Hey, my data are mine! *Active* data to empower the user

Anonymous authors
xx
xx, xx
XX@XX.XX

ABSTRACT

Privacy is increasingly getting importance in modern systems. As a matter of fact personal data are out of the control of the original owner and remain in the hands of the software-systems producers. In this new ideas paper, we drastically change the nature of data from passive to active as a way to empower the user and preserve both the original ownership of the data and the privacy policies specified by the data owner. We demonstrate the idea of active data in the mobile domain.

ACM Reference Format:
Anonymous authors. 2020. Hey, my data are mine! *Active* data to empower the user. In *Proceedings of ICSE '20: International Conference on Software Engineering (ICSE '20).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Nowadays, we are increasingly interacting and cooperating with systems like mobile devices, smart watches, (autonomous) cars, service robots, and so on. An average user of technology spends one day per week with her devices and a constant use of internet as reported by Ofcom [3]. We are also increasingly aware of the risks and side effects of sharing our personal data. The EU GDPR legislation for data protection [2] contributed to create such awareness and introduced regulatory constraints to protect citizens. Europe is at the forefront of the regulation and reflections on these issues through its institutional bodies [22].

However, in order to exploit the benefits and opportunities of the digitalization, we are obliged to share some of our personal data. The boundaries of this tradeoff are not well defined and citizens are passive consumers: once on the network, personal data are out of the control of the original owner and remain in the hands of the (software-) systems producers.

As a result, privacy and trustability of (software) systems are becoming fundamental aspects to be considered, particularly with regard to meeting the expectations of end users. Various approaches have been proposed to specify what can be done with data [26], how to enforce privacy concerns [35], how to manage access control policies [26], how to deal with transparency and accountability criteria in software development, e.g., [1, 4, 19, 22, 27], and so on. Unfortunately, these approaches may only partially solve the data management issue. The protection of data confidentiality is achieved through data encryption techniques [17]. Even though encryption is a must-have element in any security strategy, it is not enough when the need is fine-grained governed data usage. Besides encryption, current practices for managing personal information neglect the rights of the user to fully express her desires and exert her control on how and by whom her data are used, as testified in this study made in mobile apps domain [32]. According to a survey conducted among adult Americans [23], 91% of participants believe that consumers have lost control over how personal information is collected and used by companies, and that most participants would like to do more to protect their personal information.

So far, data have always been considered as passive entities carrying digitized information on which operations can automatically be performed by computer machines. The logic that controls the life-cycle (e.g., creation, manipulation and destruction) of data is decoupled from the data themselves, and the owner of the data often loses control over her data. In this paper, we propose to drastically change the nature of data from passive to active, by introducing the concept of active data. This technology empowers and protects citizens and their personal data. An active data object is a software module that wraps, encapsulates, and protects personal data. Active data mediate the access to personal data via well-defined interfaces, forbid any different and direct access while preventing any unauthorized use. To prevent unauthorized accesses to personal data locally stored we rely on encryption. Moreover, we increase the security by use of secret sharing techniques as explained in Section 3. Moreover, active data embody monitor and enforcer technology to both guarantee the preservation of privacy policies specified by the owner, and enforce actions that are needed in order to satisfy her privacy desiderata.

2 STATE OF THE ART AND RELATED WORKS

As exhaustively analyzed in [9], typically, systems make choices on behalf of the user without caring about the user desiderata.

The specification of privacy preferences has been largely studied at various levels, from requirements specification to coding [31, 36]. The Privacy-by-Design approach permits to design privacy-preserving systems by adhering to certain principles provided by high-level guidelines [5, 6, 8, 26]. The core idea behind these principles is that data protection should be proactive, i.e., capable of acting before any issue arises [13].

The work in [25] proposes an approach based on sticky policies, which enables users to control how their data are handled and shared. Specifically, the users can express their policies, which

ICSE '20, May 23–29, 2020, Seoul, South Korea

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

describe how data should be treated, and they are attached to the data as conditions and constraints.

Privacy in mobile apps is a central aspect to be considered especially when concerning the expectations of end users [36]. Due to the rigidity of the permission models adopted by the different mobile platforms, e.g., Android by Google and iOS by Apple, end users are confined into a secondary role, having the only option of choosing between either privacy or functionalities, as desirable trade-offs are not allowed. The work in [32] proposes Android Flexible Permissions (AFP), a user-centric approach that empowers end users to specify and customize fine-grained permission levels according to their own subjective privacy concerns. Authors in [33] conducted a large-scale empirical study to investigate how end users perceive the new run-time permission system of Android; results suggest that privacy-related concerns are widespread.

Major parts of the GDPR require systems to account for how personal information, and any information derived from it, moves and get stored within computer systems. For instance, it demands to keep track of with whom data get shared (article 15(1)(c)) as well as to identify *all* data associated to or deriving from individuals (article 15(3)). Information-Flow Control (IFC) [30] is a promising technology for the active data management. IFC permits to obtain guarantees of many of the GDPR requirements related to how private information gets handled and disseminated within systems. Moreover, the *right to be forgotten* (article 17) by erasing subjects data can also be formulated as an IFC problem [21].

In recent years, the Diaspora [12] and Mastodon [38] social networks have emerged, both based on the idea of adopting decentralization to protect users' privacy. Rather than using a centralized architecture, resulting in users' data being in the hands of a single entity, they rely on a decentralized network of independent, federated servers that are administrated by individual users. End-users can choose which servers to connect to and their data is shared exclusively with the selected ones.

Blockchains might be employed to secure personal data against tempering and revision [37, 39]. A blockchain consists of data-structure blocks stored in a decentralized architecture consisting of an unbounded number of nodes. Whenever new transactions occur, the consensus of the part of the network that holds the majority of some relevant resource that limits the production of blocks is needed. Each transaction is verified by the network and if a node attempts to cheat the system, it can be easily identified. A blockchain might be seen as an append-only database, which provides users with several data protection properties including immutable data storage, and secure time-stamping. The data immutability characteristic of blockchain technologies put them in collision with the right to be forgotten pillar of GDPR. To reconcile such an idiosyncrasy, a viable solution would be to encrypt personal data before writing it to a blockchain. Thus, the right to be forgotten can be applied by destroying the keys that are needed to make data readable again. Even though this represents a viable solution from a technical point of view, regulators should accept that destroying keys actually represents data erasures for the purpose of the GDPR [7].

In [20], authors propose Vanish, an approach aiming at protecting the privacy of archived data against accidental, malicious, and legal attacks: cryptographic techniques ensure that all the copies of certain data become unreadable after a user-specified time.

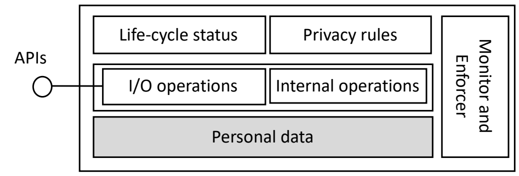

Figure 1: Active data module

The work in [28] makes data unrecoverable after a given expiration date. Instead of destroying data, the proposed approach destroys when needed data encryption and decryption keys.

3 ACTIVE DATA

As anticipated in the introduction, we aim at changing the nature of data from passive to active. The idea is to encapsulate, wrap, and protect personal data inside an "active data module" (see Figure 1).

The module encloses personal data after being suitably encrypted with state-of-the-art encryption techniques. The module offers I/O operations that provide mediated access to the personal data when, e.g., making a copy, modifying, and sharing. Internal operations serve to actually operate on the personal data, from creation to destruction, to usage. The privacy rules defined by the owner of the personal data are then evaluated according to the information available in the life-cycle status. Rules are defined in HyperLTL [15] since, differently from traditional specification, capturing privacy policies requires logic that is capable to relate many execution traces of software [16]. We are working on providing user-friendly and easy ways for specifying privacy policies in a correct way, e.g. by exploiting the idea of property specification patterns [10, 18]. The life-cycle status component contains variables to keep trace of the life cycle of the active data, e.g., number of data visualizations, number of copies of the data, accessibility right, creation and expiration dates, origin of the data (art. 15(1)(g) of [2]), but also where it can safely flow and be shared (art. 15(1)(c) of [2]). The status may also contain information about the context of use, such as the location-based information where data are accessed, the device that is used to read the data and, in general, any information that allows the run-time evaluation of privacy preferences. Status information that has to be shared and synchronized among multiple instances of the same data (e.g., number of existing copies) is stored remotely, while locally preserving a logical link to it. This concept will be clearer in the following when we will explain how we keep trace of the copies, sharing, etc. The satisfaction of the privacy rules is guaranteed by the monitor and enforcer components, which continuously check and update the life-cycle status to detect and possibly solve problems before privacy violations.

In order to control the access and to manage the life-cycle of all the copies of the personal data that have been created also via sharing, e.g., in social networks, we introduce the concept of *active data network*. An active data network is a graph that is created when an owner of a personal data decides to protect and control the data by encapsulating it inside an active data module. For instance, in the mobile domain this would be done through a dedicated Active Data app, as described in Section 4. A node consisting of an instance of the active data module in Figure 1 is created and this node plays the primary role of *owner node*, i.e., it is the root of the just created personal data flow. As shown in Figure 2, an active data network is

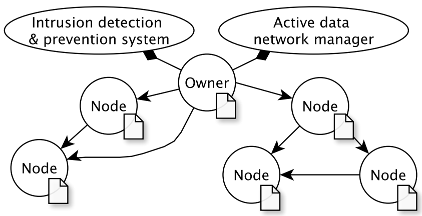

Figure 2: A box and line view of the architecture and its components.

a *hierarchical P2P network*. When an active data module is shared, a new instance of the active data module is created and connected to the active data network by following a parent-to-child relationship.

The owner node acts as:

- *Intrusion detection & prevention system* – It implements an intrusion detection and prevention system to monitor the active data network activity and detect possible intrusions. It uses network behaviour analysis to monitor inbound and outbound activities to protect the peers from attacks [29].
- *Active data network manager* – It keeps trace of its own active data network and enables the update of privacy policies at any moment. The policies are distributed to the entire active data network. Thanks to the P2P architecture, the owner node does not need to be always available and connected.

Similarly to public blockchain consensus protocols [37, 39], within and across active data networks reaching the consensus prevents any node from controlling or derailing the whole network. Differently from blockchain, we always keep the owner node of the data, which is the only one with the right to define and/or change the privacy rules. Any node, upon performing an operation, needs to check the status with the network, updates will be propagated, and inconsistencies would lead to a temporary block of the active data network, waiting for the owner node to take suitable actions. This also means that, similarly to private blockchain consensus protocols [37], the owner node has in fact the right for the final decision. In this respect, it is worth clarifying that the excerpt in Figure 2 only depicts a logical view for what concerns tracking the ownership and (hierarchical) sharing of the data. Instead, the underlying peer-to-peer network is overall strongly connected so that not only each node in the active data network can take part in the consensus process, but also any node in the whole peer-to-peer network. Clearly, only the nodes in their active data network are responsible for validating any operations upon the respective data.

The consensus protocol must be secured so to block attempts to violate policy rules and to tamper with the personal data and their state. In order to make the consensus protocol secure, it must be fault tolerant. We plan to adopt a Byzantine consensus protocol for distributed and decentralized systems capable of handling fault tolerance, such as Practical Byzantine Fault Tolerance [37].

The active data network enables the owner to delete all the copies of her active data (and the embedded personal data), at any time she decides to enforce the right to be forgotten. In fact, the enforcers of all the active data modules, by following the hierarchical parent-to-child flow(s) imposed by the network, will order the self-destruction of all of the active data copies by exploiting Internal operations (see Figure 1). The owner node will self-destroy only after all of her child nodes are self-destroyed. Obviously, it is impossible to force the immediate destruction of an active data node that is offline: it will be destroyed as soon as back online. At the moment, we rely on encryption of the personal data as the main mechanism to protect from hacking of the active data, for instance when the node hosting the active data will be offline.

Secret sharing of active data – We exploit the idea of secret sharing [34] to enhance security by (i) cutting personal data into *shares*, (ii) performing *encryption* and (iii) *distributing* those encrypted shares to various peers. Original data can then be reconstructed only when a sufficient number of shares are recombined together.

Indeed, the hierarchical P2P network in Figure 2 is inspired by the *one dealer and n players* secret sharing scheme [11], and the secret sharing technique can be used to protect both keys and sensitive data in a blockchain [14, 24].

The undesired consequence here is that it is not possible to access the data if the required number of peers is not on line. This goes with the more general problem of network volatility, notably, in mobile networks. In order to mitigate this problem, our idea is to create a number of replicas of each single share in order to avoid single point of failures while augmenting the access probability.

Observations – In general, we cannot prevent or control replication of data outside of the active-data network through usage of "external" means like an external camera. Although specific cases like taking a screenshot can be intercepted and forbidden, others remain uncontrolled (after all, when the personal data is a password, a human can just remember it and share). Still, the replicated data will be a different data (not anymore an active data) having, e.g., different format, qualities, and rendering. Moreover, depending on the nature of the data (especially those not requiring rendering), replication by external means might be impossible and not easily and effectively reusable, e.g., replication of a software package.

4 ACTIVE DATA IN THE MOBILE DOMAIN

This section describes a scenario that illustrates active data at work in the mobile domain. For the purpose of this paper, we decided to target the mobile domain since, following the issues discussed in Section 1, in this domain active data will bring clear and tangible benefits to end users. That being said, one way to introduce active data in this domain is to provide end users with an *Active data app* capable of transforming any data they might want to share from their phone into its active counterpart. Fundamentally, the app is a container for active data; it also offers intrusion detection & prevention while managing the data flow(s) network for those data it is the owner of.

Alice wants to share `MyFile.txt` with Bob. She opens the Active-Data app on her smartphone and loads the file (label 1 in Figure 3), with the intent of creating a new active data. Being concerned about her privacy, during the active data creation process, she decides to set a rule for stating that her file can only be opened by Bob (2). An active data object `MyFile.active` is created by active data app, embedding into it `MyFile.txt` and the privacy rules she specified. The active data object is given back to Alice (3), in turn identified as the owner node of it. Alice can now share the active data object with Bob (4) over conventional communication channels (e.g., text messaging, e-mail). Upon receiving the active data object, Bob opens it in his Active Data app (5). The app then establishes

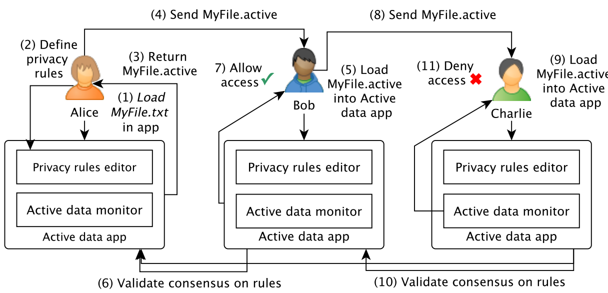

Figure 3: Scenario describing active data at work in the mobile domain

communication with the other nodes in the active data network, currently represented only by Alice (6). Each involved node verifies the validity of Bob's access attempt against the privacy rules. In the case of reaching a consensus on the validity of Bob's request, the status of the active data object is updated and Bob can successfully view Alice's personal data (7).

The day after, Bob forwards the active data object to Charlie (8), and it is loaded into his Active Data app (9). Again, the Charlie's app establishes a communication with the other nodes currently in the active data network – now both Alice and Bob (10). Since the consensus on the validity of Charlie's access attempt is not reached, his request is denied (11), and the received active data object is automatically destroyed.

5 CONCLUSIONS AND FUTURE WORKS

Current work focuses on prototyping a first complete version of active data in the mobile domain to calibrate the amount of runtime support needed to provide the level of protection implied by the user defined rules. Indeed, it is possible to categorize active data depending on the amount of information that constitutes the active-data status on which privacy rules predicate. Moreover, we will perform extensive evaluation on the performance and scalability of the approach.

REFERENCES

[1] 2018. https://standards.ieee.org/develop/indconn/ec/autonomous_systems.html.
[2] 2018. EC. General Data Protection Regulation.
[3] 2018. Ofcom Communications Market Report 2018. https://www.ofcom.org.uk/__data/assets/pdf_file/0028/155278/communications-market-report-2019.pdf.
[4] 2018. Partnership on AI. https://www.partnershiponai.org/.
[5] (Last access January 2020). DECODE. https://decodeproject.eu.
[6] (Last access January 2020). IRMA. https://privacybydesign.foundation/irma-en/.
[7] (Last access January 2020). Right to be forgotten GDPR vs Blockchain. https://archer-soft.com/en/blog/right-be-forgotten-gdpr-vs-blockchain-technology.
[8] (Last access January 2020). Vision. http://www.visioneuproject.eu/.
[9] A. Acquisti et al. 2017. Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online. *ACM Comput. Surv.* 50, 3, Article 44 (2017).
[10] M. Autili, L. Grunske, M. Lumpe, P. Pelliccione, and A. Tang. 2015. Aligning Qualitative, Real-Time, and Probabilistic Property Specification Patterns Using a Structured English Grammar. *IEEE TSE* 41, 7 (July 2015), 620–638.
[11] A. Beimel. 2011. Secret-Sharing Schemes: A Survey. In *Coding and Cryptology*.
[12] A. Bielenberg, L. Helm, A. Gentilucci, D. Stefanescu, and H. Zhang. 2012. The growth of diaspora-a decentralized online social network in the wild. In *2012 Proceedings IEEE INFOCOM Workshops*. IEEE, 13–18.
[13] A. Cavoukian. 2009. Privacy by Design. *Ottawa: Information and Privacy Commissioner of Ontario, Canada* (2009).
[14] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song. 2019. Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts. In *EuroS P*.
[15] M. R. Clarkson, B. Finkbeiner, M. Koleini, K. K. Micinski, M. N. Rabe, and C. Sánchez. 2014. Temporal Logics for Hyperproperties. In *POST 2014*. 265–284.
[16] M. R. Clarkson and F. B. Schneider. 2010. Hyperproperties. *Journal of Computer Security* 18, 6 (2010), 1157–1210.
[17] P. Dixit, A. K. Gupta, M. C. Trivedi, and V. K. Yadav. 2018. Traditional and Hybrid Encryption Techniques: A Survey. In *Networking Communication and Data Knowledge Engineering*.
[18] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett. 1999. Property specification patterns for finite-state verification. In *ICSE99*. ACM Press, 411–420.
[19] J. Larus et al. 2018. When Computers Decide: European Recommendations on Machine-Learned Automated Decision Making. http://www.acm.org/binaries/content/assets/public-policy/ie-euacm-adm-report-2018.pdf.
[20] R. Geambasu, T. Kohno, Amit A. Levy, and H. M. Levy. 2009. Vanish: increasing data privacy with self-destructing data. In *USENIX security*. 18.
[21] S. Hunt and David Sands. 2008. Just Forget it – The Semantics and Enforcement of Information Erasure. In *ESOP 2008 (LNCS)*. 239–253.
[22] P. Inverardi. 2019. The European Perspective on Responsible Computing. *Commun. ACM* 62, 4 (March 2019), 64–64.
[23] M. Madden, L. Rainie, K. Zickuhr, M. Duggan, and A. Smith. 2014. Public perceptions of privacy and security in the post-Snowden era. *Pew Res. Center* (2014).
[24] S. K. D. Maram, F. Zhang, L. Wang, A. Low, Y. Zhang, A. Juels, and D. Song. 2019. CHURP: Dynamic-Committee Proactive Secret Sharing. In *SIGSAC*.
[25] M. Mont and S. Pearson. 2011. Sticky Policies: An Approach for Managing Privacy across Multiple Parties. *Computer* 44, 09 (sep 2011), 60–68.
[26] OASIS. 2016. Privacy Management Reference Model and Methodology (PMRM).
[27] High-Level Expert Group on Artificial Intelligence (AI HLEG). access Sep. '19. The Ethics Guidelines for Trustworthy Artificial Intelligence (AI).
[28] R. Perlman. 2005. The Ephemerizer: Making Data Disappear. *Journal of Information System Security* 1 (2005), 51–68.
[29] Qi Zhang R. Janakiraman, M. Waldvogel. 2003. Indra: a peer-to-peer approach to network intrusion detection and prevention. *WETICE 2003*. (11 Jun 2003).
[30] A. Sabelfeld and A. C. Myers. 2003. Language-Based Information-Flow Security. *IEEE J. Selected Areas in Communications* 21, 1 (Jan. 2003), 5–19.
[31] A. De Salve, P. Mori, and L. Ricci. 2018. A survey on privacy in decentralized online social networks. *Computer Science Review* 27 (2018), 154 – 176.
[32] G. L. Scoccia, I. Malavolta, M. Autili, A. Di Salle, and P. Inverardi. 2019. Enhancing Trustability of Android Applications via User-Centric Flexible Permissions. *IEEE Transactions on Software Engineering* (2019).
[33] G. L. Scoccia, S. Ruberto, I. Malavolta, M. Autili, and P. Inverardi. 2018. An Investigation into Android Run-time Permissions from the End Users' Perspective. In *MOBILESoft 2018*.
[34] A. Shamir. 1979. How to Share a Secret. *Commun. ACM* 22, 11 (1979), 612–613.
[35] B. Shishkov and M. Janssen. 2018. Enforcing Context-Awareness and Privacy-by-Design in the Specification of Information Systems. In *Business Modeling and Software Design*.
[36] I. Wagner and D. Eckhoff. 2018. Technical Privacy Metrics: A Systematic Survey. *ACM Comput. Surv.* 51, 3, Article 57 (2018), 57:1–57:38 pages.
[37] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang. 2017. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In *BigData Congress*.
[38] M. Zignani, S. Gaito, and G. P. Rossi. 2018. Follow the "Mastodon": Structure and Evolution of a Decentralized Online Social Network. In *AAAI Conference on Web and Social Media*.
[39] G. Zyskind, O. Nathan, and A. Pentland. 2015. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In *IEEE Security and Privacy Workshops*.