2013 IEEE 10th International Conference on e-Business Engineering

Trading Privacy in the Cloud: A Fairer Way to Share Private Information

Constantinos Patsakis*, Agusti Solanas†
* School of Computer Science and Statistics, Trinity College, Dublin, Ireland patsakik@scss.tcd.ie
† Dept. Computer Engineering and Maths, Universitat Rovira i Virgili, Catalonia. (Spain) agusti.solanas@urv.cat

***Abstract*—Millions of people use cloud services for remote storage, events management etc. By using these services, users share and synchronise their information amongst multiple devices seamlessly. Nevertheless, the *freemium* model that most of them follow, endangers and in many cases invades the privacy of the users. With the aim to avoid privacy issues, which limit social progress, in this article we propose a new privacy-aware protocol and architecture allowing a new and fairer business model based on the concept of *Privacy as a Product (PaaP)*.**

Our proposal is based on well-known cryptography such as re-encryption, Bloom filters and the Raykova-Vo-Bellowin-Malkin scheme. With our proposal, users gain full control over their privacy and they can allow advertising companies to mine their information. Cloud service providers act as brokers and get revenues from each transaction. Thus, ours is a win-win model overcoming the limitations derived from the privacy invasion of current models.

***Index Terms*—e-Commerce Privacy, Fair business models, Privacy trading and social development.**

I. Introduction

We are witnessing the development of a new society that benefits from ubiquitous computing technologies and is more connected than ever. This can be justified by the wide spread of mobile devices, along with the wide acceptance and dependability of users from the Internet. However, due to the fact that people use more than a single device, the synchronisation of the information in each device becomes a need. Also, due to the different computation and storage capabilities of these devices (eg. a PC is far more powerful than a mobile phone), there is a need for novel approaches that allow thin terminals to synchronise and share information with much more powerful devices. To provide a solution to this problem (and some others) the Cloud Computing paradigm was born. Taking advantage of the Cloud, many companies are offering a wide range of services. Well-known examples are Google with Google Drive, Apple with iCloud, Microsoft's Skydrive, Dropbox, box, Ubuntu One, and the list goes on.

However, such services are generally provided under service conditions that might endanger the privacy of users. See for example the following conditions from Google [1]:

"When you upload or otherwise submit content to our Services, you give Google (and those we work with) a worldwide licence to use, host, store, reproduce, modify, ... This licence continues even if you stop using our Services."

It is apparent from the above excerpt that the privacy of users could be exposed. Note that, users might decide to trust cloud service providers (*i.e.* they believe that cloud service providers will behave properly and will use their data as they say). However, trust in cloud service providers (CSPs) is significantly affected when critical vulnerabilities such as [7] are revealed. So, even if the providers behave properly, there are strong reasons to believe that they can be attacked and some or all the data from their clients could be disclosed.

To protect the privacy of cloud users, several software solutions have been proposed, such as SocialFortress, Boxcryptor, and SecureCloud[2]. The functionality that most of these solutions provide is founded on the encryption of the data before their upload. The data are encrypted with a cryptographic key provided by the user, who is the only one that can retrieve and decrypt the data. Even if these solutions provide security with regard to the CSP (i.e. because he cannot decrypt the content), they face two major problems in terms of usability and performance. Firstly, users cannot query their data efficiently, as the provider doesn't have access to the key, thus the user has to download all the encrypted files, decrypt them locally and search each one of them to find the needed information. Not only this task is very demanding in terms of computational cost, but also in terms of bandwidth.

Secondly, they face problems with user privilege revocation. In many cases, users want to share some data with other users as well. To do so, a naive solution is to share the encryption/decryption key amongst all the users. Let us assume that the user (owner of a given file) wants to revoke access from user Malory. Although the user could ban Malory, she can decrypt the content if she intercepts the traffic to other users. To circumvent this problem, the owner of the files has to download the files, decrypt them with the old key, encrypt them with a new one and send it to the rest of the users. Obviously, this approach falls back again to the drawbacks described previously.

There exist intermediary-based solutions such as CipherCloud or PerspecSys[3], that aim at shifting the trust from the cloud provider and pass it to them. However, there is no apparent reason to trust those intermediate entities more than the CSP. Thus, the problem remains.

Regardless of the aforementioned shortcomings, we strongly believe that data encryption with personal keys is the most

[1] Google Inc. Google terms of service. http://www.google.co.uk/intl/en/policies/terms/regional.html, 9/9/2012.

[2] https://www.socialfortress.com/, https://www.boxcryptor.com/, http://www.trendmicro.com/us/enterprise/cloud-solutions/secure-cloud/index.html

[3] www.ciphercloud.com/, www.perspecsys.com/

978-0-7695-5111-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ICEBE.2013.63

IEEE computer society

promissing way to really guarantee their control (especially if we consider that governments may force service providers to reveal data from their users). Thus, the only way to avoid (authorised or unauthorised) leakage of information from the cloud is to keep the data encrypted.

We understand that by encrypting the data stored in the cloud, we are cutting a source of income for the CSP, and we are making queries and management much more difficult. Therefore, we introduce an alternative business scheme that can provide the user with the needed privacy and simultaneously, it allows controlled queries of the CSP and/or advertising companies.

A. Contribution and organisation of the article

In this article we propose a new protocol and architecture that allows the creation of a new business model in the cloud based on the concept of "Privacy as a Product" (PaaP). We claim that privacy has become a product with which CSPs make money (by selling it to advertisement companies) regardless of the users. The protocol and architecture proposed in this article allow a fairer business model in which users control their privacy whilst advertising companies can mine their data and CSPs can get benefit from their transactions. Therefore, this article aims to return the control information to users whilst maintaining a viable, privacy-aware business model profitable to all involved parties.

The rest of the article is organised as follows: In § II we provide the reader with some basic concepts on cloud computing (§ II-A), privacy (§ II-B), re-encryption (§ II-C) and we recall a basic building block for our architecture: the Raykova-Vo-Bellovin-Malkin protocol (§ II-D). Next, in § III we describe our proposal. In § IV we discuss the feasibility of our proposal from the technical and economical perspectives. Finally, the article concludes with some final comments and remarks in § V.

II. Background

A. Foundations of cloud computing

Although the concept of cloud computing is relatively new, the underlying ideas related to the efficient use of computational resources, the sharing of expensive equipment and the cost reduction due to economies of scale were already present in the 1950s, when costly and heavy mainframes were accessed by terminal computers. Clearly, the infrastructure (i.e. networks, servers, data centres, etc.) holding the paradigm of cloud computing have largely changed from those old times, but they serve the same main principle of time sharing.

Depending on the ownership and management of the cloud we can distinguish three main cloud types: (i) The **public cloud**, where the infrastructure is controlled and managed by third parties that offer services to clients through the Internet on a free or a pay-per-use model; (ii) The **private cloud**, where the infrastructure is controlled and managed by a single organisation which, at the same time, is the user. A special case of private cloud is the **community cloud** in which the cloud infrastructure is controlled by a number of organisations instead of a single one; and (iii) The **hybrid cloud**, which is a combination of any of the previous approaches, namely public, private or community cloud.

The new paradigm of cloud computing has very interesting properties, namely flexibility, scalability, reliability, amongst others. All these interesting properties are more or less important to the users depending on the cloud model they use. Essentially we distinguish three main cloud models[4] as follows: (i) **Infrastructure as a Service (IaaS)**, which is the lowest level service possible. In fact, it can be understood as the renting of equipments located in the facilities of a third party. Providers offer hardware infrastructure, or more generally virtual machines and users are billed in a per-use basis (i.e. providers bill users according to the amount of resources allocated and consumed); (ii) **Platform as a Service (PaaS)**, which is a higher level of service compared to the IaaS, in which, users are provided with a ready-to-use platform that in general includes an operating system with some programming languages execution environments, databases, web servers, mail servers, etc; (iii) **Software as a Service (SaaS)**, which is the highest level of service provided by the cloud. Users do not have access to any lower level of control. On the contrary, cloud providers manage applications software and users simply access it by means of thin clients (e.g. web browsers). Typically, this service is charged in a monthly or yearly flat rate. Also, there are many free services, in which users do not pay but, in return, they allow the cloud providers to access their data and mine it so as to extract information, show them advertisements and the like.

B. Privacy in the cloud

We may state that privacy is a **need** that gained the rank of **right** and has finally become a **service**. However, due to the generalisation of *freemium* (free + premium) models used by many SaaS providers, privacy is now seen as a "**Product**" or a "Good". In the context of social networks, which use also freemium models, it was shown that privacy could be understood as a product from which social networks providers benefit [11].

With the wide acceptance of SaaS, many services are provided for free in exchange for the unlimited access to users data. By using these services, our calendar events, our documents, our e-mails, etc. are scanned, analysed and mined by cloud providers. As a result, they obtain information about us that can be used to show us advertisements to which we are more willing to click on. Thus, we might say that the service is not really free. Instead, we might state that users are paying with their privacy, which might have a significant value. Thus, it becomes apparent that there is a need for a fairer business model that allows users to control the data/information that they are willing to share with service providers in order to "pay" for the services that they use. As stated by Eric Houghes in [9] "*Privacy is the power to selectively reveal oneself to the world*". In this article, we propose a private architecture

[4]A number of cloud service models can be found in the literature, namely Storage as a service (STaaS), Security as a service (SECaaS), Data as a service (DaaS), Database as a service (DBaaS), etc. But they can be classified under the three categories described above.

that allows the sharing of private information in a secure and controlled manner so as to guarantee a fairer way to trade with private information.

C. Re-encryption

The idea of re-encryption is an extension of public key encryption. In public key cryptography, we create a set of keys, the public and the private, and they are used for encrypting and decrypting the messages respectively. So if Alice wants to talk with Bob, Alice finds Bob's public key and encrypts her message with that key. Bob using his private key is the only one who can decrypt it.

We introduce now another entity in the game, Carrie, which carries the messages from Alice to Bob. Carrie's task is to carry a message, without being able to access it, while transforming it according to Alice's requests, so that only Bob can decrypt it. The need for such schemes is very important. If for example Alice wants to talk with many Bobs, without even letting them know to which one she contacts, then she has to make many encryptions. Moreover, if Malory (an attacker) is spying on Alice's connections, she might not find the message that Alice is sending, but find out that she is talking with Bobs. However, Alice could use Carrie (as a proxy) who is going to use an intermediate key, described previously, with each one of the Bobs, transforming the encrypted message for every one of them independently. The result is that Malory or any of the Bobs, not only cannot detect that Alice is talking with any of the Bobs, but the message is not disclosed to Carrie as well.

The notion of re-encryption was introduced in [3] with the introduction of the so called "Atomic Proxy Cryptography". The re-encryption schemes that have emerged after the original scheme have many applications in private communication channels [8], anonymity [13], [14], secure distributed storage [1] and privacy in cloud computing [17]. Re-encryption can also be used to allow keyword search [16], [6], [15]. In our proposal, re-encryption is used in several parts to allow users share their contents with other entities, as we are going to describe in the following sections.

D. The Raykova-Vo-Bellovin-Malkin scheme

As in our previous work [11], the main tool that is used in order to implement the concept of Privacy as a Product (Paap), is the Raykova, Vo, Bellovin and Malkin (RVBM) scheme [12]. The scheme provides a novel feature: it allows clients to perform queries on a server, without revealing any information regarding the search to the owner or the server. More interestingly, the server does not have access to the contents of the database, as all the contents are encrypted by keys selected by the user.

In order to describe the scheme we need four entities, the *User*, who supplies the data, the *Querier*, who makes queries to the database, an *Index Server* (IS) which hosts the encrypted database, and the *Query Router* (QR), which provides the anonymisation to the queriers.

The User encrypts his files and creates a Bloom filter [4] for each keyword in his entry in the database. In order to allow keyword queries, the user issues some keys and sends them to the Querier. The Querier now, whenever he wants to submit a query, encrypts the query with the given key and sends it to QR. After receiving the query, QR checks whether the Querier has the right to perform the query, and if so re-encrypts the query and sends it to IS. The IS can then perform a search on the Bloom filters, find the corresponding records, encrypt the indexes and send the results to QR. Finally, QR, transforms the results so that the Querier can obtain his search results.

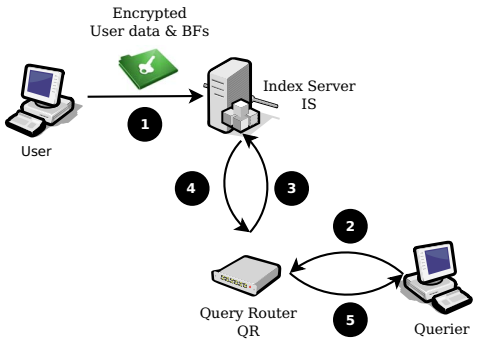


Fig. 1. The RVBM - SADS scheme: **1**- The *User* sends his data and Bloom filters to *IS*. **2**- The *Querier* sends an encrypted query to *QR*. **3**- QR re-encrypts the query and forwards it to *IS*. **4**- *IS* performs the search on the Bloom filters and sends the results encrypted to *QR*. **5**- *QR* replies to the *Querier* with the re-encrypted results.

This way, the Querier can identify which records in the database contain a certain keyword, without revealing this to the IS. The scheme can be further extended to provide aggregation queries as well and not just keyword searches. Figure 1 illustrates the Secure Anonymous Database Search (SADS) scheme of Raykova, Vo, Bellovin and Malkin. For a detailed explanation of the protocol, its security and privacy properties, the reader could refer to [12].

III. Our proposal

A. Main actors and desiderata

In our proposal we have the following entities: the User, the CSP, the Query Router and the Bidder. As it will become apparent later on, the originally proposed Privacy as a Product (PaaP) model will not need significant modifications, even if the User is hosting his own cloud service, however, in order to make the model as general as possible, we present our model considering the "CSP" entity as an independent actor.

The first concern is that the content of user's files must remain in any case secret, so it is essential to encrypt the files before sending them to the CSP. Depending on the disclosure tha the user wants for his data, we propose different modes of encryption. If the user wants them public then there is no need for encryption. However, if he wants to share it with other authenticated users, encryption algorithms that support re-encryption should be used. Finally, for private data, symmetric algorithms should be the choice. For each file the user creates additionally at most three Bloom filters. The goal of these

filters is to allow queries to the owners, the users with which the file is shared and the bidders respectively. For each filter the owner may choose a different dictionary to allow different keyword queries in his files.

Currently, advertising companies pay large amounts of money for targeted advertising. Therefore, in order to get better profiles, they pay big social networks, search engines and the like, to access user profiles and send advertisements to the users, depending on their interests. Given that users of cloud services might upload their music, videos, e-books, etc. an advertising company may create more accurate profiles about users (*e.g.* based on their interests and preferences). There are several issues that we highlight as the main desired features of our model. The **user** should be the **only** one to: **access** all his data, **negotiate** directly with the bidder and **choose** which data can be queried and by whom. On the other side, the CSP should not have any access to the user's data, as it is just hosting the data. Finally, the CSP, which might become a monopoly, like the social network case, should be able to control the whole market.

B. Managing and trading the data

In our scheme, as discussed previously, we have four parties: the User Alice, the Bidder, the Query Router (QR) and the CSP. Each Bidder that wants to perform a query, requests a certificate from QR. By showing this certificate, the Bidder is able to prove to CSP's users that he is a trustworthy entity and he has authorisation to query. Moreover, this certificate enables users to have an evidence to prosecute Bidders that do not fulfill or violate the agreed deals.

Alice, in order to upload a file, sends her data to the CSP, encrypted with the CSP's public key, so that other users cannot tamper with or retrieve them. Along with data, Alice creates and uploads the three Bloom filters previously discussed. The first one is for her private searches, the second one is for use with other users, while the last one is to allow Bidders to query. Obviously, each Bloom filter is encrypted with different key, moreover, the keywords can originate from different dictionaries, to facilitate different needs. In order to revoke a user's access on her shared information, Alice issues a command that asks the CSP to re-encrypt the data, sending him the re-encryption key. The CSP re-encrypts the data with this key and informs the rest of the users of this change. Since the revoked user has no access to the re-encryption key, he cannot access the shared files anymore.

We assume that users operate in a peer-to-peer (P2P) network. This is, that users can exchange information amogst them and with the CSP and the Bidders by means of this P2P network. (Note that there are many possible embodiments for this P2P network. However, for the sake of generality, we leave it open because this decision does not affect our protocol nor our architecture).

Users that operate independently of the CSP must authenticate by providing an appropriate certificate, issued by the CSP. On the other side, using his certificate, the Bidder posts his signed request for query in the P2P network. Users, if they are willing to share with the Bidder the requested information,

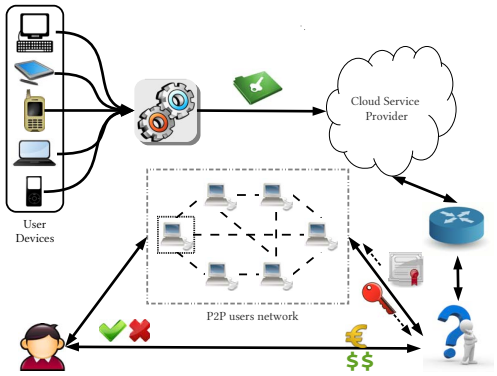


Fig. 2. The proposed scheme for PaaP in the Cloud.

for the appropriate bid, reply to his request, providing him with a key to decrypt the Bloom filters on the CSP and the appropriate file indexes, encrypted with the Bidder's public key. The Bidder now has the appropriate keys to query the CSP, without disclosing any further information and uses the RVBM scheme to retrieve the results.

If the Bidder finds data of his interest, he uses the additional trading protocol, described in [11] to place a bid and receive the results that he wants. Roughly, the protocol has the following steps. Firstly, the Bidder initiates a session with the CSP. The CSP creates a "ticket", the offerID and sends it to the Bidder. The Bidder uses the ticket that he received from the CSP and contacts the User, firstly negotiating the encryption parameters. The User getting the Bidder's requests, replies with the accepted parameters. Using them, the Bidder places his bid, which consists of the hash of the requested messages, that he retrieved when he queried the CSP, the bid value or a key that releases access to a service from t_1 to t_2 or an amount of money and for how long will the offer last. The user either accepts or rejects the offer and signs it. The Bidder acknowledges user's response and if it is positive, the user sends a decryption key, which the Bidder can use to request the file from the CSP and decrypt the content. The proposed scheme is shown in Figure 2. For a detailed analysis of the protocol, the reader may refer to [11].

IV. Discussion and Evaluation

A. Feasibility

Technically we can distinguish two major tasks, the P2P network management and the RVBM scheme. The first task has proven to be feasible and there are numerous examples of P2P networks with many applications and very high efficiency.

Regarding the RVBM scheme, it might be seem as a new scheme, however it has already been implemented with good results in terms of performance [10]. Nevertheless, we want to emphasise that the RVBM scheme is only used as a tool to proof our concept of Privacy as a Product (PaaP) in the cloud computing context, trying to show that our business model can be implemented. Probably, other schemes can provide more efficiency and a significant performance boost, however in this work we only concentrate on the fact that PaaP can be used

within the Cloud model. It could be argued that our solution could be implemented with the use of fully homomorphic encryption or PIR schemes, however, their computational cost is so high that they cannot be considered practical.

With regard to the viability of the proposed business model, we have to note that in order for a business to be viable, we have to fulfill the following requirements:

1) All parties should gain from this partnership.
2) The parties will be "forced" to play fair and trying to break the partnership will make them lose more than the others.
3) The economic revenue is tempting enough to make several entities to adopt it.

In the following lines we discuss the first point and we leave the others for the next sections (§ IV-B discusses the enforcement of fair play, and § IV-C elaborates on the economical benefits of our model).

To show that all parties benefit from our proposed model, let us analyse what they gain from their partnership.

- **Users**: The proposed model not only is more privacy aware, but simultaneously, puts users in full control of their data and enables them to have some revenue.
- **CSP**: It might appear that the CSP loses a lot of its power, as a big amount of income is lost. However, the CSP becomes more trustworthy, the proposed scheme attracts more users, as they may now trust the Cloud, which means more subscriptions for the CSP. Additionally, the CSP can claim a percentage of every sealed deal between a Bidder and a User, and charge the QR for every executed query. So, in the end, CSP might gain users and increase their revenue.
- **QR**: Query routers not only forward the queries but become certification authorities. Therefore, they can claim an amount of money for forwarding and anonymizing the queries as well as for the certification of the Bidders.
- **Bidder**: The current business model is very restrictive for the Bidders. Unluckily, the source of information has been completely centralised, by major companies, so that Bidders, in many cases, have no alternatives when they want access to information. The proposed scheme rearranges the structure, by allowing the user to have full control of his information and to come in direct contact with the Bidders, cutting in some sense any interventions. Therefore the Bidder plays a more central role in market regulation. As a result, the Bidder may achieve better prices and might gain direct access to the source of information.

Finally, an important issue that remains open is the selfish behaviour of the users. If all users decide to "free-ride", will this behaviour put in jeopardy the proposed business model? According to [2], by the end of 2011, 96% of the users of Dropbox were using the "free-ride" option, which does not stop them from encrypting the files, blinding the CSP from any information. Yet, the company had big revenues and continued its huge growth. In another approach, where by design the CSP does not have access to the information that is hosted by the users, SpiderOak provides a limited amount of space freely. In either case, we see that even if the users choose the "free-ride" option the CSPs still have at least significant revenues. Therefore, the key to make users act more "openly" is to provide them with interesting deals and much more security and privacy.

B. Enforcing Fair play

One more important question in the proposed scheme is how do we know that the scheme is going to be implemented as proposed? Some parties may try to "join forces" in order to maximize their incomes, minimising accordingly the incomes of the other parties.

Firstly, for privacy reasons, the users will not co-operate with the CSP, unless some of them believe that they should return to the current business model. Secondly, for the same reasons the Bidder will not co-operate and disclose the keys to either QR or the CSP. Even in the extreme case where a Bidder is trying to get some last money, because he is closing the business, the keys that he owns will enable him to decrypt only old copies, if they exist, of the information that he had access to.

In another scenario, QR and CSP co-operate. If the CSP becomes aware of what queries are being made, the best thing that he can do is to get in contact with other Bidders and receive the queries straight from them, circumventing the QR. It is clear that from such partnership the QR only has to lose.

Since, the CSP is the one who provides users with the certificates to join the P2P network, he is able to masquerade whenever he wants as a user. This way, the CSP may trace what queries do the Bidders want. "Replaying" previous results, with a fake user account one might argue that this can create a probable attack. However, all that this could create is an intriguing account, which the Bidder might find interesting, request access, but since the CSP does not have access to the decryption keys, he cannot extract any additional money from the Bidders.

The last scenario involves the Bidders and the Users. Since they come in contact to retrieve the decryption keys for the Bloom filters, they might as well trade the data, circumventing the CSP and depriving him of his share of the deal. This case seams rather feasible, but it is not a real risk. As said before, the Bidder, in order to query, sends his request to the P2P network of the users, some of them reply and forward their keys to the Bidder, along with the corresponding record IDs. If the Bidder finds something interesting, then normally he has to contact the CSP, request the userID and place the bid. If this is not the case, then the Bidder has to return to the P2P network, and request the users with the userIDs that he found interesting. This action can be traced by the rest of the users due to the P2P nature of the network. If the bidder masquerades as a normal user in order to circumvent this obstacle, then the user with the data loses any guarantee that he will get the promised bid, as he has to trust someone who is a normal user as him. Moreover, in order to get all the necessary information, and to be trusted, the Bidder has to create many fake accounts in a small period of time or to create users with huge chains of certificates. In both cases the

actions can be traced, blocked and reported either by the CSP or by the rest of the users who are left behind the "secret deals".

C. Indicative economical effects of the proposal

To briefly illustrate the possible economic effects of our proposal we use Dropbox as the CSP, who not only has published several numbers about its services [5], but is one of the major CSPs of small end users and one of the most rapidly developing companies. Currently it has around 100 million users, from which around 95% are users of the free service. The income for 2011 was around $240 million, when the users where around 50 million, meaning that they obtain an average income of around $5 per user. Even if the number of users is very big, that does not mean that the stored information is very important, as most users are quite reluctant to store important, hence valuable, information in the provided space.

If our proposed model is adopted, privacy that is the biggest obstacle towards embracing the cloud is circumvented. A conservative scenario, would expect a 2.5% increase in the number of users. With the aforementioned values this translates to $12.5 million of added income. This amount of money has to be split amongst the users, the CSP and the query provider. A more realistic scenario would be a 5% percent increase, which translates to $25 million of added income. The optimistic scenario of the adoption of PaaP in the Cloud would result in a 10% percent increase of users, which translates to $50 million of added income.

The numbers provided in this section are only indicative and do not have strict scientific value. However, the main point that we want to emphasise is that from an economical point of view all parties could benefit from the adoption of the PaaP model.

V. Conclusions

Cloud computing is a new paradigm that is gaining importance steadily. The growing use of mobile devices along with PCs, tablets and the like, has led to the need for synchronisation amongst all these devices – thus the cloud appears like a very promising choice. However, using cloud services might endanger the privacy of users and might be seen as a break for the proper development of the society.

In this article, we have proposed a new architecture grounded on well-known cryptographic primitives that allows users to regain control over their data. By doing so, we have proposed a new business model that considers users, bidders and CSPs. We have shown that the economical goals of bidders and CSPs can be reached whilst the privacy of the users is preserved. Thus, our strategy proves to be a "win" strategy for all parties.

Moreover, we have shown that the concept of Privacy as a Product is viable in the context of cloud services and, in addition to be fairer for the user, it can increase the revenues of CSPs by guaranteeing the most important value of users in the cloud – their privacy.

[5] http://www.dropbox.com/static/docs/DropboxFactSheet.pdf

Disclaimer and acknowledgements

Agusti Solanas is with the UNESCO Chair in Data Privacy, but he is solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organisation. This work was partly supported by the Government of Catalonia under grant 2009 SGR 1135, by the Spanish Government through projects TIN2011-27076-C03-01 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", by the European Comission under FP7 projects"DwB" and "Inter-Trust", and by La Caixa under program RECERCAIXA 2012.

References

[1] Giuseppe Ateniese, Kevin Fu, Matthew Green, and Susan Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, February 2006.

[2] Victoria Barret. Dropbox: The inside story of tech's hottest startup. http://www.forbes.com/sites/victoriabarret/2011/10/18/dropbox-the-inside-story-of-techs-hottest-startup/, Nov. 7, 2011.

[3] Matt Blaze, Gerrit Bleumer, and Martin Strauss. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *Lecture Notes in Computer Science*, pages 127–144. Springer Berlin / Heidelberg, 1998.

[4] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM*, 13(7):422–426, July 1970.

[5] European Commisision. Unleashing the potential of cloud computing in europe. http://ec.europa.eu/information_society/activities/cloudcomputing/docs/com/com_cloud.pdf, 27/9/2012.

[6] Liming Fang, Willy Susilo, Chunpeng Ge, and Jiandong Wang. Chosen-ciphertext secure anonymous conditional proxy re-encryption with keyword search. *Theoretical Computer Science*, 462:39–58, 2012.

[7] Arash Ferdowsi. The dropbox blog. https://blog.dropbox.com/index.php/yesterdays-authentication-bug/, 2011.

[8] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *In proceedings of the 2004 RSA conference, Cryptographer's track*, pages 163–178. Springer-Verlag, 2002.

[9] Eric Hughes. A cypherpunk's manifesto. http://www.activism.net/cypherpunk/manifesto.html, 1993.

[10] Vasilis Pappas, Mariana Raykova, Binh Vo, Steven M. Bellovin, and Tal Malkin. Private search in the real world. In *ACSAC*, pages 83–92. ACM, 2011.

[11] Constantinos Patsakis and Agusti Solanas. Privacy as a product: A case study in the m-health sector. In *Proc. 4th International Conference on Information, Intelligence, Systems and Applications (IISA 2013)*, July 2013. (to appear).

[12] Mariana Raykova, Binh Vo, Steven M. Bellovin, and Tal Malkin. Secure anonymous database search. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 115–126, New York, NY, USA, 2009. ACM.

[13] Jun Shao. Anonymous id-based proxy re-encryption. In *Information Security and Privacy*, volume 7372 of *Lecture Notes in Computer Science*, pages 364–375. Springer Berlin / Heidelberg, 2012.

[14] Jun Shao, Peng Liu, Guiyi Wei, and Yun Ling. Anonymous proxy re-encryption. *Security and Communication Networks*, 5(5):439–449, 2012.

[15] Xu An Wang, Xinyi Huang, Xiaoyuan Yang, Longfei Liu, and Xuguang Wu. Further observation on proxy re-encryption with keyword search. *Journal of Systems and Software*, 85(3):643 – 654, 2012. Novel approaches in the design and implementation of systems/software architecture.

[16] Wei-Chuen Yau, Raphael C. W. Phan, Swee-Huay Heng, and Bok-Min Goi. Proxy re-encryption with keyword search: New definitions and algorithms. In *Security Technology, Disaster Recovery and Business Continuity*, volume 122 of *Communications in Computer and Information Science*, pages 149–160. Springer Berlin Heidelberg, 2010.

[17] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, march 2010.