SEVENTH FRAMEWORK PROGRAMME

CloudSpaces

(FP7-ICT-2011-8)

Open Service Platform for the Next Generation of Personal Clouds

D4.1 Guidelines on Privacy-aware data sharing

Due date of deliverable: 30-11-2013
Actual submission date: 12-11-2013

Start date of project: 01-10-2012 Duration: 36 months

Summary of the document

Document Type	Deliverable
Dissemination level	Public
State	Final
Number of pages	21
WP/Task related to this document	WP4
WP/Task responsible	EPFL
Author(s)	Refer to contributors list
Partner(s) Contributing	EPFL
Document ID	CLOUDSPACES_D4.1_131112_Public.pdf
Abstract	Preliminary design of privacy trustworthy data sharing framework and draft API. Initial privacy leakage metrics, initial trustworthiness assessment mechanism, initial privacy enhancing technologies against simple threat models. Share syntactic draft APIs and abstractions including ACLs.
Keywords	privacy-aware sharing, ACL, Personal Clouds

Contributors

Name	Last name	Affiliation	Email
Hamza	Harkous	EPFL	hamza.harkous@epfl.ch
Rameez	Rahman	EPFL	rameez.rahman@epfl.ch
Pedro	García López	URV	pedro.garcia@urv.cat
Marc	Sánchez Artigas	URV	marc.sanchez@urv.cat

Table of Contents

1 Executive summary

The cloud computing paradigm has lured a lot of users to outsource their data to the cloud for obtaining various services. As a result, a significant part of sensitive users' data is being leaked to the cloud service providers. With time, people are becoming increasingly aware of the high privacy risks of exposing their data to these providers. In this report, we propose mechanisms to quantify these privacy risks based on the users' sharing policies. In addition, we discuss solutions to mitigate those risks, and we propose further directions in this domain. In the last part of this report, we present a solution based on attributed based encryption for fine grained access control.

2 Privacy Aware Data Sharing in the Cloud

2.1 Privacy vs. Services Dilemma

To tackle privacy concerns, some cloud computing companies provide the users with the option of client-side encryption to protect the data before it leaves the users' device, thus preventing any other entity from data decryption, including the cloud provider itself. Although this approach eliminates most of the data privacy concerns, it has several shortcomings. The main disadvantage of sending the client data encrypted to the cloud is that the user cannot readily utilize existing cloud services. For some of these services, companies are providing versions of installable software for the users' browsers or operating systems, which is both non-scalable and is against the initial cloud model of delegating the computations to the resourceful cloud server. For other services, attempts exist at designing alternatives that operate over encrypted data, benefiting from the recent breakthroughs in homomorphic encryption [1]. In addition to resulting in services orders of magnitude less efficient than their counterparts, homomorphic encryption is provably not sufficient for constructing several essential services including multiple users, such as collaborative document editing [2]. Furthermore, resorting to homomorphic encryption as the ultimate solution requires rewriting most of the cloud applications' code to operate over the encrypted data. New versions of existing LaTeXcompilers, photo filters, music recommenders, etc. based on homomorphic encryption will need to be programmed with the goal of keeping all data private, which is evidently non-realistic.

2.2 Managing the Trade-off

Currently, the only way to manage this trade-off between maintaining privacy and utilizing services is for the user to manually adjust his privacy settings for each group of data items. Nevertheless, the majority of users are not experienced enough to select the adequate privacy settings, and even experienced users find it cumbersome to specify individual settings for each item they outsource to the cloud. This has been the case in the context of online social networks, where users struggle to maintain such policies [3]. In cloud computing, users are sharing a wider array of information than in online social networks; hence the problem is expected to be exacerbated. Accordingly, the need arises for **privacy aware data sharing solutions** that aid the user in controlling this tradeoff while requiring minimal effort and expertise from his side.

2.3 Challenges in the Personal Cloud

The context of the personal cloud carries several additional challenges for privacy. Other contexts, such as location based services or targeted advertisements, make it simpler to formulate and measure privacy since there is a well defined threat, and the privacy loss can be measured via objective models, independent of users' attitudes. In the personal cloud, users' data is of heterogeneous nature, originating from a wide array of sources. Hence, it is neither possible to have a single privacy measure that can encompass all the different data types, nor is it feasible to enumerate all the possible data types and devise privacy measures for each.

Accordingly, instead of defining privacy risks based on the data itself, an alternative approach would be to seek other inputs to assess that risk. One way would be by directly querying the actual users of the personal cloud about their risk concerns. However, we cannot solely rely on users to declare their privacy preferences, due to (1) the significant user effort required for performing this task for enough number of items and (2) because of the well-known dichotomy between users' reported values of privacy and actual behavior, referred to as the *privacy paradox* [4]. In other words, directly asking users about what they term as privacy sensitive data might not reflect the actual sensitivity as users tend to declare that they highly regard privacy. However, observing users' sharing behaviour shows different results.

Liu and Terzi applied such an alternative approach for measuring the privacy risk of exposing profile items (e.g. age, gender, etc.) in social networks [5]. Their approach has been shown to fit real-world data and has inspired several works later on the same problem(e.g. [6]). They relied on a theory from psychometrics, called Item Response Theory (IRT) in order to model the relationship between the users' sharing behavior and the associated privacy risk. However, their technique's efficiency depended on the assumption that the same profile item is shared by multiple users, which is directly achievable with the limited, static set of profile items in question. This assumption does not hold with general types of data, as in the cloud context. The domain of data to be protected is both open and dynamic, due to the fact that people can share any type of information over prolonged time periods.

Another privacy dimension, which is often missed in the privacy literature, is data semantics. Consider a document D_A, authored by a user Alice who shares it with her friend Bob. This document is considered to be sensitive with respect to Alice. Suppose also that this same document is on the device of another user John, who wants to share it. John would most probably not consider this document as sensitive as it is not his. On the other hand, John would consider another document D_J that he himself authored as being sensitive. Therefore, syntactically similar items need not have the same sensitivity, and syntactically different items need not have different sensitivities. There is a need to not restrict the definition of sensitivity, and consequently the privacy risk, to the syntax of the data, but to extend it to account for the data semantics.

2.4 Contribution

In this work, we consider the privacy problem in the personal cloud from the perspective of an end-user, whose sensitive data needs to be protected and who aims to use the cloud services whenever suitable. We tackle this problem as a *privacy risk management* process, realized in two steps: *risk estimation* and *risk mitigation*. We attempt at solving the former by quantifying the risk of data sharing, via a mechanism that relies on users' sharing policies, works for general data types, and accounts for the sharing semantics. Building on our techniques for quantifying the privacy risks, we aim at designing a suite of applications for mitigating those risks. Our target behind that is to provide users with mechanisms for smoothly managing the trade-off between the privacy risk and the services desired.

2.5 Related Work

In this section, we will be reviewing and discussing related works that seek to tackle the problem of privacy from different angles. Starting from the main challenges in cloud privacy and the proposed remedies highlighted by Pearson[7], we will then move to the work of Liu and Terzi [5] in the context of social networks. The latter work emphasizes the need for treating the problem as a socio-technical one and serves as a basis for one of the schemes we will be proposing for the cloud context. The third work [8], which is very important and in our view shows the way forward for privacy solutions, motivates the need for data semantics consideration in the privacy problem and poses new ramifications of that problem that evolved over the years.

2.5.1 Taking Account of Privacy when Designing Cloud Computing Services [7]

In her work, Pearson describes her vision of the privacy challenges that the cloud computing model introduces, and she suggests design principles to address these challenges. This work is particularly important as it provides a fundamental and comprehensive picture about the privacy problem in cloud computing. It will also serve us later to shed light on the scope of our plans in tackling cloud privacy. **Defining Private Information** The author starts with an enumeration of different types of personal information which are part of what should be protected:

- **personally identifiable information**: that can result in identifying or locating an individual (e.g. credit card number, postal code, etc.);
- **sensitive information**: such as financial data, religious beliefs, health records, etc.;
- **usage data**: such as web viewing habits or product usage history;
- **unique device identifiers**: traceable to a user device (e.g. IP addresses, unique hardware identities, etc.).

Privacy Challenges in Cloud Computing. Although the aforementioned types of personal information are not specific to the cloud context, there are several challenges brought by this model that make this problem unique. The authors mention the following key challenges:

- **Remote data storage and processing:** An organization's data is now being hosted at remote servers, and the main functionalities are executed there.
- **Infrastructure shared between organizations:** With the increased usage of virtualization, the cloud computing companies are providing shared infrastructure among organizations, which raises concerns about the privacy of each organization's data hosted by shared resources.
- **Dynamic environment:** Cloud services might be changed with time. Data might move across organizational boundaries. In order to maintain consistent privacy standards, the organization or users should keep up with such rapid changes and movements.

- **Complex services:** Services might be combined into new ones. For example, a storage service might be combined with a printing service to provide a 'print on demand service.' With these combinations, possibly provided by multiple cloud service providers, concerns arise about the data flow among these providers and about the privacy levels to be maintained.

- **Legal compliance:** Several laws exist governing the collection, processing, and transfer of personal information across geographical boundaries. These laws bring a new dimension of privacy, which is at the country or the continent level.

Privacy Risks for the Various Parties. The implications of the above challenges on privacy are not restricted to a specific entity. Multiple parties are subject to privacy risks, including:

- **Individual users of cloud services:** who might be forced to release personal information against their will;

- **Organizations using cloud services:** which risk leakage of customers' data and subsequent loss of reputation;

- **Implementers of cloud platforms:** who risk the exposure of sensitive information stored on their platform, with the associated legal liability and loss of credibility;

- **Providers of applications on top of cloud platforms:** who might also risk reputation loss, legal non-compliance, etc.

Privacy Aware Design. In order to protect the different parties from privacy leakage, the author presents several recommended guidelines for privacy aware design. We describe these guidelines in the following:

- **Minimizing the personal information sent to the cloud:** Pearson calls for using a suite of client-side mechanisms to minimize the data leakage before outsourcing the data items to the cloud. Such mechanisms range from employing Privacy Enhancing Technologies (PETs) (e.g. generalization or anonymization) to encryption algorithms. The cloud then has the task of using privacy preserving data mining solutions to reason over obfuscated data. One realization of this guideline was another work by Mowrbay and Pearson [9], which targeted the privacy of database tables outsourced to the cloud through client-side obfuscation techniques.

- **Protecting personal information in the cloud:** On the side of cloud providers, security safeguards should be used to prevent unauthorized access, copying, modification, or disclosure of personal information. Data should be stored in an encrypted form, and tamper-resistant hardware might be used during transfer and storage.

- **Maximizing user control:** In general, providing control for users over the flow of personal information strengthens the trust relationship with the cloud computing provider [10]. Even if this control might not be feasible, users should be allowed to state their preferences, that have to be taken into account. An alternative approach would be for the users to select a privacy infomediary, which takes care of guaranteeing their privacy interests.

- **Allowing user choice:** Users should also be given the choice for opting in or out from the additional services the cloud providers offers, such as targeted advertisements. Failing to do so will result in legal non-compliance from the provider's side.
- **Specifying the limit and purpose of data usage:** Data subjects should be informed about a clear purpose of data collection, and any usage of this data should adhere to the users' preferences and the declared intentions from the service. In the presence of dynamic services that change with time, the users' consent should be obtained whenever the usage intentions change significantly. Mechanisms for enforcing these constraints include Digital Rights Management (DRM) techniques and enforceable sticky electronic privacy policies [11].
- **Providing feedback:** Designing human interfaces to clearly indicate the privacy functionalities to the user is part of expanding the reach of a system to the widest community of people. Creating visual hints also enables an informed decision making on the user's side.

Open Issues. Although there is a variety of techniques for privacy enhancement and a clear view of the major challenges, several open issues are yet to be resolved in this area:

- **Policy enforcement:** Although there are laws that govern the usage of data by cloud providers, this might not be sufficient for policy enforcement. Technical solutions should be combined with contractual assurances. The quest for new techniques is for providing stronger level of evidence that the policies are actually being applied.
- **Determining the data processors:** It is not easy for the users or organizations to determine who are the actual data processors in the cloud, especially with the potential involvement of subcontractors.
- **Dealing with the Dynamics of the Services:** Upon the design of a new cloud service, it might be difficult to foresee all the later service evolutions that might occur, which are partly due to coping with user requirements. Hence, a full privacy design might not be feasible in advance.

Discussion. This work serves as a starting point for identifying the privacy challenges in the cloud computing era. Although it does not provide a full solution for each of these challenges, it attempts at providing suggestions for future directions in that field. It is to be noted that the author considers the cloud as a general concept involving all entities which process users' data, involving social networks, internet sites, storage services, etc.

Nevertheless, an important issue which is not touched by the author is **how to discover the sensitive data** in the cloud context. The author lists examples of such data without indicating the complexity of defining whether a certain data item contains personal information. As we have previously hinted, relying on syntactical patterns to find sensitive information is not accurate enough.

Another shortcoming we see is that the author has **varying implicit assumptions about the trust in the cloud provider** in different parts of the text. In other words, sometimes the cloud provider is trusted to carry out the functionalities announced while in other cases it is not. A precise model of the cloud providers' privacy threat would have made the discussion more accurate.

The author **devotes a significant part of the her work for server-side solutions**. However, as we have seen from the recent PRISM program, privacy protection cannot rely mainly on the trust relationship and the laws governing the data processing. Laws might change after the data is shared with the cloud provider; hence, the primary role should be given to protection mechanisms on the user's side.

2.5.2 Subjective Privacy Measure [5]

The work of Liu and Terzi [5] was one of the pioneering works that sought an alternative way by estimating the privacy risk of data disclosure based on people's privacy settings. It focused on the privacy of profile items (e.g. birthday, political affiliation, relationship status ...) in online social networks. Such items are typically assigned certain visibility levels via the privacy controls. Facebook, for example, includes privacy settings that allow specifying the items' audience, such as (private, visible to friends/friends of friends, public, etc.). Liu and Terzi tried to use these policies for deducing how sensitive each profile item is and to infer the privacy risk of sharing it. Towards that purpose, for each user and item, they formulated a risk definition that increases monotonically with the sensitivity of the item and with the visibility this item gets in the network. The *Privacy Score* of a certain user *j* due to all the *n* items he shares is computed as:

$$PR(j) = \sum_{i=1}^{n} PR(i,j) = \sum_{i=1}^{n} \beta_i \times V(i,j) \tag{1}$$

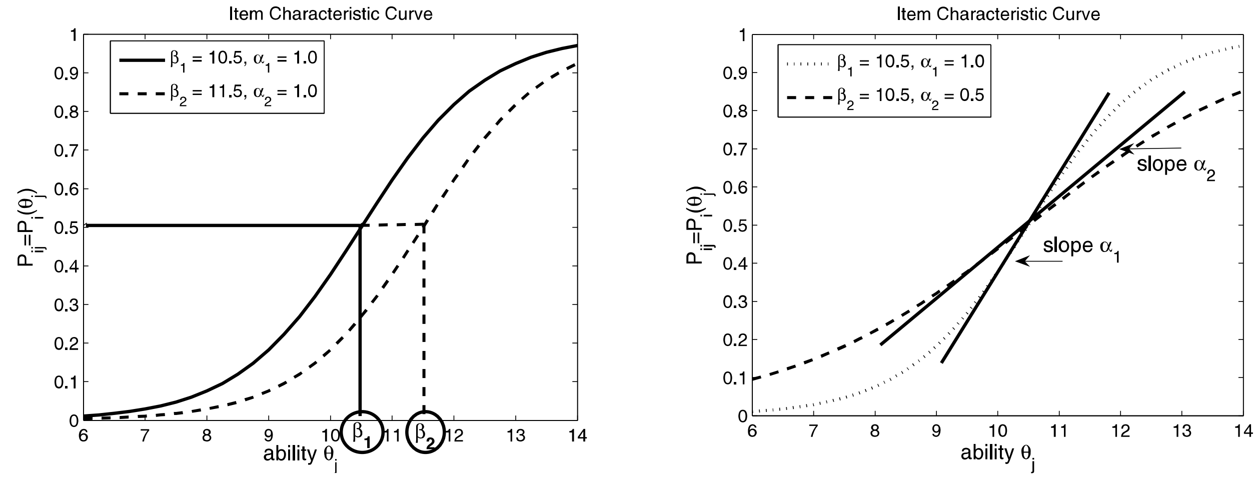
where β_i is the sensitivity of item i while $V(i,j)$ is the visibility of item i as shared by user j.

Visibility is evidently dependent on the level of exposure defined by the privacy settings. However, the main contribution of their work was computing the sensitivity of the profile items. A major challenge towards that target was to avoid the bias resulting from the population of users. Even if all the users in a selected sample are privacy concerned, in principle, it is desired that the sensitivity should not be biased towards large values. Accordingly, the authors employ a theory called *Item Response Theory (IRT)* from psychometrics that mitigates that problem and takes into account the various people's attitudes towards privacy. IRT is a modern test theory typically used for analayzing questionnaires and tests to measure the difficulty of questions (or in general a property β_i of item i), the examinee's abilities to answer questions (or in general a trait θ_j of user j), and the probability of the examinee answering a certain question correctly (or in general a correct response probability P_{ij}). These entities are related by the following equation:

$$P_{ij} = \frac{1}{1 + e^{-\alpha_i(\theta_j - \beta_i)}} \tag{2}$$

An additional item parameter appearing in Equation 2 is the discriminatory power α_i, which indicates how much the response to the item differentiates users with different trait values. Figure 1 shows the *Item Characteristics Curve*, which represents P_{ij} as a function of θ_j.

The authors of [5] applied IRT by mapping the item's difficulty to the sensitivity, the user's trait to the privacy attitude (or willingness to expose the items), and the correct response probability to the likelihood of exposing the item to the public. It is to be noted that



(a) $\beta_1 < \beta_2$.

(b) $\alpha_1 > \alpha_2$.

Figure 1: Item Characteristic Curves (ICC); y-axis: $P_{ij} = P_i(\theta_j)$ for different β values (Fig. 1(a)) and α values (Fig. 1(b)). x-axis: ability level θ_j - (Source: [5]).

the visibility of an item i for a user j is itself the likelihood P_{ij} of exposing i to the public. Taking any reasonable sample of users' profiles, it is easy to obtain a sufficient number of users sharing each profile item. IRT is then used to evaluate an item's sensitivity based on the policies of all those users.

Sensitivity Estimation. The authors start by assuming that the privacy settings are dichotomous, i.e. the privacy setting $R(i,j)$ of user j for item i takes values in $\{0,1\}$. $R(i,j) = 0$ means that the user j has made the profile item i private while $R(i,j) = 1$ implies that this user has exposed that item to the public. This assumption is made to simplify the discussion of parameter estimation techniques, which are later generalized for the polytomous case. In the latter, the privacy setting $R(i,j)$ can take any value $k \geq 1$, signifying that j discloses item i to users at most k links away in the social network.

In order to estimate the sensitivity of a certain item, the authors consider two cases: when the users' attitudes are known or unknown. The sharing policies are always assumed to be given. Assuming the attitudes are known, finding the sensitivity parameter can be reduced to solving a maximum likelihood estimation problem. In particular, the problem is finding the couple (α_i, β_j), such that the following likelihood function is maximized:

$$\prod_{j=1}^{N} P_{ij}^{R(i,j)}(1 - P_{ij})^{(1-R(i,j))} \tag{3}$$

The authors apply a numerical algorithm called Newton-Raphson method [12] to efficiently solve this estimation problem. Assuming the attitudes are unknown, the couple(α_i, β_j) is computed using an Expectation-Maximization procedure.

Discussion. This work by Liu and Terzi showed the applicability of IRT for the field of privacy in social networks. Although the theory has been employed in several domains before [13], this was the first work to study its utility in the privacy domain. Nevertheless, this approach, as it is, cannot be directly generalized to tackle more complex privacy problems, even in social networks. The reason behind that is that the scheme is built on measuring the responses of different people to the same item in order to get the item's sensitivity. Consider for instance the privacy of Facebook posts. It is not very probable to find the same post written by different users. Hence, this approach based on responses to syntactically similar items will fail to generalize due to the described **data sparsity problem**. Moreover, applying this approach to other problems while being tied to a syntactic item definition is also problematic, as we described in Section 2.3. Furthermore, the authors considered a static scenario, where all the items are shared by all the users, and the sharing policies are known. In most sharing scenarios, items are shared over time. Hence, if we need to use these sensitivity values before all the items are shared by all users, we have to seek a different approach. Accordingly, even though our approach is inspired by this technique, tailoring it to the personal cloud context needs solving significant problems, which we will be tackling in the rest of this work.

2.5.3 Privacy and Online Social Networks - Can Colorless Green Ideas Sleep Furiously [8]

This recent work by Krishnamurthy highlights major limitations found in most of the existing privacy mitigation solutions. Although the author puts these problems in the context of online social networks (OSNs), they are by no means limited to this context. As we explain later, we aim at tackling several of these limitations in the cloud context. Hence, we consider this work as paving the way for our later works.

Emerging Privacy Problems. According to the author, the core of the privacy problem in social networks lies in two dimensions: **data leakage** and **data linkage**.

So far the main focus in the literature was on the leakage problem, which can be traced along multiple axes: **across time**, **passive leakage via regular OSNs**, and **inference through active mining**. Studies have shown that the sheer number of privacy settings and the complexity of tracking them over time was a major reason for the leakage of users' information when interacting with the OSNs [14]. On the other hand, active mining in OSNs might result in inferring users' private profiles [15], benefiting from the tendency of humans to have affinity for people with likes similar to their own.

Nevertheless, the leakage of information is nowadays exacerbated in the presence of content aggregators. In social networks, these aggregators are typically associated with ad providers who aim at linking personal information and creating complete profiles of users, along with their browsing habits and shopping preferences. The issues that arise due to the ability of linking information can be summarized as follows:

- **Aggregation from multiple sources:** Information gathered from a certain social network can be linked with that obtained from another social network. Furthermore, such data can be linked with offline data, such as that collected from the users' shopping history available from supermarkets' records.
- **Aggregation over time:** Users sharing a piece of information on a social network might not be aware that linking it with previously shared data might lead to leaking sensitive data.

- **Secondary Use:** Users are usually aware of the first hop receiving their shared data. However, with the potential of leaking/relaying this data over multiple hops, the risks are certainly exacerbated.

Classification of Existing Solutions. The author argues that the popular privacy protection proposals are lacking since they are syntactic in nature. Before motivating the need for semantics, he provides a review of the existing solutions, which we summarize into two types:

- **Mitigating the problem within the current architectures:** Some of the existing solutions are working on designing mechanisms to limit the release of personal data within the existing model of OSNs. Several approaches are taken towards that. Some attempts, such as PrivacyBucket, presents detailed information about expected leakage to the users. Other approaches provide visualizations (e.g. WebCrumbs, Privacy Dashboard) or logging tools (Fourth Party). Furthermore, browser extensions such as AdBlock and Ghostery can prevent connections to aggregators while NoScript can prevent executing suspected JavaScript. Masking user's personal information is another approach that falls in this category too. For example, some privacy preserving location based services attempt at sending queries about multiple locations and locally reconstruct the answer for the current location, thus avoiding revealing the latter to the server. Nevertheless, as the author shows later, these approaches are merely discrete steps without a comprehensive privacy solution.
- **Designing new architectures that avoid the problem:** Architectural approaches bypass the syntax vs. semantics dilemma and address the privacy problem from a comprehensive point of view. For example, Vis-a-vis [16] and Safebook [17] projects attempt at partitioning the data over the set of peers participating in the network; thus the need for a third party and advertisements disappears. However, such approaches suffer from two main concerns: guaranteeing availability of users' data at all times and scalability to large numbers of users.

The Need for Semantics. After examining the leakage and linkage threats, the author shows how syntax-oriented solutions, based on pattern detection and blacklisting do not address the problem in its entirety. The main argument is that the syntactic approaches rarely go beyond the first hop of communications while understanding the flow of data over multiple hops for a longer period of time requires a semantic based approach. The authors proceed by providing counterexamples to show the shortcomings of the syntactical techniques. The key counterexample shows a typical sequence of events involving multiple interacting parties: a user visits a popular website (www.AGEGROUPS.site), which triggers the fetching of http://metrics.AGEGROUPS.site/:

```
GET http://metrics.AGEGROUPS.site/...
Referer: http://www.AGEGROUPS.site/
Cookie: ...e=jdoe@email.com&f=John&l=Doe&...
```

The new URL appears to be related to the visited website, based on the second-level domain name. However, examining the authoritative DNS server of http://metrics.AGEGROUPS.site reveals that it belongs to a popular aggregator site. Moreover, according to the HTTP protocol, the cookie associated with www.AGEGROUPS.site will be sent to the other site, as the second level domains match. The cookie contains the name and the email address of the user which will be leaked to the aggregator site, who might possibly perform the linkage with previous information from OSNs. This highlights that simple privacy protection schemes that block third party cookies fail to capture the semantic complexity of the cookies behaviour and the flow of data across sites.

Towards Semantics Based Solutions. After motivating the need for semantics, the author presents his attempt at bridging the gap between perception of privacy settings and the reality. This work tries to highlight the need for capturing dynamics of the data spread and the leakage possibilities. It comes in the form of a Facebook extension called Privacy IQ, which presents the user with different questions related to posts or photos he shared on the social network. Examples of those questions include: "can a certain user (non-friend) tag you in this photo?", "who can see the list of a user's friend?", etc. Privacy IQ allows the users to see their past privacy settings and to observe the connections that can be made due to them. It ultimately serves as an educational tool raising users' awareness about the access rights to their objects, the reach of their social graph and the privacy implications of applications they have installed.

Studying the responses of 200 people who used this application, it was observed that the users did not expect that pictures posted in the past had a much wider permitted audience. Also, there was an absence of clarity on the reach of their data (data meant to be viewable by friends was open to broader range of users). In addition, a lot of fine-grained privacy settings were available but not used (e.g. pictures in albums can have their own privacy settings).

Discussion. This vision paper by Krishnamurthy emphasizes the need for taking the social networks privacy solutions to the next level, where not only the first level of audience is considered, but the whole flow of users' information is analysed to guard against potential hidden parties. Nevertheless, despite arguing about the need for improving the current syntactical solutions, the author **does not suggest how semantic tools can be used in order to remedy the current approaches**. Moreover, the issue of **discovering in advance the undeclared parties who might have access to the data** is not fully investigated. Several challenges might occur in that regard. He also **does not suggest alternatives that can have a high penetration rate in the current online social networks**. Despite implementing Privacy IQ, this service can only serve as an educational tool that only convinces its users about the need for being aware about the leakage of their data over time and the hidden parties who might have access to such data. The author also restricts his discussion of privacy semantics to entities interacting in the social networks. However, he **neglects the semantics of the data itself**, starting from the data contents to the metadata, which can play a deciding role in any privacy aware sharing system.

2.6 Privacy Risk Estimation

2.6.1 Overview

The problem of risk quantification has been tackled previously in software engineering. The definition given by Liu and Terzi in Equation 1 matches previous proposals for risk estimation, such as that of Charette [18] and that of Hall [19]. All these works revolve around estimating two elements: the probability of potential loss (i.e. the risk probability) and the consequences or magnitude of the identified risk [20]. In [5], visibility represents the risk probability while the sensitivity represents the risk magnitude. In the personal cloud context, we will follow the general risk definition, and quantify the privacy risk of sharing an item i by a user j as the product of the sensitivity β_i of item i with the disclosure probability $P_D(i,j,k)$ of i as shared by user j with a data observer k:

$$PR(i,j) = \beta_i \times P_D(i,j,k) \tag{4}$$

At this level, we consider generic items, and we leave the definition of what these items represent and their granularity level to Section 2.6.4. Consequently, by the notation β_i we do not mean that the item's sensitivity is independent of the user j, entity k, or other factors.

2.6.2 Probability of Disclosure

We build our quantification of the disclosure probability on the intuition that it depends on (1) the level of trust $T_{j,k}$ given to the data observer to not misuse the data in violating the sharer's privacy and on (2) the protection level PL_i of the item i against disclosure. This protection level ranges from zero, for the case of no protection to PL_{max} for the highest protection level. This protection can be realized via various privacy enhancing technologies (PETs) and cryptographic tools. At this stage, we deal with these levels generically, and we assume that there is a predefined mapping from each such technique to a specific protection level in the given range. Accordingly, we propose to quantify the probability of disclosure as follows:

$$P_D(i,j,k) = T_{j,k} \times PL_i \tag{5}$$

2.6.3 Sensitivity

We will be utilizing Item Response Theory by doing the same mapping as Liu and Terzi did (cf. Section 2.5). However, our definition of the disclosure probability is the one given above. In addition, we consider a dynamic system, where items are not shared all at once as in [5]. Hence, we propose a mechanism that allows computing the sensitivity values during the progress of the system.

The users starts by triggering a sharing operation, where the probability of disclosure is known (i.e. via trust and protection level). Our mechanism alternates between the three following steps, depending on the available data, until it determines the sensitivity of each shared item:

- **Bootstrap** Assume there is a set of items of unknown sensitivities, each of which is shared with enough number of users of unknown attitudes. The bootstrap phase consists of estimating these attitudes and sensitivities via IRT through the joint parameter estimation method, as described in [21].
- **Sensitivity Estimation** Assume there is an enough number of users with known attitudes who shared the same item *i*, whose sensitivity is unknown. IRT allows calculating the sensitivity of *i*, as explained in Section 2.5 via solving a maximum likelihood estimation problem.
- **Attitude Estimation** Assume there is an enough number of items of known sensitivity, which are shared by a certain user *j* whose attitude is unknown. IRT allows calculating the attitude of *j*, as via solving a maximum likelihood estimation problem too [21], similar to the way sensitivity is calculated.

2.6.4 Item Definition

Up till now, we considered generic items. In order to avoid the sparsity problem while being of utility to IRT, the item should be defined so that multiple users typically share the same item. Moreover, the item representation should account for the sharing semantics too due to the inadequacy of the syntactic one on its own, as explained in the introduction. Hence, to solve the above two issues, we introduce the *semantic item definition* concept.

This definition depends on some ontologies (also known as controlled vocabularies) describing the people, entities, types of items being shared, sharing context, etc. They can be built and customized based on existing ontologies, such as *Friend of a Friend Ontology (FOAF)* [1] or *NEPOMUK File Ontology (NFO)* [2].

A semantic item definition consists of a set of instances of the concepts in the ontologies. For example, one such definition can be:

```
ItemType is Photo and Tag is {me}
and Observer is {X}
and X is Friend
```

Accordingly, each item in the previous sections is represented by its semantic definition. In order for the scheme in Section 2.6.3 to work, similar items shared by multiple users must be found. This similarity computation has to be done in the cloud at the intermediary CSP.

2.6.5 Private Similarity Computation

Despite the fact that a semantic item representation allows mitigating essential problems, sending it to the CSP introduces a new privacy threat: now the items' metadata, the context of the users, in addition to several other pieces of personal information are being leaked to

[1] www.foaf-project.org

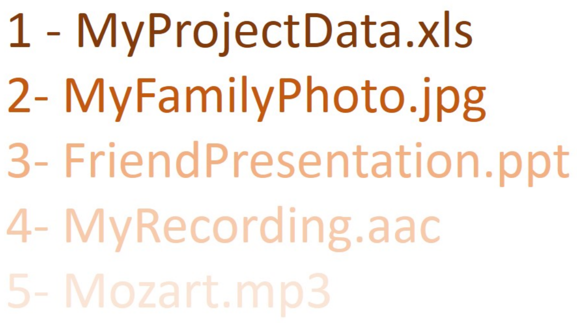
[2] http://oscaf.sourceforge.net/nfo.html

the cloud. To solve this problem, we are developing privacy preserving similarity computations techniques, which consist of anonymizing the context before it is sent to the CSP while still allowing the similarity computation.

2.7 Privacy Risk Mitigation

Building on our techniques for quantifying the privacy risks, we aim at designing a suite of applications for mitigating those risks. Our target behind that is to provide users with mechanisms for smoothly managing the tradeoff between the privacy risk and the services desired.

The first example of the tools we are planning to implement is called the **Risk Rank** tool. This tool, visualized in Figure 2 allows the user to visualize the files exposed to the highest privacy risk on his device via a search-engine-like interface. The second example of tools is what we call the **Risk Meter** tool. This tool, appearing in Figure 3 allows the user to visualize the tradeoff between the attainable services and the privacy risk. In specific, it provides the user with a slider-like interface that ranges from low risk to high risk and gives the user a visual hint on what can be achieved at each risk level.



1 - MyProjectData.xls
2- MyFamilyPhoto.jpg
3- FriendPresentation.ppt
4- MyRecording.aac
5- Mozart.mp3

Figure 2: Risk Rank

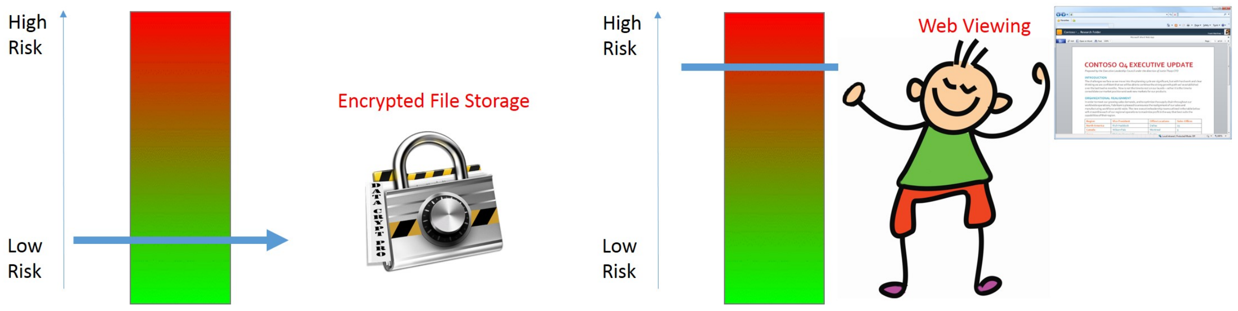


Figure 3: Risk Meter

The third example of tools we plan to realize is a policy recommendation tool that learns from users' privacy settings to aid similar users to adjust their settings. At the core of this recommendation process is the semantic definition of the data and the relationships between users.

2.8 Other Pieces of the Puzzle

2.8.1 Data Linkage in the Cloud

Similarly to the case in social networks, cloud providers can also act as data aggregators. Thus there is a need for enabling the user of preventing the risky linkage of their sensitive data. Our vision for solving that problem consists of designing user-side aggregators, that exploit the semantics of the data in order to analyze the current sharing decision with the previous and possibly the future sharing decisions in order to determine the privacy risk of sharing. We also aim at considering different models of aggregators, where we can have various cooperation assumptions among the cloud providers and study the impact of each such model on the privacy risk.

2.8.2 Temporal Access Patterns and User Privacy

A novel research direction we envision in the cloud context is studying the relationship between the temporal patterns of accessing cloud services and the sensitive information that can be discovered from observing those patterns. We first note that it is usual for the cloud providers to log events such as the start time of syncing data shared on a certain device, of updating a certain item on the cloud, etc. The associated privacy risk in this case is twofold. First, the cloud provider can learn information about the life schedule of a certain user, such as when he opens the laptop (possibly correlated with when he wakes up), when he is on the go (i.e. when he uses his mobile device for data access without having the laptop on sync), etc. Profiling users' life can be made possible with such simple logged data. Second, even with solutions that encrypt the data on the client side, the temporal patterns of editing this data can leak some information, such as the types of the files being edited. Obviously, one edits word documents more frequently than he edits his photos. It might also leak information about the social or business relationships between users. For example, users who frequently edit excel sheets with each other are likely to be colleagues. Protecting the private information leaked due to exposing such temporal patterns is a major and novel research direction we will be pursuing.

2.8.3 Data Retention Problem

Another dimension of privacy is the duration for which the data should be allowed to remain at the cloud provider. This problem, commonly referred to as the *data retention problem*, is currently tackled from the policy and legal-compliance perspectives [7]. However, we are willing to tackle it from the user's perspective. The user would prefer to withdraw his data from the cloud for various reasons, including the privacy concerns and the quota management. The longer the data is kept at the cloud provider, the higher the risk that this data would be leaked or abused. Hence, we aim at designing mechanisms that recommend a time for the user to withdraw his data from the cloud, assuming cooperation from the cloud provider's side. In addition to the privacy risk and the quota factors, we will be utilizing the temporal patterns of access to the data to recommend the most suitable time for withdrawing the data from the cloud.

3 Syntactic APIs and ACLs

The syntactic APIs already specified in D2.2 and D5.1 include explicit support for coarse grained access control. This means that is is possible to share entire folders with either read or read/write permissions. This initial API does not take into account the use of encryption for protecting the content. In this line, if a shared secret key is used to protect shared content, the involved Personal Clouds should be responsible for managing the group key. In this case, if a member is removed from the group, the group key should change, members should receive a more recent key, and the content must be re-encrypted (lazily or proactively on all the content). It is beyond the scope of the syntactic APIs to provide support for management of group keys and encryption technologies .This would complicate the API precluding its adoption.

In any case, we consider that fine-grained access control and encryption is required in a variety of high security settings that require collaborative or group work. This is especially visible when the risk rank tool suggests that a given object is sensitive and the disclosure probability is high. For this reason, we analyse here the potential use of Attribute Based Encryption for providing privacy-aware fine-grained access control in Personal Clouds. We propose an initial candidate solution for this specific problem.

In our proposed solution, the security provided by the transmission channel could not be enough: the storage server itself could have vulnerabilities that lead to the exposure of our private data to the rest of the users of the system or, in the worst case, to the rest of the world. Some of the actual solutions to this problem is to apply cryptographic techniques in order to cipher the data that will be stored on the server and only allow the authorised users access to that data, giving them the corresponding encryption keys. That represents a high work overhead to the data owner, that has to distribute and manage the keys, reducing scalability and only allowing coarse-grained access control, problems that we want to avoid in our system.

We understand as fine-grained access control the system ability to guarantee different access privileges to a set of users and also provide the flexibility of specify the access privileges only to individual users. The most common techniques that guarantees fine-grained access control are based on using a trusted server that stores the data in plain text. In these cases, access control is done by software routines that checks if the users accessing to the data are are allowed or not. This represents a huge danger if the server is exposed using some exploit and this check is evaded, or if the data is leaked by some entity from the system itself. These techniques leads to a set of limitations, such the limited scalability and coarse-grained access control, that makes its application non-viable for the data sharing context.

One of the most adequate solution to these challenges was proposed by Sahai and Waters, calling their method Attribute Based Encryption (ABE) , which implementation is proposed in [22]. This scheme consists on associating a set of Access Policies (defined by a set of attributes and boolean expressions involving them) to the system users, and a set of attributes to the data files, so that the users will be able to access to the data if and only if the data file attributes fits the defined user access policy. In Figure 4 we can see an example of an ABE access policy that is fulfilled with the user private attributes.

In our proposed solution, the data files will be stored on the server in an encrypted way, where different users will be able to access depending to their access policy implicit in the

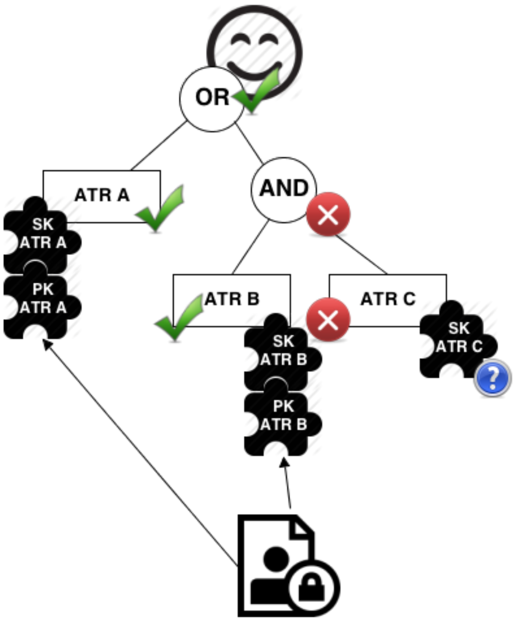


Figure 4: Example of an ABE access policy.

data and the user keys, removing the need for the server to check data access using software routines, avoiding the danger of exposing data in the case that the server is compromised.

We have observed that in most cases it is common the association of a set of meaningful attributes to the data files or users in the system. We can use that attributes to create also meaningful policies for the data owner and the system users.

But in this scheme, when we have to revoke a user, a problem appears: the additional cost of create and redistribute the keys to each of the allowed users, the reencryption of all the affected files, and the mandatory availability of the data owner greatly reduces the system performance and scalability. The solution to this issue consists of delegating all this work to the proxy server and do all the job on-demand, i.e. , when the system users requests some file, the proxy checks if the user is allowed, then checks the user secret key and updates it if it's necessary, and finally checks if the data file public key components are updated and does the corresponding in the case that they are not. Through this scheme, the user is acquitted of all the hard work, improving the efficiency and scalability of the system.

Our proposed implementation life cycle is the following:

1. The data owner will define an universe of attributes meaningful in the sharing context creating an Attribute Based Encryption (ABE) public key component for each attribute. Also, has to add the users to the system (adding to the server's User List), defining the access policy for each of them and create the respective ABE secret keys for each user.

2. The data owner will encrypt the data file using a symmetric encryption key, e.g. an AES key, and will encrypt this key using ABE, associating to the encrypted data file the set of attributes that satisfies the policy of the users with whom we want to share the data. Then, the file will be uploaded to the data server where it will be stored.

3. When a user wants to download a file, he first makes a request to the data server. First, the data server will check if the user is in the User List. If he is, the data server responds to the query. When the user has downloaded the file, he has to decrypt the symmetric key using his ABE secret key. If the attributes of the data file satisfies the user's policy tree, then the user can recover the symmetric key and decrypt the data file content.

As Personal Clouds include functionality on both server and client sides, ABE-based access control for privacy-aware data sharing must be implemented as a security layer over the existing system implementations.

On the client, some new functionalities are required in order to allow user management and policy association to them. Also, the data owner client program needs to be able to manage and store the system keys and the attribute universe. The encryption will be applied in the upload and download system methods, so the systems should offer an API that will be easily embedded in the existing code, minimizing any changes in the current workflow.

As observed in Figure 5, the proxy or server-side (CloudABEProxy) requires a database for store the following data: (i) User List, (ii) File attributes and (iii) an attribute history list for the key update in case of user revocation. Also, the files will be stored with a header containing the encrypted symmetric key, and the set of associated attributes, so the system has to be adapted to this new data storage format.

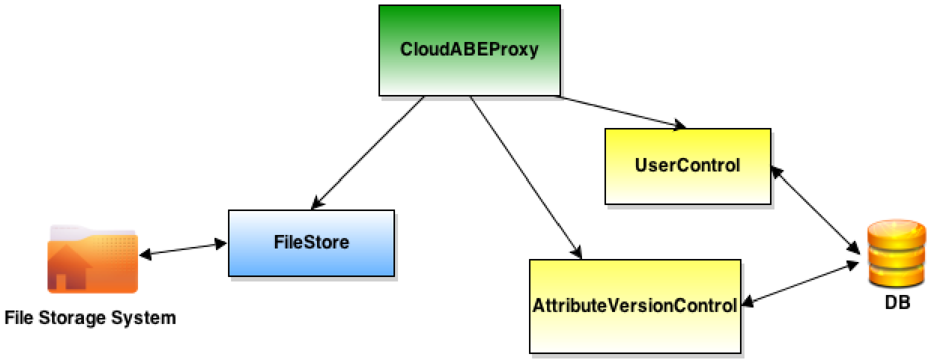


Figure 5: ABE architecture.

To conclude, we outline that a prototype example of this fine-grained access control solution is being developed in the context of the CloudSpaces project.

References

[1] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[2] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in Proceedings of the 5th USENIX conference on Hot topics in security, 2010, p. 1–8.

[3] K. Strater and H. R. Lipford, "Strategies and struggles with privacy in an online social networking community," in Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1, ser. BCS-HCI '08. Swinton, UK, UK: British Computer Society, 2008, p. 111–119. [Online]. Available: http://dl.acm.org/citation.cfm?id=1531514.1531530

[4] S. B. Barnes, "A privacy paradox: Social networking in the united states," First Monday, vol. 11, no. 9, Sep. 2006. [Online]. Available: http://firstmonday.org/ojs/index.php/fm/article/view/1394

[5] K. Liu and E. Terzi, "A framework for computing the privacy scores of users in online social networks," in Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on, 2009, p. 288–297.

[6] T. Munemasa and M. Iwaihara, "Trend analysis and recommendation of users' privacy settings on social networking services," in Proceedings of the Third international conference on Social informatics, ser. SocInfo'11. Berlin, Heidelberg: Springer-Verlag, 2011, p. 184–197. [Online]. Available: http://dl.acm.org/citation.cfm?id=2050728.2050759

[7] S. Pearson, "Taking account of privacy when designing cloud computing services," in Software Engineering Challenges of Cloud Computing, 2009. CLOUD'09. ICSE Workshop on. IEEE, 2009, pp. 44–52.

[8] B. Krishnamurthy, "Privacy and online social networks: Can colorless green ideas sleep furiously?" IEEE Security & Privacy, vol. 11, no. 3, pp. 14–20, 2013.

[9] M. Mowbray and S. Pearson, "A client-based privacy manager for cloud computing," in Proceedings of the fourth international ICST conference on COMmunication system softWAre and middlewaRE. ACM, 2009, p. 5.

[10] A. Tweney and S. Crane, "Trustguide2: An exploration of privacy preferences in an online world," Expanding the Knowledge Economy: Issues, Applications, Case Studies, 2007.

[11] M. C. Mont, S. Pearson, and P. Bramhall, "Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services," in Database and Expert Systems Applications, 2003. Proceedings. 14th International Workshop on. IEEE, 2003, pp. 377–382.

[12] T. J. Ypma, "Historical development of the newton-raphson method," SIAM review, vol. 37, no. 4, pp. 531–551, 1995.

[13] R. K. Hambleton and H. Swaminathan, Item response theory: Principles and applications. Springer, 1984, vol. 7.

[14] K. Strater and H. R. Lipford, "Strategies and struggles with privacy in an online social networking community," in Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1. British Computer Society, 2008, pp. 111–119.

[15] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: inferring user profiles in online social networks," in Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010, pp. 251–260.

[16] A. Shakimov, H. Lim, R. Cáceres, L. P. Cox, K. Li, D. Liu, and A. Varshavsky, "Vis-a-vis: Privacy-preserving online social networking via virtual individual servers," in Communication Systems and Networks (COMSNETS), 2011 Third International Conference on. IEEE, 2011, pp. 1–10.

[17] L. A. Cutillo, R. Molva, and T. Strufe, "Safebook: A privacy-preserving online social network leveraging on real-life trust," Communications Magazine, IEEE, vol. 47, no. 12, pp. 94–101, 2009.

[18] R. N. Charette, Software Engineering Risk Analysis and Management. Mcgraw-Hill (Tx), Feb. 1989.

[19] E. M. Hall, Managing Risk: Methods for Software Systems Development, 1st ed. Addison-Wesley Professional, Feb. 1998.

[20] S. Coyle and K. Conboy, "A case study of risk management in agile systems development," 2009. [Online]. Available: http://vmserver14.nuigalway.ie/xmlui/handle/10379/1392

[21] F. B. Baker, The basics of item response theory. ERIC, 2001.

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of the 13th ACM conference on Computer and communications security. ACM, 2006, pp. 89–98.