. Section 3 describes the application implementation. We briefly evaluate the performance in Section 4. A discussion of these results takes place in Section 5. Related work is described in Section 6. Conclusions conclude the paper.

2 The Approach

Korba and Kenny observed in in [3] that the interests a service user has in dealing with sensitive data are similar to those of providers of copyrighted digital contents. Both, the copyrighted content provider and the service user, i.e., the personal data provider, are interested in making data available only for limited use and processing. Furthermore, unauthorized onward transmission and use should be prevented. Subsequently, control over transmitted data or contents has to be enforced.

Sensitive personal data is therefore sent in a protected way to the service provider thus preventing unauthorized usage and information leakage. This encrypted data has a license attached to it when communicated to the service providers. The license limits the use of this personal data. The service user now takes the role of a content provider and license issuer. Because it would be unmanageable if every service user had her own slightly different license attached to her data interest groups should act as liaison and offer standardized licenses.

This is an orthogonal approach to classical anonymization techniques with the concepts of data minimality and data obfuscation. This attacker model assumed here is weaker than the model usually assumed for privacy enhancing technologies. We expect some level of cooperation by some service providers. The service user needs to find out which service providers are willing to cooperate. Another new aspect of this attacker model is that the service provider is not seen as one atomic entity. There may be parts of the service providers organization that could be more trustworthy than others.

3 Technical Solution

In our proof of concept implementation we built a client – server application for the car sales scenario to make and serve a request using rights protected personal data while at the same time adhering to policies. The personal data is encrypted and the usage license for the service provider is created and issued. It is the duty of the application, respective the developer of it that it adheres to the semantic of a specified right or a corresponding policy. To ensure this property, a review by an independent party should certify the source code. The

source code could also be published to give somebody the chance to do this task.

Despite the obvious need for a hardware root of trust such as a Trusted Computing Platform[4], they are not widely deployed.

Therefore, we have chosen a rights management framework on the Windows platform which currently does not support the level of security as hardware based approaches but is widely available. In addition to dealing with digital rights, this framework provides similar primitives as realized by TC-Platforms by using a hardened software implementation[1]. At this point we don't focus on the security of this hardened implementation. Instead we speculate that with the availability of Trusted Computing Platforms this could be improved easily. We focus on the services the framework provides and how they can be used for privacy protecting applications.

The framework implements necessary requirements for distributed access control on an application level. This includes the *application identification and signing* to detect if a rights managed application is tampered with. This is necessary to prevent a tampered application making use of a granted right. A *secure storage* is realized by binding keys and licenses to the platform by encrypting it with a platform specific key. Data could be bound to dedicated platform configurations and users. This is realized by *authentication framework* for *platform attributes*(e.g. the absence of a screen-grabber) and the *user*.

3.1 The Used Rights

In the selected scenario, we implement two rights. The right *VIEW* is based on a built in right of the framework and decrypts protected content. The semantic interpretation of this right here is to view personal data and use it for the calculation of a car offer. Despite the fact that the right is called view, the application does not allow a sales clerk to view the personal information. The right *ANONYMIZE:* is specified by our own. It is introduced because there is no delegation feature in the rights management (RM) framework. From a functional point of view, the framework treats it as the *VIEW* right and provides decryption solely. Special functionality must be defined in the conforming application. If *ANONYMize* is specified as a right, the car dealer is allowed to transmit user data only after removing identifying information.

3.2 The User's Application

The user's application protects the personal information for use by the service application. While the representation of the data similar to a *vcard* address form remains, this content is encrypted and suitable licenses for the consumers are issued. A detailed description of this procedure is in [5]. In Figure 1 the necessary

[1] Because of the limitations in protecting code this is mainly done by security through obscurity such as anti-debugging techniques and embedded secret keys in libraries

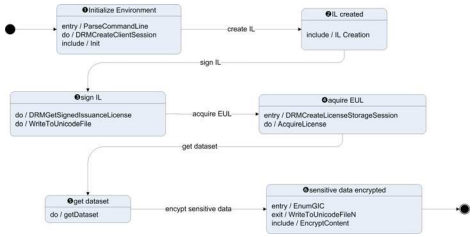

Fig. 1. State chart of the user's application

steps are visualized in a state chart of the application. After the initialization of the environment and the RM-framework (step 1) the application prepares an *Issuance License, IL* and grants rights with a validation date to the principal under whose ID the server application is running (step 2). The Issuance License is signed. Therefore a prior acquired *Client Licensor Certificate, CLC* from the license server is loaded to sign the Issuance License and make it to a *Signed Issuance License, SIL* (step 3). To encrypt the content an *End User License, EUL* is derived from the IL which allows to extract the encryption key (step 4). After this step, the personal data is read in (step 5) and encrypted by the framework (step 6). It is ready for transmission to the service together with its SIL.

3.3 The Service Application

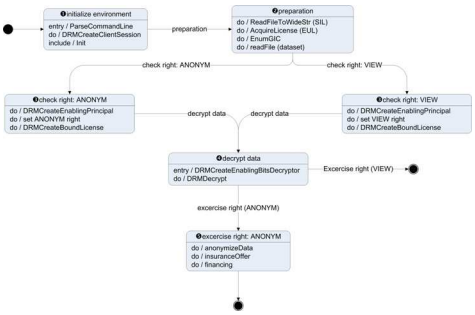

Fig. 2. State chart of the service application

The service acts from a rights management viewpoint as a consumer of protected content and licenses. Therefore the service application uses the RM-framework for decrypting and consuming content in a granted context. The service application logic has to ensure that it does not leak any personal data.

The user has to rely on the assurance that a particular application really adheres to the specified rights. Figure 2 visualizes the states of the service application. First, the environment and the RM-framework is initialized (step 1) and the protected personal data set and the SIL is loaded. The license server is contacted and a EUL, corresponding to the SIL is requested (step 2). The application checks the EUL for the granted rights *VIEW* and *ANONYMIZE* (step 3). Using a granted right, the content is decrypted and processed with the method implementing the purpose for the right specified (step 4,5).

4 Implementation Evaluation

Because the evaluation of the security of the rights management framework is out of scope for this paper, the performance of our implementation was measured basically in order to figure out very time consuming phases. This is important if one envisions large number of transactions privacy protected with rights management. We could not identify such a time consuming phase which would make it unfeasible to use. When a service supports several users at the same time the time consumption of step 2, 3 and 4 on the service side are of special interest. Processing of these steps are necessary for each user.

For our measurement, the code of our application was instrumented with a time reporting procedure at important phases.

Issuing of the license took just under two seconds, while consumption (steps 2, 3 and 4) took 0.7 seconds. The issuing part of the application contacts the license server via the network when it acquires a CLC, accounting for 1.3 seconds of this time. Therefore this value depends on the round-trip-time of the network and the workload of the license server. The same appears when the consuming application acquires an EUL. Under heavy load of the license server this time can increase. However, our test setup was not designed to stress test the server.

5 Discussion

The implementation represents a first step towards using cooperative mechanisms to protect the privacy of users which are reported and enforced technically by the service provider. The operating system provides a rights management framework for distributed access control at the application level. Currently the root of trust of the framework is software based, but will hopefully support a hardware attested *Trusted Computing Base* in the future.

Limitations by design is the effort necessary for a check of an implementation for processing a certain right is compliant with the semantic meaning of this right. Currently the rights *VIEW* and *ANONYMIZE* can be granted by a service user. Their compliance has to checked by the service user or a trusted independent party. If the code of the service application is publicly available, everybody can check that it does what it claims. Our implementation could easily

be extended to rely on third party certification of the code. Another alternative that can also handle legacy applications would be to sandbox the service application, to grant the license to the sandbox, which then limits the capabilities of the application to store or communicate data. Programming languages with information flow tracking capabilities can simplify this procedure, because they detect inferences between confidential data and unclassified data.

The performance data as measured is not great. However, one can expect service pack 1 of the rights management platform to massively improve performance as the platform verification is optimized and does not require Internet connectivity any more. In addition, as rights management performs distributed access control, there is only one central performance bottleneck: the rights management server. Under the assumption that rights management is also used for other digital content (such as music), it would be surprising if this infrastructure could not scale similarly for privacy protection.

6 Related Work

There has been lots of work on the regulatory aspects of privacy, such as EU data protection legislation, HIPAA, Gram-Leach-Bliley Act, or the safe harbor agreement. All of these regulations set the backdrop against which all technical work will be evaluated by the marketplace. Related technical work covers three main areas: expression of privacy policies, negotiation of policies and enforcement of policies. The World Wide Web Consortium standardized in its platform for privacy preferences (P3P) project an exchange format for privacy preferences of web server users [6]. It also defines a protocol by which the users preferences can be compared with those of the server and support for negotiation. A big drawback of the original P3P specification was that the policies were described on a level of detail that was not understandable to the general public. Even for an expert it is difficult to determine what the effect of a given policy can be. Compact policies [7] try to remedy this situation by defining default settings that can be given a meaningful name.

To remedy the fact that P3P policy specifications are very web centric, E-P3P [8, 9] tried to generalize the policies to enterprise applications. It then evolved to EPAL [10] which was submitted to W3C.

Negotiation can take place over policies expressed in P3P or E-P3P. The fundamental reason for this is that their underlying framework permits comparison of policies. The language APPEL [11] was designed to enable preference specification for negotiation of P3P policies.

Our work does not try to design a new privacy policy language. For our prototype we stated the policy in self defined terms. However, our design allows to include any policy interpretation engine.

On the enforcement side, hippocratic databases [12] were a first step towards enabling privacy aware data processing. The concept of a hippocratic database is that every information element in the database is tagged with a privacy policy.

Whenever data is retrieved form the database, the privacy policy is returned as metadata. It is then up to the application retrieving the data to adhere to the policy. The hippocratic database concept also includes the idea of doing an audit module to trace privacy breaches. This work restricts itself to database level, dealing only with one database.

Work at HP [13] adds the aspect of tracing data access by encrypting the returned data, which includes the privacy policy, with identity based encryption. An audit trail can be built up by logging the requests for keys and where access to the data was required. By using a tagging OS they propose to limit the flow of information marked as sensitive. An implementation of this concept is described in later papers [14, 15]. In these papers the architecture described contains a reference monitor doing access control at the database server and an enforcer module which is not described to greater detail. This work again is focussed on a single large enterprise.

An approach to dispose of this central reference monitor is described by Korba and Kenny [3]. The key observation is that digital rights management and management of personal information are very similar. Their paper gives an analysis of the entities involved in DRM and in data processing of personal data and the relation between them.

Our work shows the feasibility of the approach of Korba and Kenny by presenting an implementation. We chose the Microsoft information rights management framework which has a widely available software development kit. Enforcement of privacy policy is done at the application level in a distributed fashion. It is therefore possible to implement this approach across multiple domains. We also have the advantage over the HP approach that we do not leak information about accesses to the personal data which would produce privacy sensitive information at the IBE key server. If audit trails are required the application in question can produce the required information directly. Another aspect that is not mentioned in the work done at IBM and only mentioned in passing in the work done at HP is the concept of attestation.

Langheinrich's work [16] tries to tackle the problem in extremely distributed settings such as pervasive computing. His approach relies on cooperation of the entities being asked to refrain from recording or keeping personal information.

7 Conclusions

The results from the first trials are encouraging and lead us to believe that mechanisms similar to rights management can be used to enforce privacy. Because service application certification does not scale well, it seems not to be a general approach but to provide solutions in cases where certified privacy conforming components can be reused, e.g. for sandboxes. It is also thinkable to use certified modules for querying servers in client/server-scenarios where a client has obligations concerning to the usage of his submitted data.

Additionally, this approach can be used to shift the work of ensuring the correct handling of data from the person installing and maintaining the computing environment to the software vendor for the service application.

We believe it was a useful exercise to try to validate the proposal by Korba and Kenny by implementation. Only in this way the missing links (such as delegation) became obvious. In conclusion, it can be said that the fundamental idea may work, but the rights management platforms need to be tailored accordingly.

References

1. Zugenmaier, A., Claessens, J.: Privacy in Eletronic Communications. In: Network Security. IEEE Press (to appear)
2. Stajano, F.: Will your digital butlers betray you? In: WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society, New York, NY, USA, ACM Press (2004) 37–38
3. Korba, L., Kenny, S.: Towards Meeting the Privacy Challenge: Adapting DRM. (2002) ACM Workshop on Digital Rights Management.
4. Trusted Computing Group: TCG Backgrounder. (2003)
5. Hohl, A., Zugenmaier, A.: Safeguarding personal data using rights management in pervasive computing for distributed applications (to appear)
6. Clarke, R.: P3p re-visited. In: Privacy Law and Policy Reporter. (2001) 81–83
7. Cranor, L.F., Lessig, L.: Web Privacy with P3p. O'Reilly & Associates, Inc., Sebastopol, CA, USA (2002)
8. Ashley, P., Hada, S., Karjoth, G., Schunter, M.: E-P3P Privacy Policies and Privacy Authorization. In: Proc. 1st ACM Workshop on Privacy in the Electronic Society (WPES). (2002) 103–109
9. Karjoth, G., Schunter, M., Waidner, M.: The platform for enterprise privacy practices - privacy enabled management of customer data. In: 2nd Workshop on Privacy Enhancing Technologies (PET 2002). LNCS, Springer (2003) 69–84
10. Karjoth, G., Schunter, M., Waidner, M.: Privacy-enabled services for enterprises. In: DEXA Workshops. (2002) 483–487
11. Langheinrich, M., Cranor, L., Marchiori, M.: APPEL: A P3P preference exchange language. W3C Working Draft (2002)
12. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic Databases. In: 28th Int'l Conf. on Very Large Databases (VLDB), Hong Kong. (2002)
13. Mont, M., Pearson, S., Bramhall, P.: Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services. (2003) HPL-2003-49.
14. Mont, M., Thyne, R., Chan, K., Bramhall, P.: Extending HP Identity Management Solutions to Enforce Privacy Policies and Obligations for Regulatory Compliance by Enterprises. Technical Report HPL-2005-110, HP Laboratories Bristol (2005)
15. Mont, M., Thyne, R., Bramhall, P.: Privacy Enforcement with HP Select Access for Regulatory Compliance. Technical Report HPL-2005-10, HP Laboratories Bristol (2005)
16. Langheinrich, M.: A Privacy Awareness System for Ubiquitous Computing Environments. (2001)