INSTITUTO SUPERIOR TÉCNICO

Privacy for the Personal Data Vault

SUMMARY

Tamás Balogh

July 13, 2014

1 INTRODUCTION

The majority of interactions on today's internet is driven by personal user data. These information pieces come in different shapes and forms, some being more valuable than others. For example, banking details might be considered more valuable than a person's favourite playlist. What all of these data pieces have in common is that they all belong to some specific user. This property, however, is not reflected in how data is hosted and organized over the web, since the hosting entities of personal user data consists of multiple service providers. Data belonging to a single user is fragmented and kept independently under different control domains based on the context. For example data related to somebody's social life might be stored in some social network provider, while the same person's favourite playlist might be hosted by his music provider service. Different initiatives exist to unify these scattered data. The Personal Data Vault (PDV) can be considered one of such proposed solution.

The PDV is a user-centric vision of how personal digital data should be hosted. Rather than having bits of informations scattered around multiple sites, the PDV tries to capture these under a single control domain. Every user is associated with his own PDV where he hosts his personal data. PDVs are not only secure storage systems, but also offer ways to make access control decisions on hosted data. External entities, such as different service providers, can request user data at the user's PDV, in order to provide some functionality beneficial for the owner of the PDV. By unifying the source of the personal user data, we are expected to achieve a more flexibility and better control over how data is being disclosed. By employing an access control solution users can have assurance that only authorized entities are going to get access to their data. It does not, however, provide any privacy guarantees with regard to how personal data is being protected after it leaves the control domain of the PDV.

PrimeLife was a European project [3] that researched technical solutions for privacy guarantees. Their privacy enhancing model introduces a novel privacy policy language, which empowers both users and service providers to specify their intentions with regards to data handling. The privacy policy language, however, lacks the technical enforcement model needed to support its correct functioning. This enforcement model is required to provide trust and assurance to end users. A trust relationship needs to be established between remote entities prior to personal data exchange, while assurance needs to be provided as proof that user intentions have been respected.

We propose a novel privacy policy enforcement model with an integrated trust and assurance framework. Our solution utilizes the completely decentralized construct of a Distributed Hash Table (DHT) to sustain a mediated space between PDVs and service providers. This mediated space serves as a platform for privacy enhanced data sharing. Pointers to the shared data objects, which live in the mediated space, are kept by both the owner and the requester. This way data owners can stay in control over their shared data. A distributed logging mechanism supports our enforcement model in delivering first hand assurance to end users.

2 Background: The Personal Data Vault

The Personal Data Vault also appears under various other terminologies, like “Personal Data Store” or “Personal Data Locker” [7]. The attempts to formalize the concept of a PDV are complementary in the sense that they all try to focus on providing a better control over personal data for the end user. However, a clear formalization of the term is still missing, since the research projects involving PDVs are built with different aims in mind. Some of them conceptualize a raw storage service with the only purpose to host data securely, while others focus on providing software solutions to manage already existing storage spaces or even link different user accounts.

Since security is a central concern of all of these projects, they mostly come with an additional data access layer on top of the storage system. This access layer facilitates the interoperability between different entities in a secure manner. The fine grained control can be achieved through the use of access control mechanism that rely on predefined policies. These policies can either be preformulated by the end user, or constructed on the fly.

Another key focus point that these projects follow is the interoperability of different entities [5]. PDVs should integrate seamlessly with other entities and facilitate the secure sharing of data across different control domains. The security of these operations can be guaranteed by providing encrypted channels between entities. These interactions can be of multiple types depending on the acting sides. Person-to-person connections are trying to connect individuals: independent entities that serve as representative hosts for a person. Person-to-community solutions try to formulate groups of persons depending on some social context. Person-to-business connections are describing how individuals are interacting with different service providers. In order to achieve these features interoperability needs to be provided, that overcomes the differences in the underlying data model with the aid of standardized APIs and protocols.

For the purpose of this thesis work we are going to treat PDVs as abstract entities made out of two layers: a data layer and a manager layer. We consider these to be entities made out of a single or multiple machines with high availability. Moreover, we consider them resilient in face of failure and secure in face of vulnerabilities and exploits that could be used directly by a potential attacker. Herein we disregard these security aspects, and focus on the privacy concerns that appear in interoperability scenarios.

3 Related Work: The PrimeLife Project

The PrimeLife Project was a research project conducted in Europe under the Seventh Framework Programme (FP7) [3], concerned with privacy and identity management of individuals. They are addressing newly appearing privacy challenges in large collaborative scenarios where users are leaving a life-long trail of data behind them as a results of every interaction with services. Its extensive research domain investigates privacy enhancing techniques in areas such as policy languages, infrastructure, service federation and cryptography.

One of its major contribution is the investigation and design of a suitable policy framework that encompasses the privacy features which promote user-centrality and control of private data. The proposed solution is centred around the development of the PrimeLife Policy Language (PPL) [9] which is a proposed extension of the existing XACML standard [2].

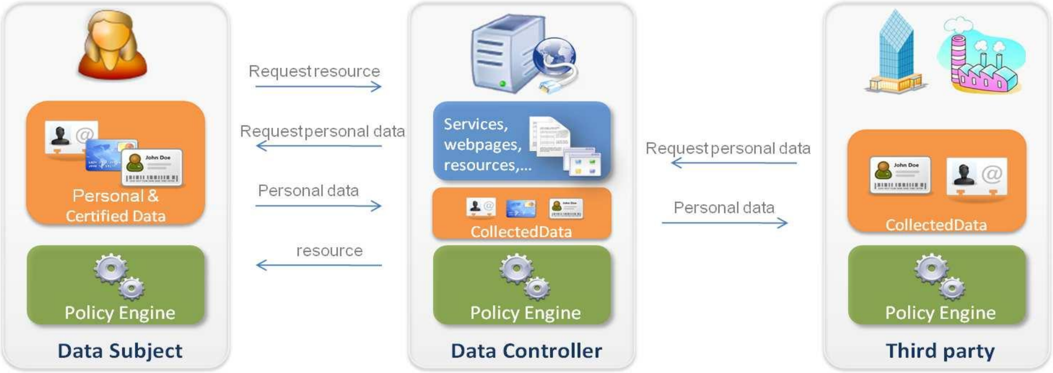

Figure 3.1: Collaboration Scenario[1]

The core idea of how PrimeLife is intended to use PPL to facilitate privacy options can be described using a simple collaboration diagram in Figure 3.1. The scenario describes the interaction between the Data Subject (DS), who is considered the average user or data owner whose privacy needs protection; Data Controller (DC), which denotes a wide range of service providers that the user can be interacting with; and the Third Party, who is considered to be another entity involved in the business process, like an associate of the service provider. The interaction is initiated by the Data Subject who is requesting some sort of resource from the DC. The DC responds with its own request, describing what kind of information he expects from the user in exchange for the resource, and how he is willing to treat that information. The description provided by the DC on how he will treat private personal data is called Data Handling Policy (DHPol). The DS examines the list of information requested together with the DHPol, and combines it with his own Data Handling Preference (DHPref). The DHPref is the user's way to describe how his personal disclosed information is preferred to be treated. A combination between the DHPol and

[1] Figure 3.1 source: http://primelife.ercim.eu/images/stories/deliverables/d5.3.4-report_on_design_and_implementation-public.pdf

DHPref results in a Sticky Policy that is sent together with the requested personal data, in exchange for the resource. The Sticky Policy contains all the relevant data protection rules which have to be respected by the DC. The direct collaboration between DS and DC ends here. However, the DC may decide to forward the collected personal data from the DS to a Third Party. In this case, the DC has to consult the Sticky Policy first, in order to examine whether he is allowed to forward the information collected from DS or not, and act accordingly. In order to support such a scenario an expressive language is needed. The PPL is a highly descriptive and easily extendible language that can support the collaboration scenario described above.

This thesis work considers the PrimeLife Policy Language (PPL) as its main tool by which privacy guarantees are provided. However, instead of focusing on the language components of the PPL, it targets the enforcement model that can be used together with it.

4 Requirements

This thesis work is concerned with proposing different privacy enforcement models built on top of the PrimeLife research. The requirements that these enforcement models have to fulfill are as follows:

1. **Establishing trust** relationship between actors, like service providers and data owners. Trustworthiness refers to the degree of assurance in which an actor can be trusted to carry out actions that he is entrusted with.

2. **Transparent user data handling** should be a priority for every Data Collector. Users need to get assurance that their preferences on how to handle their data are carried out by the actors.

3. **Data protection across multiple control domains** is needed in order to facilitate the safe interoperability of multiple Data Controllers. Delegation of rights to forward user data is a common use case, therefore there should be a clear model that describes how delegations take place, and how does the data protection rules apply to the third party who receives the data.

4. **Maintaining control** over distributed data promotes user centrality. In the user-centric model the owner of the personal data is considered to be the user, even in the case when he chooses to share it with other parties. He must have a way to continue his rights to exercise operations on his personal data, such as: modifications, revocation of rights, deletion, etc.

5 System Design

5.1 Verifiable Privacy

This model is constructed based on the generalization of multiple research projects on data privacy involving enhanced hardware and monitoring services [6][8]. As seen in Figure 5.1 every machine of the system comes equipped with a TPM [1] which is responsible for authenticity checks and key distribution. Another important feature

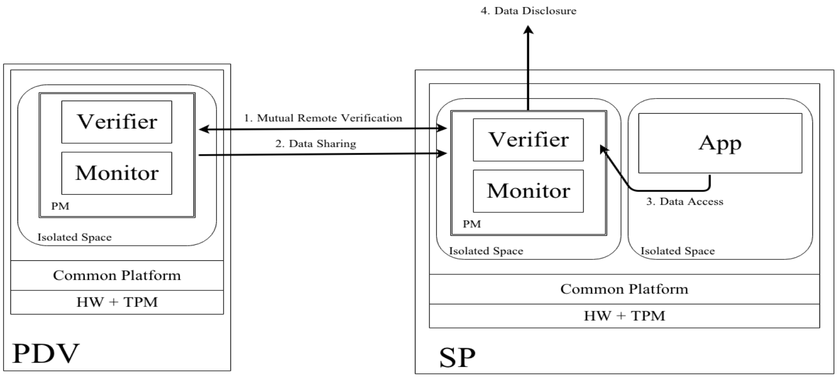

Figure 5.1: Verifiable Privacy: Interaction diagram between a PDV and a Service Provider (SP)

of this model is that every running application is sandboxed in its own isolated space. This can either be achieved with process or system virtualization. Since application are contained within this sandbox any interaction with the outside world can be monitored.

The core component of this model is the PM which sits in its own isolated space on every machine, both on PDV and SP side. The PM is responsible to establish initial trust between actors, enforce Sticky Policies on remote platforms, and monitor every interaction of locally running applications.

Trust is established by software verification that is attested by the TPM. Every application that comes in contact with shared data, even the PM, has to pass a verification test before the access to the data is granted. In forwarding scenarios the PM is responsible for distributing shared data, and keep track of it in a forwarding chain. The Forwarding Chain is a tree-like structure of nodes that share a copy of a user data object. The root of the Forwarding Chain is the source of the private data, which is a PDV in our case. Every node of the chain is responsible for keeping track of the forwarding path, thus maintaining a distributed view by which every data copy can be tracked and identified. The tracking and identification of different data copies is important in order to maintain control of previously shared data.

The monitoring service is responsible for keeping logs of every event happening in the system. Both internal and external communication channels between applications are closely inspected by the monitor in order prevent data from exiting the system unsupervised. Logs are also kept in the forwarding chain. An external Trusted Third Party (TTP) is responsible for aggregating and verifying the logs from every forwarding chain.

5.2 Trusted Privacy

Much like the Verifiable Privacy, the Trusted Privacy relies on the use of a specialized software: the Privacy Manager (PM). The architecture supporting the PM component is relaxed by employing a middleware oriented design [4]. Apart from the basic Sticky Policy enforcement that is guaranteed by the PM, it comes with a different view on the employed trust framework. As its name suggests, the Trusted Privacy model relies on the correct functioning of the trust framework, which is the composition of two independent

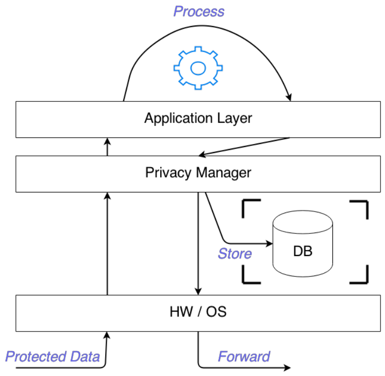

Figure 5.2: Trusted Privacy: Interaction Model of the Data Flow

sources of trust.

With the introduction of a new component into the PM, called Trust Negotiator, users can evaluate the trustworthiness of an entity they are about to interact with. The sources and mechanism by which trust is evaluated are the Trust Seals and Reputation Systems. Trust Seals, certified by TTPs , are combined into a trust score. Reputation systems, such as customer feedback services or blacklist providers, are used in order to derive a reputation score. The intuition behind the outsourcing of trust to multiple sources, is that many independent trust scores from independent authorities can complement or cancel out each other, leaving the end user with a trustworthy estimate.

The shared user data can take multiple paths once it has been disclosed to an external entity. The Figure 5.2 describes how user data is handled by a service provider. Shared user data is passed through the PM to the Application Layer which carries out the service provider logic. Two usual use cases include storing and forwarding of the processed data. Both of these operations have to pass through the PM middleware in order to evaluate whether they are allowed to be stored or forwarded, respectively. The evaluation is carried out based on the Sticky Policies attached to the data objects. The PM only lets data through for applications which are authorized to operate on the requested data.

Similarly to the Verifiable Privacy (VP) model, this also uses the Forwarding Chain as its platform for keeping track of exising data copies and data usage logs. We introduce a slight deviation, however, in the way that logs are aggregated and verified, by eliminating the requirement of a TTP. The aggregation of logs is the responsibility of the original data requester who is in direct contact with the PDV. The PDV initates a pull request for logs that is forwarded over the chain. Using this schema, the log verification can be carried by each PDV.

5.3 Mediated Privacy

The Mediated Privacy sticky policy enforcement model makes use of a mediated space between DSs and DCs, on which shared data lives. The idea of a mediated space can easily be captured by the concept of a Distributed Hash Table (DHT) [10]. DHTs are decentralizes overlay networks, where each node is seen as equal. Nodes forming this overlay are responsible to maintain a predefined keyspace, meaning that every node is responsible for a subset of the keyspace, called the keyspace slice. New data is entered

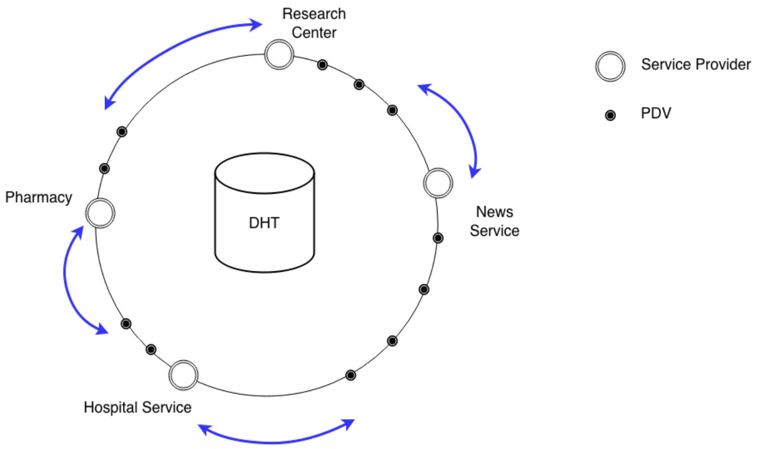

Figure 5.3: Mediated Privacy: Business Ring formed around a healthcare scenario

under a key in the DHT, called the lookup key, which is hashed in order to compute its place on the keyspace. Its place in the keyspace determines the node which will host the data physically.

We introduce the concept of a Business Ring. We propose a solution where Business Rings are spawn as needed around a group of services that have a closely integrated business model. Service providers belonging to the same Business Ring are assumed to have an existing business agreement, which ties them together. For example the Business Ring used in case of the healthcare scenario using Personal Health Record (PHR)s, could be formed according to Figure 5.3. The black nodes are representing PDVs, the white nodes are the service providers and the arrows mark the keyspace slices assigned for each of them. The ring-like representation of the DHT from Figure 5.3 resembles a Chord network [11]. The business model that ties the service providers together in Figure 5.3 could be the public health services provided to users. These service providers, although they offer independent services, belong to the same logical ring, since they operate on the same set of PHRs. Together they form a clear business model, which is used as a basic characteristic of the Business Ring.

Every DHT node follows the design of an architecture on three layers. The bottom layer, which serves as the base for the other two layers, incorporates all the conventional DHT functionalities. This includes the maintenance of the overlay topology, and the serving of basic operations, such as: insert and retrieve. The Privacy Manager layer, on top of the DHT layer, is responsible for the safeguarding of protected data objects via their Sticky Policies, and trust establishment. The Logging layer sits on top of the stack and is responsible for keeping track of every DHT event regarding operations on private data.

The interaction model presented in Figure 5.4 focuses on the data flow between a single DS and DC. In the first step, the DC makes his request to the DS together with the Data Handling Policy (DHPol) and a *LookupKey*, defining his intentions on data handling and the key under which the requested data is expected. The *LookupKey* is a valid key in the Business Ring, residing under the DC node's keyspace slice. After the received request, the DS interrogated the Business Ring for relevant details about the DC. These could include information on his keyspace size, and other trust measures, which contribute

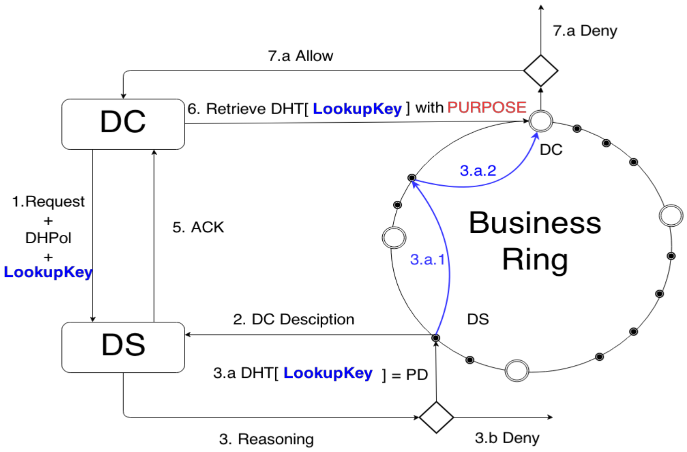

Figure 5.4: Mediated Privacy: DC - DS interaction model

into the reasoning in step 3. Depending on the trust level and the predefined data policies of DS the reasoning can have two outcomes. In step 3.a access is granted and a Protected Data (PD) object is created, in step 3.b it's denied. After granting access the DS issues an insert operation to the Business Ring in step 3.a. The insert request tries to put the PD under the *LookupKey* provided by the DC. Steps 3.a.1 and 3.a.2, marked with blue arrows, represent the internal routing steps of the DHT. Once the request reaches the DC's node, the DS sends an ACK back to the DC with the status of the operation.

When the DC wants to access the PD for processing, he issues a request to his Business Ring node in the form of a *local retrieval* operation. The request asks for the PD under *LookupKey* with a specified PURPOSE. The PM layer of the DC node checks the PURPOSE attributes against the Sticky Policy of the PD. Based on the decision of the PM layer, it can either disclose the PD to the DC or not.

At the end of the interaction the *LookupKey*, which serves as a reference for the shared PD, will be known both by the DC and DS. By the means of the mediated space, supported by the Business Ring, both actors share a pointer to the data object, and both can operate on it.

We are employing a distributed logging mechanism, where a request event log is placed in deterministic position inside the Business Ring, and retrieved by the DS using a pull method. As a consequence of our logging mechanism with a deterministic hash function, there is no need to explicitly aggregate logs. Whenever the DS wants to verify the request traces of the shared PD under *LookupKey*, it simply issues a pull request for the *hash(LookupKey)*. Multiple shared PD objects will result on multiple pull requests targeted to different keys. By collecting the logs, the DS can get first hand assurance derived from the traces.

6 Evaluaton on Requirements

6.1 Establishing Trust

Both the Verifiable Privacy (VP) and the Trusted Privacy (TP) models strive to achieve trust by proving that the overall system run by the Data Controller (DC) is trustworthy and secure. These proofs are the result of software verification techniques. Software verification follows the idea, that a verified software system should run and behave according to some predefined requirements. The TP model has an outsourced trust model, that strives to combine multiple sources of trust into a single score. The Mediated Privacy (MP) model, on the other hand, comes with a built in trust measure that is not dependent on any existing TTP. The keyspace slice size is a quantification that can be measured by any peer of the system. Instead of focusing on proving the trustworthiness of some software, the MP is build around the concept of a trustworthy crowd.

Given the different nature of the approaches for trust establishment presented by the three models, it is hard to highlight a single model that provides a higher trust level than the other two. We can conclude, however, that the trust establishment mechanism of the VP is most fitted when it comes to establishing trust between two physical machines. On the other hand, the other two solutions are focusing in providing proof of trust for a DC entity in a more broader sense.

6.2 Transparent User Data Handling

All three privacy protection models presented above are based on the usage of the Sticky Policy paradigm. Thus transparency in this context translates to assurance that the pre-agreed Sticky Policies have been met. The most common way of getting assurance is by verification of logs. Logging is part of all of the three models, but are realized in different ways. Keeping event logs is the responsibility of the Monitor component of the PM of every machine according to the VP and the TP models. The equivalent component in the MP model is the Logging layer present on every node of the Business Ring. The Monitor of the VP model offers the most thorough log keeping solution, since applications are running in their isolated spaces. The middleware approach of the TP model offers a similar monitoring solution, as long as the application layer is not bypassing the middleware. The MP model only offers logging functionalities on Business Ring peers, and is not concerned with the application layer, as the previous models.

In conclusion we can state that the VP and TP models offer a more localized logging solution that focuses on the application layer. On the other hand, the solution in MP is built to support a resilient logging system with log aggregation built in mind. Moreover, the TP and MP both offer first hand log delivery mechanism, which result in a higher assurance level, than the one provided by the log digest in the VP.

6.3 Data Across Multiple Control Domains

Sticky Policies are the main tools that dictate whether data forwarding can take place. A general rule that applies to all of the models is that data forwarded to third parties should have the same or a more restrictive Sticky Policy than the original Protected Data (PD). On the other hand, the data protection across multiple domains also relies on the properties of the dissemination platform on which the data is forwarded. The VP and TP models use the Forwarding Chain as their dissemination platform, while the MP has its own novel solution represented by the Business Ring.

One of the main differences of the two platforms is that, while the Forwarding Chain is a highly dynamical construct with ad-hoc properties, the Business Ring is defined around existing business models. A separate Forwarding Chain is build for every single shared PD object, while there is only a single DHT encapsulating every internal data exchange in case of the Business Ring. In conclusion, the open nature of the Forwarding Chain used by the VP and TP offers a more flexible solution, but lacks the structured property of the MP. By its design, the Business Ring offers a clearer data forwarding model than its counterpart.

6.4 Maintaining Control

Both VP and TP offer the functionality of control over direct data via the Forwarding Chain platform. The DS issues the modified PD to the original data requester, who in turn is responsible to push the modification on through the chain. One of the downsides of this solution is that, since every node hosts his own data copy, every modification operation initiated by a DS triggers a cascade of requests that is flooded over the Forwarding Chain. The MP, on the other hand, assures that the DS will possess a lookup key associated with every shared PD object. If there are two live copies of the same PD under different DCs from within the same ring, the DS maintains a lookup key for each of them. Modification of PD can be achieved via a simple DHT insert that replaces the old version. The Business Ring not only keeps track of all the existing data copies throughout the ring, but also lets the DS modify every PD separately.

7 Conslusion

The goal of this thesis project was to explore different privacy enforcement models employing the Personal Data Vault (PDV) as the source of personal data. As related research suggested, we turned our attention towards privacy enhancing models, that employ the use of privacy policy languages and the sticky policy paradigm. The PrimeLife project, in particular, offered a PrimeLife Policy Language (PPL), which formulates Sticky Policies based on a Data Handling Policy (DHPol) of a Data Controller (DC) combined with the Data Handling Preference (DHPref) of a Data Subject (DS). We proposed three different policy enforcement models: two based on previous research (Verifiable Privacy (VP) and Trusted Privacy (TP)), and one novel approach called Mediated Privacy (MP). The VP provides privacy guarantees through remote software verification methods attested by enhanced hardware solutions. The TP offers a similar design, but a different trust framework, which relies on the combination of independent trust sources in order to provide a quantification of trust. The MP is our novel proposed solution for privacy enforcement, that introduces the concept of a mediated space, which serves as a platform for user data exchange.

REFERENCES

[1] Trusted computing group. URL http://www.trustedcomputinggroup.org/developers/glossary.

[2] Oasis extensible access control markup language (xacml) tc. URL https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[3] Primelife, 2011. URL http://primelife.ercim.eu/.

[4] Christer Andersson, Jan Camenisch, Stephen Crane, Simone Fischer-HÃijbner, Ronald Leenes, Siani Pearson, John SÃűren Pettersson, and Dieter Sommer. Trust in PRIME. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pages 552–559. IEEE, 2005.

[5] K&L Gates Drummond Reed & Joe Johnston, Connect.Me; Scott David. The personal network: A new trust model and business model for personal data. May 2011. URL http://blog.connect.me/whitepaper-the-personal-network/.

[6] Gina Kounga and Liqun Chen. Enforcing sticky policies with tpm and virtualization. In Liqun Chen, Moti Yung, and Liehuang Zhu, editors, *Trusted Systems*, volume 7222 of *Lecture Notes in Computer Science*, pages 32–47. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-32297-6. doi: 10.1007/978-3-642-32298-3_3. URL http://dx.doi.org/10.1007/978-3-642-32298-3_3.

[7] Markus Sabadello. Startup technology report. 2012. doi: http://pde.cc/2012/08/str201201/.

[8] Ravi Sandhu and Xinwen Zhang. Peer-to-peer access control architecture using trusted computing technology. In *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, SACMAT '05, pages 147–158, New York, NY, USA, 2005. ACM. ISBN 1-59593-045-0. doi: 10.1145/1063979.1064005. URL http://doi.acm.org/10.1145/1063979.1064005.

[9] Dave Raggett (W3C) Slim Trabelsi (SAP), Gregory Neven (IBM). Report on design and implementation. PrimeLife, 2011. URL http://primelife.ercim.eu/images/stories/deliverables/d5.3.4-report_on_design_and_implementation-public.pdf.

[10] R. Steinmetz and K. Wehrle. *Peer-to-Peer Systems and Applications*. Lecture Notes in Computer Science / Information Systems and Applications, incl. Internet/Web, and HCI. Springer, 2005. ISBN 9783540291923. URL http://books.google.ee/books?id=A8CLZ1FB4qoC.

[11] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for Internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.