Exploiting the Blockchain to Guarantee GDPR Compliance while Consents Evolve under Data Owners' Control

Massimiliano Calani, Giovanni Denaro and Alberto Leporati

University of Milan-Bicocca, Department of Informatics, Systems and Communication, Viale Sarca 336, 20126 Milano, Italy

Abstract

The last 20 years have seen increasingly wide spread of online services, including the advent of social media, and therefore increasingly massive sharing of personal data between users and companies, thus underscoring the importance of protecting the privacy of any involved personal data and avoid abuses. In 2018 the General Data Protection Regulation (GDPR) came into force, committing companies to comply with lawful rules that stress their role and responsibilities in protecting the privacy of the legal persons that share personal data with them. In this paper we address the crucial challenges that companies face to achieve compliance with GDPR, and specifically to i) let data owners full visibility and control on the consents related to their own personal data, and ii) design services that can cope with consents that may change or be revoked dynamically. We propose a solution that relies on the blockchain technology to let data owners grant, access and rectify their consents in a decentralized peer-to-peer fashion, while guaranteeing consensual agreement of data owners and companies on the status of the relevant consents at any time. Although blockchains let all users access all contents freely, our solution suitably exploits encryption to both guarantee the integrity of the consents, and avoid any disclosure to third parties. At the company side, our approach settles a compliance broker that works in a publish-subscribe style to assist services in controlling their compliance to GDPR while the status of consents evolves on the blockchain.

Keywords

Blockchain, Privacy, Consents, GDPR

1. Introduction

GDPR is the General Data Protection Regulation, a law put into effect on May 25 2018, which states the rights of legal persons to pursue and be guaranteed of the privacy of any personal data, against the companies who may process those data to offer them services [1, 2]. The GDPR regulation acknowledges that processing persons' data indiscriminately and carelessly can easily lead to disclosing and loosing control on sensitive information, with many possible critical impacts, from social incidents, up to threatening the well being and the life of some people. For instance, examples of sensitive personal data include (but are not limited to) racial origins, ethical facts, religious and political believes, health-related data, sexual habits, bank

ITASEC'21: Italian Conference on CyberSecurity, April 7–9, 2021, Online
m.calani@campus.unimib.it (M. Calani); giovanni.denaro@unimib.it (G. Denaro); alberto.leporati@unimib.it (A. Leporati)
0000-0002-7566-8051 (G. Denaro); 0000-0002-8105-4371 (A. Leporati)

account and credit card numbers, geo-localization data, and so forth. Thus, the GDPR regulation commits companies to process the persons' data with maximum care, risk awareness and legitimated permissions, granting the data owners the right to negotiate the scope of their consents, and modify or withdraw their consents at any time. Uncompliant companies may face reputation losses and risk fines up to 4% of their turnovers. Worth noting, by explicit choice of the regulation authorities, the GDPR does not indicate which technical solutions the companies shall adopt for being compliant; it just sets the goals.

In this context, this paper focuses on the crucial organizational and technological challenges that many companies face nowadays to (i) give data owners full visibility and control on the consents related to their own personal data, while (ii) guaranteeing that the company's applications and processes remain GDPR-compliant with respect to consents that may dynamically be changed by the data owners. On one hand, letting data owners control the consents on their personal data requires to institutionalise procedures to readily receive and handle all inquiries from those people. On the other hand, the applications and the processes that work with any consented data asset shall be capable of promptly reacting and adapting whenever any relevant consent gets modified or withdrawn. Perspective solutions shall also enable companies to document their compliance with the consents of the data owners, e.g., for auditing purposes, and provide lawful basis on the status of any relevant consents at any time, e.g., to handle possible litigation cases.

Existing solutions fall short in addressing the above challenges. Some solutions address consents and GDPR-compliance with respect to the data that Web applications store into *cookies*, e.g., to remember and profile their users [3]. These solutions exploit the technical fact that Web applications store the cookies-data on the users' machines (then under direct control of users), but do not generalize to many other applications that store the relevant data into databases or files at the company-side. Commercial tools like OneTrust [4] or Absolute [5] address asset management of GDPR-relevant data, providing functionality for risk analysis and monitoring, but without support for data owners to control their consents directly.

Recently some researchers proposed approaches that allow visibility and control of data owners on their consents by exploiting blockchain technologies [6, 7, 8, 9]. These approaches store consents or consent-relevant data in a decentralized fashion on the blockchain, aiming for such information to remain permanently available to all parties, with consensual agreement on its latest status [10]. For example, both the approach of Zyskind et al. and the one of Wirth and Kolain let data owners register on the blockchain the hash of the personal data that they like to share, along with smart contracts that interested third parties execute to request consents on those data [6, 7]. A similar approach discussed by Vargas allows for data owners to log digital consent records on the blockchain, and for data processors (e.g. companies) to execute blockchain transactions that refer to those records whenever they start processing actions [8]. In this way, the data owners maintain continuous visibility on both their consents and the processing actions executed correspondingly. Recently, Guggenmos et al. discussed two crucial design principles that any blockchain-based solution should respect for being GDPR compliant [9]: (i) no personal data must be directly shared on the blockchain, since those data could not be ever removed thereafter, and (ii) attributing data stored on blockchains to natural persons should be possible only based on secure off-chain mapping mechanisms, because the information on the blockchains is publicly accessible in a peer-to-peer fashion. While all the aforementioned approaches ([6, 7, 8]) respect the former design principle, none of them protects satisfactorily the confidentiality of the information stored on the blockchain. Furthermore, to the best of our knowledge, no previous approach copes with keeping the services and the processes of a company in-synch with consents that may evolve dynamically under the control of the data owners.

In this paper we propose a blockchain-based approach that addresses in an integrated fashion both the problem of providing data owners with full visibility and control on their consents, and the problem of managing the compliance of applications and processes that work with those dynamically-evolving consents. Our approach exploits the blockchain to store the consents in digital form, while guaranteeing the integrity of the association between the consents and the corresponding off-chain personal data, and without disclosing any sort of information to third parties other than the data owner and the data processor (a company) themselves. At the company side, our approach settles a compliance broker that works in a publish-subscribe style to assist applications and processes in staying GDPR-compliant. The compliance broker polls the blockchain regularly, in order to identify changes that relate to any consent granted to the company, and notifies its subscribers (the company services) in real-time of any change of the consents on which they depend. At any time, the data in the blockchain provably certifies the status of the consents. Moreover the compliance broker centralizes the inventory of which applications and processes depend on which consents, allowing for auditing the overall compliance status.

This paper is organized as follows. Section 2 briefly introduces core concepts about the blockchain technology, to make the paper self-contained. Section 3 presents our novel solution, detailing the protocol to create and rectify consents on the blockchain, discussing the strong integrity and confidentiality guarantees that characterize our protocol, defining the publish-subscribe mechanisms of our compliance broker to assist companies in addressing GDPR-compliance dynamically, and introducing a prototype implementation of our solution. Section 4 surveys the related work in the field. Section 5 summarizes our conclusions and plans for future research on these topics.

2. Basics on Blockchain Technologies

We can describe a blockchain as a decentralized, immutable, tamper-proof, public ledger [10]. At the very core, it consists of a simple data structure, organized as a list of chained blocks, and shared by all parties that participate in a peer-to-peer network. Each block represents a set of transactions of the ledger, and the transactions of all blocks, interpreted in the order corresponding to the sequence of the blocks, represent the status of the ledger. By adding new blocks (i.e., new transactions) to the blockchain, users can update the status of the ledger incrementally.

The most important characteristics of the blockchain technologies is that they do not require a central authority to guarantee the validity of the information stored by the peers. Rather, the peers that participate in the blockchain agree to a consensus protocol to validate the transactions in peer-to-peer fashion. The consensus protocol guarantees that, once a block is validated, all peers see that block in exactly the same status in their local copy of the blockchain, and cannot alter it [11]. Indeed, the solution that we present in this paper relies on the immutability of the data stored in the blockchain to guarantee that, once the consent records are stored in the blockchain, they represent durable information, and companies and data owners cannot disagree on the status of those consents at any moment in time.

Blockchains can be *permissionless* or *permissioned.* In the former case anyone can propose to add a transaction, and anyone can be a peer participating in the consensus protocol. Blockchains that store cryptocurrencies transactions are typically of this kind. On the other hand, *permissioned* blockchains are run by a *consortium* of peers – in our case, companies – that sustain the economic costs of running *validator nodes*, the peers that process transactions and store a copy of the blockchain. In this setting, each user has *credentials* that allow him to write (or even access/read) information to the blockchain. Usually no consensus algorithm is adopted, and the blockchain is used just as a distributed ledger in which every user writes information (that is, makes *public affirmations*) that cannot later be altered or deleted. A permissioned blockchain is used especially when the members of the consortium potentially have contrasting interests but they want to reach a common goal, and no member is trusted enough to run a centralized database.

Finally, an important aspect of modern blockchains is the use of *smart contracts*, that make possible some kinds of trusted interactions between users. Let us note, however, that in our solution we will not need to use smart contracts.

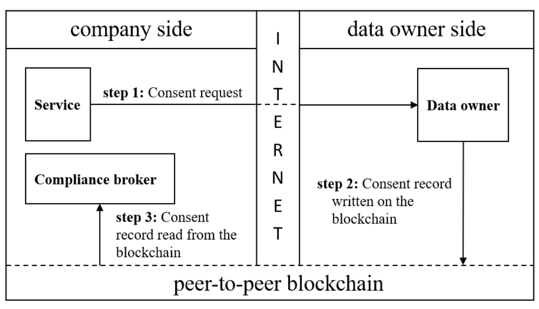
3. Approach

This section discusses the core components of our solution, i.e., a protocol to store and rectify consents on the blockchain, and a publish-subscribe architecture that allows companies to keep their services in synch with the evolving status of the consents.

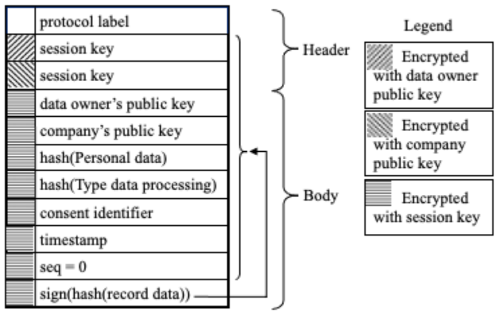
3.1. Granting a consent

We first introduce the operations of our protocol by which data owners can grant a company new consents on some personal data. Figure 1 (a) illustrates how the protocol proceeds to grant a new consent. The interaction starts with a company-side service (hereafter *the service*) that sends a consent request to a data owner (step 1), continues with the data owner that generates a *consent record* that encodes the consent in digital format and writes it on the blockchain (step 2), and completes with a dedicated company-side application, which we refer to as the *compliance-broker*, polling the blockchain to acquire the consent record (step 3). Below we explain each step, and we discuss the integrity and confidentiality guarantees that characterize the protocol.

The consent request (step 1) defines the personal data and the type of processing for which the consent shall be granted. Our protocol requires that the service shares these definitions with the data owner in the form of two digital files of any suitable file type. The personal data under concern may range from a specific datum or set of data, to categories of personal data on which the service requires consent once for all. Examples of specific data are family names, age information, email addresses, credit card data, specific images, documents and medical records. Categories of personal data could for example refer to geo-localisation data or other profiling



(a) Protocol interactions



(b) Consent record structure

Figure 1: Protocol to grant a new consent

information that the service will be collecting. With no loss of generality, we assume that this initial interaction might happen either in electronic fashion, e.g., via a Web page that the data owner visualizes during the subscription of an electronic service, or consists of a request sent via email, or might happen during an in-person session between the data owner and a company delegate, in case of services that consist partially or entirely of processes run by people.

When the data owner receives the consent request, he generates a digital record that encodes the consent, and performs a transaction to write it on the blockchain (Figure 1 (a), step 2). The data owner signs the data in the consent record, to guarantee the authenticity of the information in the record, and encrypts all data with the public key of the company, such that the compliance broker is the only other party that can access the information.

Figure 1 (b) shows the format of a consent record according to our protocol. This format aims to avoid attacks on the integrity and the confidentiality of the information in the record. The record consists of a header and a body section. The header section includes a label that identifies the protocol version and a session key, a binary string of 256 bits, which the data owner generates at random and encrypts in two copies, with his public key (first copy) and with the public key of the company (second copy), respectively. This guarantees that no third party who may intercept the record can extract the plain value of the session key. The data owner will then use the session key to encrypt all remaining data, i.e., the data in the body section of the record. Our solution relies on the availability of a public key infrastructure that allows data owners to certify the identity of the companies while using their public keys, and vice-versa.

The body section of the record includes the public keys of the company and the data owner, respectively, two (crypto-)hash values that characterize the specific consent that the record represents, an identifier of the consent, a timestamp, a sequence number set to zero, and the signature of the data owner. The consent record does not contain any personal data directly, but uses hash values to unambiguously identify the two documents that the data owner received at step 1, i.e., the documents that define the personal data and the type of data processing. The consent identifier links the record to the request that the data owner received from the service. The timestamp indicates the time at which the record is generated. The sequence number refers to the history of rectifications of the record: it is set to zero for the records that represent newly granted consents (like the record in Figure 1 (b)) and will be incrementally greater than zero for records that represent rectifications (as we discuss in the next section). The last datum in the record is the signature of the data owner, applied on the hash value computed on all other data in the record. The signature certifies the authenticity of the record, i.e., both the integrity of the information in the record and the identity of the the data owner. As said, all data in the body section of the record, including the signature, are further encrypted with the session-key.

The encryption of the record allows to the compliance-broker to validate the integrity of the record. In fact, upon reading the consent record from the blockchain (Figure 1 (a), step 3), the compliance broker can: (i) decode the session key with the private key of the company, (ii) authenticate the data owner based on the public key and the signature in the record, (iii) verify the integrity of the record by computing the hash value of the data and comparing it with the hash value decoded from the signature, (iv) confirm the integrity of the consent with respect to the relevant personal data and type of data processing by computing the hash values of those documents and comparing such hash values with the ones in the record, and (v) validate that the consent identifier, the timestamp and the sequence number are consistent. If all these checks succeed, the compliance broker approves the consent record and notifies the relevant services run by the company.

Without the ability to decode the session-key encrypted in the header section of the record, no information can be accessed by any third party, in case the record is read by other users of the blockchain. The protocol is also robust to reflection attacks[1], since the compliance broker will easily reveal (and discard) any duplicated record that refers to the same consent identifier of a previously received record, and has both the same timestamp and the same sequence number as the previous record.

3.2. Rectifying or Revoking Consents

The right of rectifying or revoking (right to be forgotten) personal data and consents are fundamental rights of the data owners according to the GDPR regulation (art. 16 - *Right to rectification*, art. 17 - *Right to erasure*). Our solution guarantees these rights by allowing data owners to unilaterally accomplish rectifications or revocations of any consent they have granted.

In our consent protocol, rectifying (or revoking) a consent simply requires the data owner to store on the blockchain a new consent record that i) is well formed, i.e., it has the structure that we presented in the previous section, and ii) is consistently related with the consent record to be rectified. For instance, the right part of Figure 2 exemplifies a well-formed consent record that consistently rectifies the consent record in the left part of the figure. The figure highlights that the two consent records have exactly the same structure, share some identical data (indicated with gray background in the figure), differ in either the hash value that defines the personal data or the type of data processing, include mutually related timestamps and sequence numbers, and are both internally consistent, i.e., they both include correct signatures of the data owner, and are both correctly encrypted with the session key available in the header section. In short, the rectifying consent record is consistent with the previous (rectified) consent record if it is correctly encrypted, correctly signed, maintains the constraints on the mutually identical and mutually related data as indicated in the figure, and is consistent with all previous rectifications

[1]Attacks that try to reuse an encrypted record after having intercepted it.

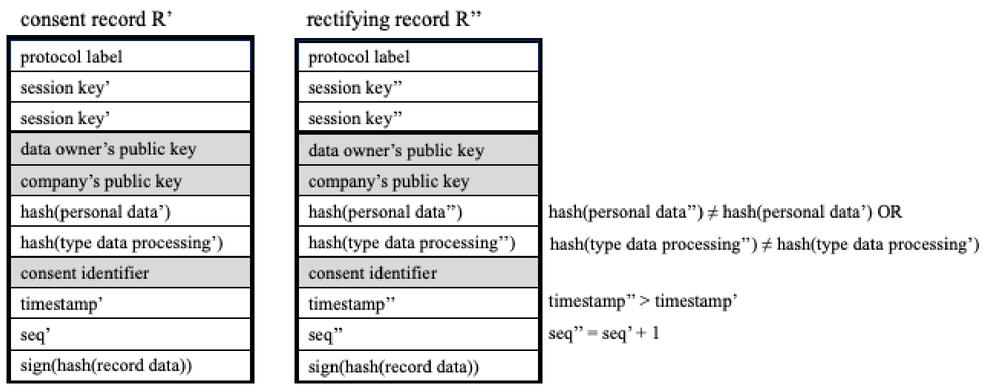


Figure 2: Relations between a rectified (R') and a rectifying (R") consent record

already in the blockchain. If so, the hash values of the personal data and type of data processing in the rectification record represent the new status of the corresponding consent.

Let us note that both the rectification operation and all the checks on the rectifying record could be performed by an open source API provided by the consortium that maintains the permissioned blockchain. Albeit not strictly necessary, this would greatly simplify the management of rectifications, simultaneously reducing the number of wrong rectification records written on the blockchain. Notice instead that these checks cannot be performed by a smart contract, as such a contract should know the secret keys of the company and of the data owner, a requirement that we deem excessive.

More formally, a rectifying consent record R'' is consistent with a previous (rectified) consent record R' if all the following conditions are met:

- The body section of R'' successfully decrypts by using the session-key decoded from the header section. This guarantees that both parties can decode the body section of the record R'', even if the data owner generated R'' unilaterally;
- The public keys and the consent identifier are identical in both consent records. This identifies the record R'' as a rectification of the consent represented by R';
- The rectifying record R'' has a later timestamp and a sequence number one-unit-higher than to R'. This chains R'' to R' within a sequence of rectifications of an original consent (the one with sequence number equal to zero) to which both records refer;
- The two records differ in either the hash of the personal data or the type of data processing. A differing hash of the personal data indicates a rectification of the personal data. A differing hash of the type of data processing indicates that the rectification is concerned with the type of data processing allowed for the (unchanged) personal data. In this latter case, the special value zero indicates a revocation action;
- The signature of the data owner in R'' is correct with respect to the data in R'';

- R' is not already consistently rectified by any other consent record that is stored between R' and R'' on the blockchain. This guarantees that there can be no valid duplicates in a chain of consistent rectifications.

According to the above properties, the status of a consent is the result of a chain of consistent rectifications of an original consent. Any rectification record that violates one or more of the above rules will be deemed invalid, and will be ignored. Thus, even though the protocol let data owners free to add new consent records as they like, only the valid rectification chains contribute to define the status of their consents, while any inconsistent record will be just ignored. The valid rectification chains stored in the blockchain unambiguously certify the latest status of each consent at any moment in time, and constrain the parties to comply with those consents. In case of litigation, either party can provide (e.g., to competent authorities) evidence of the status of the consents at a given moment by revealing the session keys of the valid chain of records, with no need to disclose their private key. If two parties report mutually contradictory pieces of evidence, the above properties unambiguously discriminate the correct one. Yet, companies can provably confute the validity of any consent record for which they cannot decode the same session key claimed by a data owner.

When the status of a consent changes on the blockchain, we leave for the company-side services the task of initiating interactions with the data owners, in order to get the rectified versions of the personal data or the rectified data processing information. Since when a rectifying record is filed on the blockchain, up to the moment in which they obtain the rectified data, those services must behave like if that consent was revoked. Similarly, if the new status of a consent does not meet their requirements, the services must behave like if there is no consent, and initiate procedures to inform the data owner of the issues. In the next sections, we present how our solution supports the ability of the services to promptly react to rectifications of consents, and we discuss possible design patterns to build applications that adapt to evolving consents.

3.3. Compliance with Dynamically Evolving Consents

Our solution allows services to remain in continuous compliance with the consents that undergo rectification actions. To this end, the compliance broker works in a publish-subscribe style to: (i) keep track of which services depend on which consents, (ii) maintain an updated inventory of the latest status of each consent, and (iii) notify the relevant services in the event of status changes related to any consents on which they depend.

Figure 3 illustrates the main interactions that lead the compliance broker track the dependence between a service and a consent (step 1: subscription), identify the change of status of the consent in the event of rectification actions (step 2: polling), and notify the service of a consent update (step 3: notification). Below we describe these steps in further detail.

The compliance broker records the dependency between a service and a consent immediately after completing the protocol steps that we described in Section 3.1, i.e., as soon as it stores a new consent record in the blockchain. In this phase, the compliance broker initialises the status of the new consent within its inventory as a triple $\langle id, h1, h2 \rangle$, where id is the consent identifier stored in the consent record, and $h1$ and $h2$ are the hash values of the personal data and purpose data that comprise the consent, respectively. Then, it accepts the target service as a subscriber

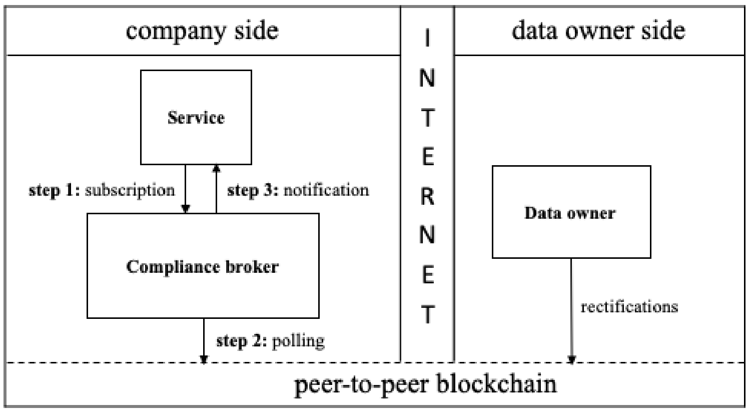


Figure 3: Notifying rectification actions to services

(Figure 3, step 1) who shall be notified of any change of that consent. The subscription includes a notification address, given as either a service API endpoint or an email address, according to whether the service is a software application or a human-driven process, respectively. In the event of a status change of the consent, the compliance broker will notify the target service by calling the API endpoint or sending an email, in either case, respectively.

The compliance broker identifies the status changes of the consents that undergo rectification actions (as we described in Section 3.2) by polling the blockchain regularly (Figure 3, step 2). At each new polling step, the broker scans the blockchain blocks that were added since the last polling step (as the previous blocks have not been altered, due to the immutability properties of the blockchain), matches the records that belong to our protocol by their label, decodes the third field of the header section to reveal the session key of the records that belong to the company, decodes the body section of those records with the session key, identifies the latest valid (cfr. Section 3.2) rectification of each consent, updates the status of the consent in its inventory accordingly, and notifies the subscriber services of any new consent status (Figure 3, step 3). To consistently update its inventory, the broker matches the consent identifiers in the rectifying records with the consent identifiers in the inventory.

3.4. Designing Compliance-Aware Services

We finally discuss the problem of designing services that are able to properly adapt their behavior upon being notified of changes of relevant consents. While diving into service-specific adaptations is admittedly out of the scope of this paper, we indicate a core set of design patterns and design principles that may generally apply to the services that participate in the publish-subscribe interactions of our solution.

Personal data registry Upon being notified of changes or revocations of relevant consents, the company services must retrieve the impacted personal data from the physical locations at which they are stored, aiming to execute suitable adaptation actions. For instance, if a revoked personal datum has been stored in a database, the service must recall the corresponding table and record, such that it can access and modify the datum. Thus, to support the execution of adaptation actions, service designs may include a *personal data registry* component that maps between consent identifiers and the physical locations of the corresponding personal data.

Personal data ping Services must be designed with suitable *personal data ping* functionalities, either scripted in their implementation or assigned to designated persons, to be able to retrieve new versions of the rectified personal data or rectified type-of-processing definitions, upon being notified of rectifications.

Independence between personal data and primary/foreign keys As a design principle, services shall not rely on personal data as unique identifiers of data structures, or to define relational associations between data structures. Otherwise, the rectifications or the revocations of the related consents would be at risk of jeopardizing those integrity constraints, up to causing service failures in the worst cases. For data stored in relational databases, this principle indeed corresponds to stating the *independence between the personal data and the primary or foreign keys* in the database.

Revocation-aware functionalities Any service functionality that depends on personal data shall be designed with careful consideration of the possible revocation of the consents on those data. Revocations may particularly impact many common functionalities that define stateful behaviors based on the personal data. In general, the impacted functionalities must include proper (possibly degraded) behaviors to robustly cope with the revoked data.

Personal data objects A specific solution for coping with revoked data is to anonymize (rather than deleting) those data. In particular, this solution suites for the applications in which deleting the personal data may contrast with some integrity constraints of their implementation. To this end, a possible design choice is to mediate the access to the revocable data with objects (to which we refer as *personal data objects*) that provide suitably paired *forget-data* and *get-data* operations: The forget-data operations delete the data by replacing them with anonymized counterparts; The get-data operations either retrieve the actual data if these were not revoked, or render the anonymized data with anonymous data in proper format to comply with the business rules of the considered service.

3.5. Prototype

As a proof-of-concept demonstrator, we implemented our solution on top of a custom instance of the Ethereum blockchain created with Ganache [12]. We used the online Remix IDE [13] to deploy smart contracts on the blockchain, and the Node.js runtime engine and web3.js library to create the back-end and the front-end of all the user and company functionalities, including the execution of the hashing, encryption and decryption algorithms. Our demonstrator includes a client that allows data owners to produce the consent records according to the protocol of our solution, an implementation of the company-side compliance broker, and other clients to visualize the status of the consents either granted by a given data owner to any company, or granted to a given company by any data owner. The sources of our demonstrator can be found on GitHub [14].

4. Related Work

We have already mentioned other approaches that exploit blockchain technologies for fostering compliance to GDPR [7, 8, 9, 6]. Our approach has several distinctive points of novelty over these previous pieces of research. First, our approach is unique in guaranteeing that no information on the granted consents is disclosed to the other users of the blockchain but the interested parties. Second, no previous approach copes with keeping the services and the processes of a company in-synch with consents that may evolve dynamically under the control of the data owners.

Some applications provide their own functionalities to let users express consents or remove their personal data. In particular this is the case of many social media applications. For example, Google provides in every user account a dashboard that lets users express consents and eliminate their data, e.g., data related to email management, geo-localization information, web activity, phone activity, and so forth. Notably, the application Suicide Machine [15] works as a gateway to the data-removal functionalities of the social media platforms, like Facebook, Twitter, or LinkedIn, and allows users to remove their profiles from a set of platforms at once. The downside is that these solutions are application-specific, and generally apply only for fully-automatic, fully-online services. Relying on application-specific solutions does not allow for companies to audit the compliance status of all their personal-data-driven processes, or users to directly monitor the status of all their consents across all companies.

Another body of research proposals enforce limited time of validity of the personal data released to companies, to guarantee that companies cannot retain the personal data forever [16, 17, 18]. Technically, these approaches work by maintaining the personal data in encrypted format, ensuring that the decryption keys expire after a predefined time: after the expiration of the keys, the data cannot be read any longer, and the companies must ask the data owners to grant access to their personal data once again. Expiring decryption keys could be used also in our solution, but an alternative that is probably easier to manage is simply to add an expiration date field in the consent record; when such a date is reached, if the consent is needed to perform some operation it should be requested again to the data owner. Expiring decryption keys are also used by Xpire [19], that allows the users of social media platforms to create self-destroying contents. These approaches mitigate the risk of the data owners to loose control on their data, but do not let them directly control, modify or revoke their consents, which is a primary goal of our approach.

5. Conclusions

The GDPR regulation commits companies to architect their services with maximum care in complying with the users' consents on any personal data that the services may require from them. Moreover, according to the GDPR, it is of paramount importance that companies guarantee the right of their users to rectify and revoke their consents at any time, ensuring prompt reactions to comply with any change of those consents.

In this paper we presented a novel solution that exploits a peer-to-peer blockchain to store and maintain the consents that users grant to the companies, and provides a secure protocol by which users can autonomously operate on the blockchain to rectify and revoke any of the consents that they granted. Our solution is specifically designed to guarantee full privacy of the information stored in the blockchain, meaning that only the interested parties can know the status of the consents that are relevant to them, while any third party that has access to the blockchain cannot infer any meaningful information on those consents. At each company, our solution settles a compliance broker application that pools the blockchain regularly to identify any change of status of the consents granted to the company. Upon identifying relevant changes, the compliance broker works in a publish-subscribe style to notify the impacted services, thus allowing them to implement timely reactions to the consent changes.

In the future we aim to extend our current prototype to work with other features, such as the automatic expiration of consents. Further, we plan to work on the current implementation of our solution [14], which is currently in an embryonic state, both at the blockchain side and concerning the implementation of the company services described in Section 4.3.

References

[1] European Commission, General data protection regulation (GDPR), https://gdpr.eu/tag/gdpr/, 2018.

[2] P. Voigt, A. v. d. Bussche, The EU General Data Protection Regulation (GDPR): A Practical Guide, 1st ed., Springer Publishing Company, Incorporated, 2017.

[3] Cliqz GmbH, Ghostery, https://www.ghostery.com/, Last visit: February 6, 2021.

[4] OneTrust LCC, Onetrust privacy security and governance, https://www.onetrust.it/, Last visit: February 6, 2021.

[5] Absolute Software Corporation, Absolute, https://www.absolute.com/, Last visit: February 6, 2021.

[6] G. Zyskind, D. Zekrifa, P. Alex, O. Nathan, Decentralizing privacy: Using blockchain to protect personal data, in: 2015 IEEE Security and Privacy Workshops, 2015, pp. 180–184. doi:10.1109/SPW.2015.27.

[7] C. Wirth, M. Kolain, Privacy by blockchain design: A blockchain-enabled GDPR-compliant approach for handling personal data, in: Proceedings of 1st ERCIM Blockchain Workshop 2018, 2018, pp. 1–7. doi:10.18420/blockchain2018_03.

[8] J. C. Vargas, Blockchain-based consent manager for GDPR compliance, in: Open Identity Summit, 2019, pp. 165–170. URL: https://dl.gi.de/bitstream/handle/20.500.12116/20985/proceedings-14.pdf.

[9] F. Guggenmos, J. Lockl, A. Rieger, A. Wenninger, G. Fridgen, How to develop a gdpr-compliant blockchain solution for cross-organizational workflow management: Evidence from the german asylum procedure, in: 53rd Hawaii International Conference on System Sciences, HICSS 2020, Maui, Hawaii, USA, January 7-10, 2020, ScholarSpace, 2020, pp. 1–10. URL: http://hdl.handle.net/10125/64234.

[10] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, http://www.bitcoin.org/bitcoin.pdf, 2008.

[11] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, C. Yang, The blockchain as a decentralized security framework [future directions], IEEE Consumer Electronics Magazine 7 (2018) 18–21. doi:10.1109/MCE.2017.2776459.

[12] ConsenSys Software Inc., Ganache, Truffle suite, https://www.trufflesuite.com/ganache, Last visit: February 19, 2021.

[13] Open source, Remix, https://remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.6.6+commit.6c089d02.js&runs=200, Last visit: February 19, 2021.

[14] M. Calani, Progetto ITASEC 2021, https://github.com/Massi1994/Progetto_ITASEC, 2021.

[15] W. Rotterdam, Web 2.0 suicide machine, http://suicidemachine.org/, Last visit: February 6, 2021.

[16] R. Perlman, File system design with assured delete, in: Proceedings of the Third IEEE International Security in Storage Workshop, SISW '05, IEEE Computer Society, USA, 2005, p. 83–88. doi:10.1109/SISW.2005.5.

[17] L. J. Bannon, Forgetting as a feature, not a bug: The duality of memory and implications for ubiquitous computing, CoDesign 2 (2006) 3–15. doi:10.1080/15710880600608230.

[18] R. Geambasu, T. Kohno, A. A. Levy, H. M. Levy, Vanish: Increasing data privacy with self-destructing data, in: F. Monrose (Ed.), 18th USENIX Security Symposium, Montreal, Canada, August 10-14, 2009, Proceedings, USENIX Association, 2009, pp. 299–316. URL: http://www.usenix.org/events/sec09/tech/full_papers/geambasu.pdf.

[19] M. Cuban, Xpire, http://www.getxpire.com/xpireApp/, Last visit: February 6, 2021.