Regaining the end-users' trust with transparency-enhancing tools

by R. Peeters[1] and T. Pulls[2]

Abstract

With the constant news of data breaches and global (governmental) surveillance, end-users are becoming more and more reluctant to share sensitive data. As a result trust of end-users is an essential business enabler. For the end-user, sharing data with companies is often a prerequisite for using their services. Moreover, for certain services, e.g., governmental and healthcare services, end-users do not really have the option to not share personal data with these services.

There is an information asymmetry between the organisations that handle personal data and the individuals to whom these data relate: these organisations know more about the individual than just the data disclosed by that individual (e.g., inferred data, enriched data from combining these with other databases), while the individual has little information on what these organisations do with the collected personal data. By offering transparency, e.g., through the deployment of transparency-enhancing technologies (TETs) at service providers, while giving individuals insights in what is happening with their personal data, organisations can reduce this information asymmetry and profile themselves as trust-worthy. With these insights, end-users can hold organisations more accountable for their actions and file a complaint in case of abuse. With the upcoming European data protection regulation, organisations will also be required to offer some kind of transparency.

An important prerequisite for individuals to put trust in the system is the integrity of the data generated by such a transparency-enhancing tool. In particular, it should be impossible to alter the information on which the individual bases his/her insights, i.e. metadata about personal data processing. Such a tool should also take into account both data privacy and confidentiality, since the mere existence of metadata already reveals information, e.g., the individual visited the hospital. Ideally, the integrity of the data generated by the tool could also be verified by a trusted third party or auditor without infringing on the individuals' right to privacy.

Within project Opacity[3], we developed a transparency-enhancing tool that meets these criteria: integrity, data privacy, confidentiality and public verifiability. This solution is based on strong state-of-the-art cryptographic building blocks and validated protocols. It allows organisations to easily generate an end-user specific history of data processing in real time, which can be consulted by the end-user at any point in time. The end-user is ensured of the integrity, confidentiality and timeliness of the presented events. The integrity of the global history for all end-users of a single organisation can be validated by an external auditor without violating the end-users' privacy. Finally, this tool can also be deployed for processes (data handling) that span multiple organisations. A reference implementation of this tool, named Insynd, is freely available under open source.

[1] KU Leuven, COSIC and iMinds

[2] Karlstad University, Department of Mathematics and Computer Science

[3] Project Opacity: http://www.project-opacity.com

Introduction

With our society becoming more and more digital and interconnected, a lot of new and previously unimaginable services have emerged that facilitate our day-to-day lives. However, together with these new services, our personal data gets ever more widely disclosed and out of our control, resulting in new severe privacy issues.

One of the core principles of privacy is that of data minimisation, service providers should only store the minimal amount of personal data and for the minimal duration they need these in order to provide their service. Ideally from this privacy perspective, end-users do not disclose any personal information at all. However, in many situations, disclosure of one's personal data is:

- in one's best interest, e.g., healthcare;
- a practical necessity for the service provider for providing one with the service, e.g., for mobile phones, the location of the nearest cell tower is needed to route incoming calls to;
- dictated by the business model of the service provider, e.g., one's billing information for paid services, one's profile on free email or social media accounts that rely on advertising for generating their revenue.

In some cases, end-users even have no choice than to share certain personal data, e.g., governmental services, taxes.

End-users can rely on their legal rights and make use of Privacy-Enhancing Technologies (PETs) [1] to protect their privacy. The right for privacy has been recognized in Article 12 of the United Nation's universal declaration of human rights[4], where it is stated that everyone has the right to protection of the law against arbitrary interferences on their privacy. Next to this strong fundamental right there are data protection regulations and privacy laws both in Europe and the USA, such as the Data Protection Directive and the Fair Information Practise Principles respectively, which can be general or sector specific like for example in healthcare. However, from a technical point of view, once data has been disclosed, it is hard to protect it in the sense that one is no longer in control of what is happening with these data. That is why PETs that can be deployed by the end-user are designed to assist the end-user with disclosing only the minimal amount of personal data needed to make use of the service. Well known examples of such these PETs include PGP[5] (Pretty Good Privacy) encryption for emails, anonymous browsing through the use of Tor[6], and anonymous credentials systems like Idemix[7] that can, e.g., prove that one is of legal drinking age without disclosing all other attributes on one's passport/driver's license.

Once personal data has been disclosed, end-users only hold limited information on the flow of their personal data beyond what little information the service provider's privacy policy offers (if it is read at all by the end-user). There are two big exceptions: companies have to report data breaches (for instance like recently happened with the OPM breach where a highly sensitive personal data of 21.5 million US federal workers were compromised[8]), and the right of individuals to request which data service

---

[4] http://www.un.org/en/documents/udhr/

[5] https://gnupg.org/

[6] https://www.torproject.org/

[7] http://www.zurich.ibm.com/idemix/

[8] http://edition.cnn.com/2015/07/09/politics/office-of-personnel-management-data-breach-20-million/

providers store about them (e.g., Facebook[9] and Google[10] have special web pages where you can get an overview of the data they have about you). Both exceptions can be seen as a form of transparency: end-users get some limited insights in which personal data was stored where at one specific point in time (the time of the breach, or the time of the end-user consulting the privacy overview). However, this still does not inform the user about the actual data processing by the service provider taking place on the disclosed personal data, i.e. when the data was consulted, updated, merged or shared with another party and for which reasons. This means that the information asymmetry between the end-user who has little information about what is happening with the disclosed data and the service provider who has a lot of information about the user (more than merely the personal data as disclosed by the end-user) continues to exist. This unknown, alongside with the constant news of global (governmental) surveillance, often with the cooperation of service providers, have a negative impact on the trust of end-users in the service providers. By making data processing transparent, this information asymmetry is reduced: instead of giving a general consent to data processing of their personal data, end-user can now see what is happening with their data and revoke consent. The end-user is given back partial control over his/her personal data and can, if deemed necessary, take actions like changing service providers or filing a complaint. This is currently already the case in healthcare where individuals have the right to consult their medical journals and the corresponding access logs. If a patient (after reading his/her medical details in a newspaper) is looking through the access log of his/her medical journals sees that a nurse from an unrelated unit in the hospital was looking into his/her medical journal, can file a complaint with the ombudsperson of the hospital.

Clearly privacy is about more than data minimisation and service providers making sure that data does leak to third parties (security). It is also about taking into account the rights of the individual, being transparent and accountable. This is also recognised by the EU's upcoming data protection regulation, which defines eight privacy principles among which transparency.

Within the project Opacity, we have built a transparency-enhancing tool (TET) that allows service provider to inform users about the actual processing on their data. The rest of this paper is structured as follows: first we give a high-level overview on TETs and the meaning of data minimisation, then we give an overview of project Opacity and describe the underlying technology, and provide our conclusions.

The ideal TET and the role of data minimisation

Transparency-Enhancing Tools (TETs) are tools that at their core provide individuals with information that concerns their privacy [2]. These tools can be both legal and technological. Legal TETs, like the EU Data Protection Directive 95/46/EC, give individuals access to their personal data at controllers and specify information to be shared, e.g., when further personal data is obtained from third-parties (Articles 10-12). TETs can further be categorised into ex-ante and ex-post TETs. Ex-ante TETs provide information to an end-user *prior to* the user disclosing his/her personal data to the service provider. Ex-post TETs provide data to the user *after* the user disclosed his or her personal data. A prime example of an ex-ante TET, that provides information *before* a user discloses data, is that of a privacy policy of a service provider. The privacy policy

[9] https://www.facebook.com/settings/

[10] https://www.google.com/dashboard/

of a service is something that a user should be presented with *before* signing up to a service such that the user can give *informed consent* to the processing of the user's personal data. An example of a technological ex-post TET is Mozilla Lightbeam[11], which is a Firefox add-on that visualises relationships between sites and their embedded links to third-party sites as you browse the web. Mozilla Lightbeam is an ex-post TET, because it informs a user of the relationships between sites *after* the end-user has visited these.

The aim of a TET is to reduce the information asymmetry between the end-user and the service provider, by increasing the transparency towards the end-user. Information asymmetry in this case means that the end-user and service provider have access to different information about an end-user's personal data, such that one party knows more than the other. Typically the service provider knows more about the end-user's personal data, since the service can deduce more information about the end-user due to, e.g., correlation with other data sources or access logs. In essence, the goal of TETs is to reach a state where neither the end-user nor the service provider could learn any information about the end-user's data disclosure [3]. In other words, both the end-user and the service provider have perfect knowledge about the data disclosure. There are a number of issues with this stated goal of TETs, but it tells us something useful about the relationship between TETs and the privacy principle of data minimisation.

When deploying a TET, one must be careful to not actually *increase* the information asymmetry between the end-user and service provider. For example, a TET could leak information about the end-user to the service provider, e.g., due to how the TET facilitates the transfer of data from the service provider to the end-user. It should also not be possible for the service provider to tamper with the information it shares with the end-user. The lesson is straightforward: TETs that attempt to address information asymmetry should strive to minimise the new data they produce to function optimally. Furthermore, security and privacy of TETs are of paramount importance to ensure that TETs are not used against end-users by their service providers.

Project Opacity

Project Opacity is a collaboration between researchers are KU Leuven, Belgium, and Karlstad University, Sweden, focused around technological tools for providing transparency with strong cryptographic guarantees. One application of the technology is to provide end-users of services with an overview on what is happening with their (sensitive) data at different service providers. As such, the end-user can claim back some control over their data and if necessary, take appropriate actions, e.g., file a complaint or switch to another service provider. By offering this kind of transparency, service providers effectively commit to their actions and show to their end-users that they are trust-worthy.

The approach of project Opacity is to create a relevant TET that can give solid cryptographic guarantees in terms of security and privacy. On the one hand, relevant for our transparency-enhancing technology means that it can provide meaningful feedback to end-users about the integrity and time of the stored metadata without introducing new privacy risks for these end-users. On the other hand, relevant also means that the technology can easily be deployed by service providers, i.e. with minimal impact on existing business processes.

---

[11] https://www.mozilla.org/en-US/lightbeam/

Our technology allows service providers to store metadata, describing how their end-user's data is being processed, for their end-users, who can later-on retrieve the metadata concerning their data. While being agnostic to the content of the metadata, the actual description of data processing, our technology contributes to the meaningfulness of the retrieved metadata by deploying technical measures to ensure the integrity of the metadata, i.e. metadata cannot be altered after being stored together with some time granularity on when these metadata were stored.
Towards not introducing new privacy risks for end-users, the stored metadata is encrypted and made unlinkable for anyone but the end-user, who can link back together metadata stored for him/her and decrypt these. Furthermore, we have looked into how to ensure that if the end-user wants to file a complaint with a third party for a specific description contained in one of the retrieved metadata, he/she can do so without introducing privacy leaks with respect to the rest of the metadata stored for him/her while convincing the third party of the validity of the complaint.
Through our high-level RESTful APIs, service providers can easily store descriptions for their end-users in real time while running their business processes that deal with end-user data. Because of the strong integrity and privacy properties, it is possible to store the data with any commercial cloud provider.
To summarise, Opacity provides the following:

- Integrity: what goes in comes out again together with a guaranteed integrity and within some verifiable time frame.
- Privacy: metadata are stored encrypted and unlinkable, the end-user can make a verifiable complaint to a third party while only disclosing selective data.
- Deployability: record what is happening when it is happening, storage can safely be outsourced.

Technology

Opacity provides a transparency-enhancing tool in the form of a cryptographic scheme that enables companies to inform end-users about the actual data processing that takes place on their personal data. This is done by storing and serving encrypted metadata, generated by those companies, about the data processing, in a secure and verifiable way. By removing any link between stored data, we go beyond traditional solutions. The result is transparency with maximal privacy for both the individual and the organisation or commercial entity. The technology was developed to handle large volumes, have a minimal impact on existing company processes and is easy to implement.
Our technology has three major components as shown in Figure 1: the sender component, the storage component, and the consult component. The service provider will store metadata for its end-users by using the sender component API, which will encrypt the metadata and ensure that it is unlinkable before forwarding it to the storage component. The end-user will retrieve his/her relevant metadata by using the consult component API that decrypts and links back together the metadata. This three components model covers a large number of use cases, from service providers wishing to share (potentially privacy sensitive) information with its customers to auditors wishing to construct a publicly verifiable audit trail of an auditing process.

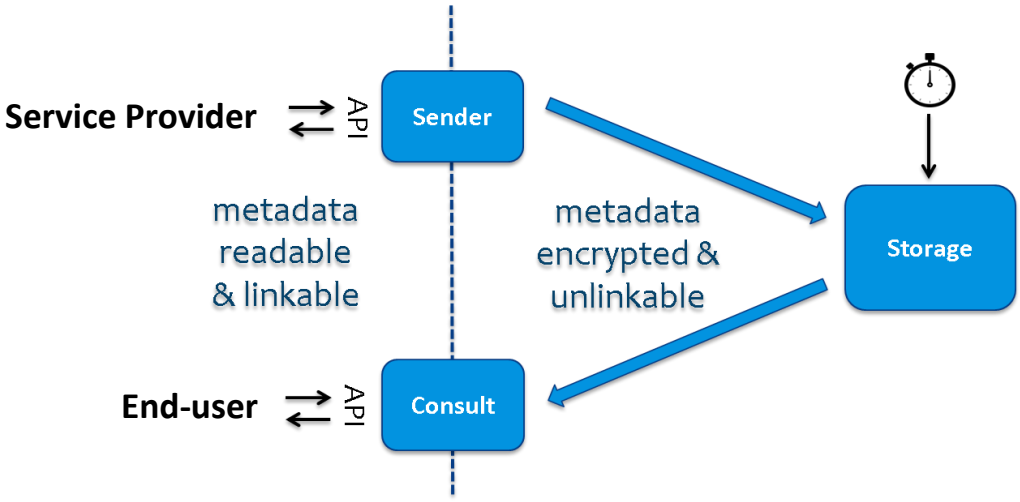

Figure 1: Opacity technology consisting of three major components: sender, storage and consult.

Our implementation makes use of a state-or-the-art cryptographic library NaCl [5], which is an easy-to-use high-speed library with many modern cryptographic algorithms, such as Curve25519, Ed25519, and SHA-512. This library is used for the cryptographic building blocks necessary to ensure the metadata is encrypted and unlinkable. For the integrity, we make use of a block-chain alike technology, an authenticated data structure named Balloon [4].

Block chains, snapshots and time

Bitcoin is a popular cryptocurrency that relies on a *block chain* of *transactions* in conjunction with a *proof of work* mechanism. Both the block chain and proof of work rely on the hardness of finding collisions for cryptographic hash functions, i.e. it is difficult to come up with two distinct messages of the same length for which the output of the hash algorithm is identical. As such the output of a hash algorithm effectively fixes the input. A transaction with the bitcoin network transfers an amount of bitcoin *C* from a payer *A* to a payee *B*, where the payer *A* signs the transaction to establish that the transaction was authorised by *A*. In the block chain, each block consists of a number of transactions structured in a Merkle (hash) tree. Blocks are then linked together in a way that requires miners to perform a proof of work by finding a particular output of a cryptographic hash function. The network achieves consensus on the block chain thanks to the proof of work being challenging and no single entity controlling a too big part of all the computing power in the network[12]. Figure 1 shows a sketch of the Bitcoin block chain and how transactions are structured in each block.

[12] This is due to the conflict resolution mechanism in case two miners come up with a new block roughly at the same time, and the chain forks. In this case, it is the longest chain that is recognized in the bitcoin network as the valid one, invalidating blocks in the side-chain. If a single entity controls a large part of the network, it becomes possible for this entity to invalidate parts of the main chain, by creating a longer side-chain that then at some point becomes the main one.

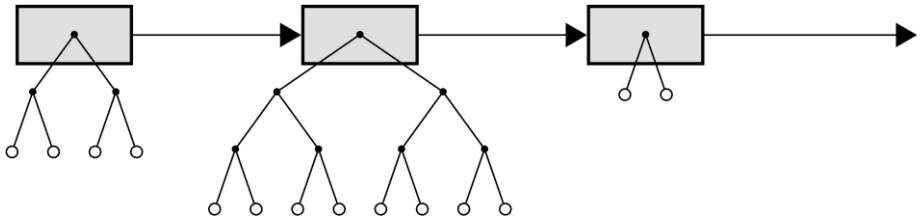

Figure 2: The block chain and transactions stored in trees.

In Opacity, we use an append-only authenticated data structure named Balloon. Balloon enables the sender component to outsource the storage of append-only data to the storage component and to let the storage component provably answer queries made by the consult component on the data. The consult component can verify that replies from the storage component are correct with regard to what the sender stored at the storage component. This is possible without placing additional trust into the storage component (beyond availability) thanks to the use of cryptography (the `authenticated` part in the term authenticated data structure). Balloon is the composition of two tree-based data structures: a hash treap and a history tree. The history tree is basically a versioned Merkle tree, like the data structure that is used in each block in Bitcoin to structure transactions, and contains all data stored in the Balloon. The hash treap is a type of *sorted* Merkle tree that acts as an authenticated *index* over the data stored in the Balloon. Each version of a Balloon results in a snapshot that fixes all data stored in the Balloon so far. A snapshot is analogous to a block in the block chain, with the key difference being that it is the sender component that signs the snapshot[13], and no proof of work is involved. The security comes from the snapshots being broadly available and only the sender component being able to create valid snapshots. Figure 2 shows a sketch of how Balloon grows over time and produces new snapshots for each version. Note that data is always appended to the trees in Balloon, as compared to the Bitcoin block-chain, where each block has its own tree with new transactions. Just as the blocks in the block-chain, snapshots are linked together such that the latest snapshot/block fixes everything that has taken place so far.

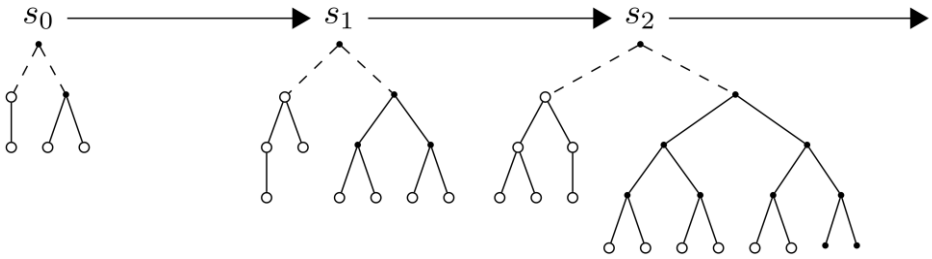

Figure 3: Two trees make up Balloon, each version fixed by a linked snapshot.

To ensure that snapshots are broadly available and to lock these in time, we make use of the Bitcoin block-chain. Bitcoin can be seen as a distributed time-stamping scheme, where performing proof of work to agree on the block-chain fixes the time each block was added to the chain: the difficulty for the proof of work is adjusted daily, more precisely every 182 blocks, such that the time to generate a block is on average ten

[13] Note that since there is only one entity that stores metadata into a Balloon, the origin of the metadata can be established by the signatures on the snapshots and the individual metadata do not need to be signed separately.

minutes. We periodically insert the snapshots produced in Balloon as transactions in the Bitcoin block-chain. Figure 3 illustrates this, where the snapshots s_0 and s_2 are put as transactions into two blocks in the block-chain.

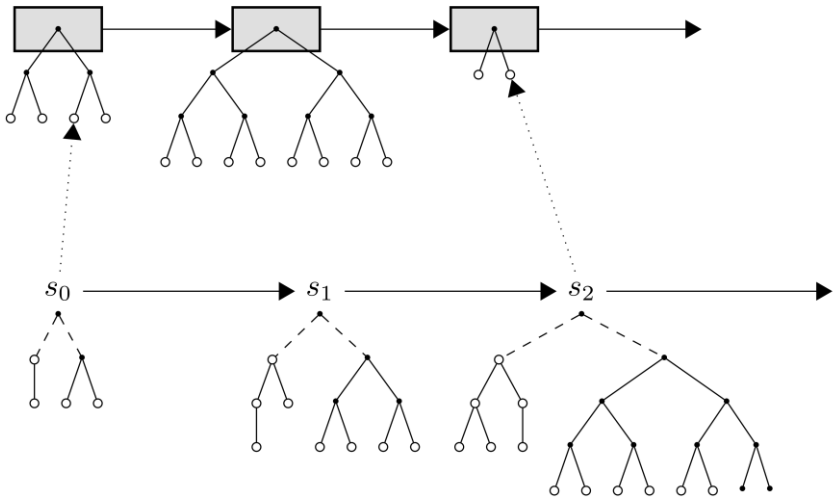

Figure 4: Snapshots from Balloon are periodically put into Bitcoin transactions in the block chain.

With the above structure, it is possible to create a proof that shows from a Bitcoin transaction in the block-chain that a particular piece of data *existed* at a particular *time*, and that the data was sent by a particular *service provider* to a particular *end-user*. This proof is publicly verifiable, in the sense that anyone can verify the proof using standard cryptographic operations and public information. Furthermore, it is also publicly verifiable that all snapshots that have been put into the block-chain for a Balloon are *consistent* and have not been tampered with.

There are advantages of using Opacity over pure block-chain: for applications with one (or few) contributor(s) to the data in the chain, the setting of Opacity matches better. Security of the block-chain's proof-of-work is highly dependent on having a vast amount of computational power in the mining network to do the proofs of work, making it practically impossible for a single entity to control a majority of it. This is where there is also a danger in using the block-chain technology with only a small network or miners, especially if one cannot use the Bitcoin block-chain. Opacity, based on Balloon, is rock solid security wise: it relies on standard assumptions about hash functions and signature algorithms. Opacity can still use a block-chain to fix all data, and get rough timing information as a bonus. For more exact timings, one could also easily use other sources like time-stamping authorities.

Implementation and source

The three components that make up the core of Opacity are written primarily in the memory safe programming language Go[14]. Each component exposes a RESTful API for easy integration. All storage is authenticated and encrypted at rest, with easy to follow procedures for key management and backups. Opacity also supports using cloud services for redundant storage, such as Amazon S3.

[14] https://golang.org/

The core of Opacity, namely Balloon and a related technology named Insynd, was developed as part of the on-going EU FP7 project A4Cloud[15]. Both Balloon[16] and Insynd[17] are available as open source under the Apache2 license.

Conclusions

In this paper we presented Opacity, a transparency-enhancing tool in the form of cryptographic scheme that enables companies to be transparent about processing of personal data towards their end-users. By providing this kind of transparency, not only will companies have an easier time being compliant with the upcoming EU data protection regulation, but more importantly show to their customers that they are trust-worthy, which is an important business enabler.
The technology behind Opacity is block-chain alike, but better suited for settings with one (or a few) metadata provider(s), i.e. the service providers that provide descriptions of their data processing, and provides stronger security. By encrypting the metadata and making these unlinkable we ensure strong privacy properties. Finally it should be noted that Opacity was developed with easy deployability in mind, i.e. with minimal impact on existing business processes a company should be able to start using Opacity to start storing metadata for its end-users.

Acknowledgements

Roel Peeters has received funding from the Seventh Framework Programme for Research of the European Community under grant agreement no. 607049. Tobias Pulls has received funding from the Seventh Framework Programme for Research of the European Community under grant agreement no. 317550. This work was supported in part by the Research Council KU Leuven: GOA TENSE (GOA/11/007).

References

[1] C. Diaz, S. Gürses and O. Tene, "Hero or Villain: The Data Controller in Privacy Law and Technologies," in *Ohio State Law Journal Symposium "The Second Wave of Global Privacy Protection"*, 2012.

[2] H. Hedbom, "A Survey on Transparency Tools for Enhancing Privacy," in *The Future of Identity in the Information Society*, 2008.

[3] S. Berthold, S. Fischer-Hübner, L. A. Martucci and T. Pulls, "Crime and Punishment in the Cloud," in *Trustworthiness, Accountability and Forensics in the Cloud*, 2013.

[4] R. Peeters and T. Pulls, "Balloon: A Forward-Secure Append-Only Persistent Authenticated Data Structure," in *ESORICS*, 2015.

[5] D. J. Bernstein, T. Lange and P. Schwabe, "The security impact of a new cryptographic library," in *LatinCrypt*, 2012.

[15] http://www.a4cloud.eu/
[16] http://www.cs.kau.se/pulls/balloon/
[17] http://www.cs.kau.se/pulls/insynd/