SAFETHINGS: Data Security by Design in the IoT (Position Paper)

Manuel Barbosa[†], Sonia Ben Mokhtar[‡] Pascal Felber*, Francisco Maia[†], Miguel Matos[¶]
Rui Oliveira[†], Etienne Riviere*, Valerio Schiavoni*, Spyros Voulgaris[§]
*University of Neuchâtel, Switzerland.
[†]INESC-TEC & University of Minho, Portugal.
[‡]CNRS & University of Lyon, France.
[§]University of Patras, Greece.
[¶]INESC-ID & University of Lisbon, Portugal.

***Abstract*—Despite years of research and the long-lasting promise of pervasiveness of an "Internet of Things", it is only recently that a truly convincing number of connected things have been deployed in the wild. New services are now being built on top of these things and allow to realize the IoT vision.**

However, integration of things in complex and interconnected systems is still only in the hands of their manufacturers and of Cloud providers supporting IoT integration platforms. Several issues associated with data privacy arise from this situation. Not only do users need to trust manufacturers and IoT platforms for handling their data, but integration between heterogeneous platforms is still only incipient.

In this position paper, we chart a new IoT architecture, SAFETHINGS, that aims at enabling data privacy by design, and that we believe can serve as the foundation for a more comprehensive IoT integration. The SAFETHINGS architecture is based on two simple but powerful conceptual component families, the *cleansers* and *blenders*, that allow data owners to get back the control of IoT data and its processing.

***Keywords*-Internet of Things, Data privacy, Distributed Systems**

I. INTRODUCTION

The promise of the Internet of Things (IoT for short) has been fueling intense research and industrial development for several years now. Yet, the realization of the vision of the IoT as an ubiquitous, interconnected, and integrated system is far from being fully realized. At the same time, the development of connected "things" has never been so prolific. A study projects, for instance, that each person will have on average 26 interconnected "things" by 2020 [18]. Such a significant number of connected devices creates new opportunities for innovative applications but there are also a number of challenges that must be addressed in order to unlock this potential. We believe that such a critical aspect lies in *data management*. In particular, this aspect pertains to how data is *exchanged* between devices (communication), how data is *gathered* (storage) [22], and how data is *processed* (analytics). Further, addressing correctly two facets of data management is crucial to the adoption of IoT by society, that is, how data is kept *safe* (security) and how it remains in possession or *control* of its rightful owner (data privacy). Although all these aspects are well studied individually, their integration and composition in frameworks and runtimes taking into account the dynamics and scale inherent to the envisioned IoT deployments remains a challenge.

Collaborative data processing in IoT scenarios is still typically addressed in an ad-hoc way, and a general solution to the problem of IoT data management is yet to be found. If we add privacy and security concerns to the equation, things become even more challenging: trust management, authentication, key management, data usage control, incident response, etc., are a few of the aspects of information security and privacy that become extremely hard to implement correctly for a heterogenous and highly dynamic set of participants. One crucial stepping stone towards tackling these challenges is to consider security and privacy in the development of IoT systems from the very beginning, to ensure what is usually referred to as *security by design* [1], [2].

Our proposed architecture, SAFETHINGS, is designed to be flexible enough to allow the integration of current technologies, while being incisive enough to integrate data communication, data storage, data processing, and data security by design. This is built around two simple but powerful concepts that we introduce in this paper: *data cleansing* and *data blending*. Data cleansing refers to data filtering and transformation technologies that enable data producers or (partially) trusted data processors to enforce data sanitisation at trust boundaries, in order to enable secure and controlled data flows. Data blending refers to data analytics techniques applied on data that has been previously partially or fully cleansed. Additionally, we consider data exchangers which are communication middleware components responsible for the data flow between cleansers and blenders. We show how these frugal concepts can enable a flexible IoT architecture where data privacy is a first-class citizen.

978-1-5386-0602-5/17 $31.00 © 2017 IEEE
DOI 10.1109/EDCC.2017.33

II. Related Work

In the context of IoT, traditional problems related to data storage, data processing, and data security become even more challenging due to the intrinsic heterogeneity and dynamism of the considered environment [7].

Currently, the largest deployments of storage systems [19] rely on peer-to-peer (P2P) approaches to provide membership maintenance and data assignment, for instance resorting to Distributed Hash Tables (DHT) [16].Nevertheless, these protocols assume a moderately stable system and current implementations of DHTs struggle with churn rates observable in real workloads [15]. This challenge suggests that the world of IoT requires novel approaches to data storage.

Storing data in third-party IoT devices and back-end Cloud services raises privacy concerns that must be addressed. Previous work protects sensitive information by focusing on the *hiding-in-a-crowd* principle [10], which motivates both anonymisation [17], and differential privacy [8] techniques. The authors of [21] survey how to achieve location privacy. Another approach is to add noise to location data. Differential privacy has been successfully applied to location data [3]. However, finding the appropriate level of protection is difficult as stronger protection levels (e.g., noise) may degrade data quality to a point where it can no longer be exploited.

Besides protecting sensitive information, it is also important to protect the information leaked while performing analytics over the data [20].Traditional techniques rely on nodes with reasonable computational power and communication capabilities, such as secure multiparty computation, homomorphic encryption, coding, etc [11], [23]. Analytics over private data has also been explored in Hub-of-all-Things [3], which envisions data privacy in an IoT personal data market where users can trade and exchange their personal data, but does not investigate advanced privacy-preserving analytics. Finally, in User-Centric-Networking [6] the authors proposed a personal information hub, where contextual data collected from various devices is used to offer recommendation services based on homomorphic schemes [9].

SafeThings is a proposal for a unified architecture where these different approaches, until now used independently to solve specific challenges, can coexist and cooperate with each other. The proposed architecture is modular, to let choose the best techniques according to the trust models and privacy requirements of underlying applications.

III. Architecture

The goal of this position paper is to chart the foundations of an architectural design able to support a new generation of IoT applications. This architecture allows *things* to share and/or analyze data to provide integrated services to their users with fair data management principles. The main challenges in this scenario are i) the integration between things themselves, and ii) between things and applications, iii) data security and privacy, and iv) the need to support legacy devices and applications.

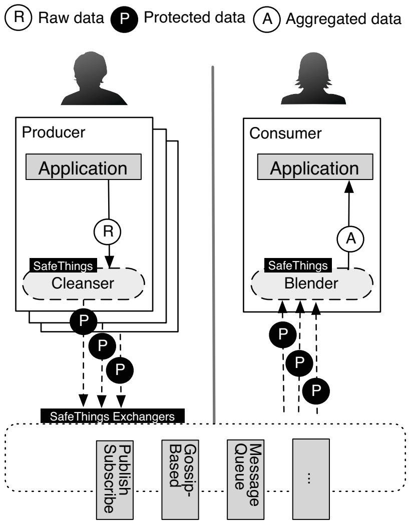

Figure 1. SafeThings: workflow overview.

The SafeThings architecture features three main components: *cleansers*, in charge of data security and privacy enforcement, *blenders* for data analytics, and *exchangers* for data exchange. Figure 1 depicts their interactions. Each component has a well-defined logical behavior while allowing distinct implementations, to enable the integration of legacy and new applications. Additionally, we consider three entities of interest: *producers*, *consumers* and *resource providers*. These entities are the end-points of a specific SafeThings workflow. They can be end-users, applications or other things producing or consuming data. Producers and consumers are also the trusted entities in the SafeThings workflow. We assume that data is secure and private when it is under the control of a data producer or when being used by a consumer. The untrusted domain lies between the output of a *cleanser* component and the input of a *blender* component.

The fundamental workflow of a SafeThings-based application is as follows. A *producer* generates some data (Figure 1–Ⓡ). This data is passed to a SafeThings cleanser, deployed close to/on the same device as the producer. The data is then processed and transformed into protected data (Figure 1–🅟). In our design, data protection includes the ability to encrypt, filter, or transform data into anonymous data. After cleansing, data is handed over to the SafeThings *exchanger* components. These components allow any party interested in that data to get access to it. Such an interested party might be some consumer of the data or a SafeThings blender. The *blender* processes data to provide aggregated knowledge (Figure 1–Ⓐ). The framework allows to chain blenders. The result of a (chained) blending process can be used by a consumer or shared again via SAFETHINGS exchangers. Data blending processes can be arbitrarily complex. Hence, blenders can also be deployed on larger cloud-based infrastructures to leverage more powerful computational resources.

The integration of legacy applications raises a problem for the considered trust relations, as these applications typically establish *direct* communication channels towards cloud-based aggregation services. These channels often use proprietary protocols and can be encrypted. Naturally, we do not expect these applications to change or be ported to the SAFETHINGS scenario. In those cases, the integration of completely unmodified applications is infeasible. However, for those applications where intercepting data is possible, we believe the *cleanser/blender* approach fits immediately and without any change to the original software. Note that our architecture allows the definition of an entire new data workflow, based on legacy data, without having to disable the original one.

New applications that follow the *cleanser/blender* model from the beginning do not face the same issue. Instead, *cleanser* components can be implemented in order to export publicly available data formats to be consumed by a diversity of *blender* instances. Producers have full control over their data privacy and can tune their *cleansers* according to their specific needs. Moreover, the flexibility of the architecture can be further exploited with the combination of different components and entity instantiations. The focus of the approach is to ensure that data consumers are able to extract meaningful knowledge from large amounts of data produced in a wide range of producers, while ensuring that data security is deployed by design. This is guaranteed by the deployment of data cleansers in the control of the producing entities. Note that producers can also be consumers, both applications or end-users, possibly chained.

The cleanser/blender architecture allows to deploy a large variety of security mechanisms. In particular, we envision the opportunity to leverage new approaches to data privacy that consider different trust models and security techniques, such as those that spread trust across multiple domains. In this context, resource providers are considered an important architectural entity. Typically, the manufacturer of the thing, with its own cloud infrastructure, must be entirely trusted. This level of trust is highly undesirable.

The remainder of the section details the three main SAFETHINGS components, and provides examples of possible instantiations.

A. SAFETHINGS *Cleansers*

The privacy risks of collecting data from things are increasingly visible, not only in attack reports that demonstrate what information can be learned about users by a malicious party, but also in the everyday lives users when they can realise how techniques such as profiling are enabling service providers to track and characterise them with precision. Cleanser components are aimed at transforming sensitive or private data in such a way that it can then be shared and pushed to an arbitrary number of Blenders. Naturally, this implies deploying novel machine learning algorithms and automation tools that aid users to identify sensitive data and propose ways to protect it. In particular, we consider the design of adaptive anonymisation techniques that reason on data sensitivity at a fine-grained resolution, in order to improve the trade-off between privacy and data quality. In fact, preserving extensive properties of the original data enables useful data analytics.

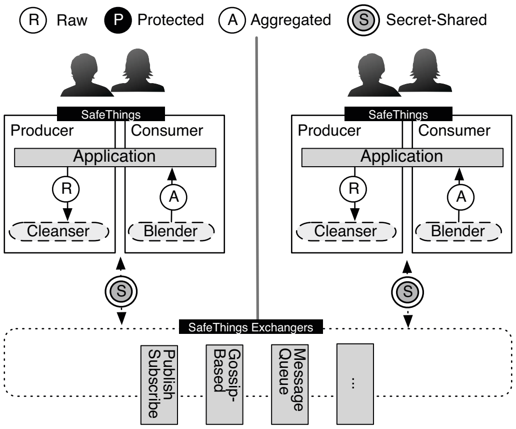

Figure 2. SAFETHINGS secret sharing analytics set-up.

The truly interesting aspect of the SAFETHINGS design is the ability to instantiate Cleansers with a large variety of innovative approaches to data privacy. This includes allowing Cleansers to secret-share data across different analytics components. These are able to cooperate in secure multi-party data analytics processes, yielding aggregated data results, without ever being able to disclose the original data the Cleanser worked on (Figure 2).

B. SAFETHINGS *Blenders*

Blenders are software components that enable the privacy-preserving aggregation of users' data at a massive scale. They operate on already cleansed data, which allows producers to retain control over their data privacy while still enabling data analytics at a societal level. These can be generic statistical functions, such as averaging, minimum and maximum estimation, distribution estimation, density maps, decentralized clustering of information, polls, and other data analytics primitives. Central to their design is the development of protocols and mechanisms for global-scale communication across blenders, which provably preserve user anonymity and data privacy. Along these lines, secure communication among blenders will be performed using a combination of epidemic or gossip-based protocols for reliability and multi-path/hop protocols for intractability.

C. SafeThings *Exchangers*

The framework deals with massive amounts of data being produced. Any instantiation of a communications component must be highly scalable. Additionally, considering the specific characteristics and heterogeneity of IoT devices, the same middleware has to deal with high levels of system dynamism. This dynamism arises from the fact that IoT devices may continuously enter and leave the system, and they can fail at any time.

The starting point in the design of a suitable middleware is the idea of global dissemination and local decision. Conceptually, data would be delivered at every Blender and each one of the Blenders would decide which data was of interest to process it. Such a design allows complete decentralization, key to scalability and resilience. Interestingly, by resorting to gossip-based protocols and judicious engineering it is possible to achieve efficient data dissemination in the large scale [4], [5], [13], [14]. Additionally, not only dissemination at scale is possible but it still allows to offer strong properties on such dissemination [12].

IV. Conclusion

We have proposed in this paper the principles and guidelines of SafeThings, a novel architecture for the IoT. This architecture embodies the notion of global, continuous and integrated data dissemination. The approach taken by SafeThings is based on the concept of Cleansers and Blenders: the former source data into protected data, the latter process protected data to provide analytics over it.

This simple starting point leads to an important capability: given a particular data producer (be it a user, an application, or a thing), the SafeThings approach would allow the cleansing of the data according to specific privacy or security needs. At the same time, cleansed data can be used by multiple Blender components (possibly users, applications, or things) to provide integrated services and aggregated information.

SafeThings is a first stepping stone towards an open IoT framework where the combination of data privacy and global scale data analytics not only is possible but is provided *by design*.

References

[1] Introduction to AWS Security by Design. https://aws.amazon.com/compliance/security-by-design/. Accessed: 2016-10-06.

[2] Privacy and Security by Design: An Enterprise Architecture Approach. https://blogs.oracle.com/OracleIDM/entry/privacy_and_security_by_design. Accessed: 2016-10-06.

[3] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *SIGSAC conference on Computer & communications security*, pages 901–914. ACM, 2013.

[4] R. Barazzutti, P. Felber, C. Fetzer, E. Onica, J.-F. Pineau, M. Pasin, E. Rivière, and S. Weigert. Streamhub: a massively parallel architecture for high-performance content-based publish/subscribe. In *DEBS'13*, pages 63–74. ACM, 2013.

[5] N. Carvalho, J. Pereira, R. Oliveira, and L. Rodrigues. Emergent structure in unstructured epidemic multicast. In *DSN*, pages 481–490. IEEE, 2007.

[6] Y.-A. De Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3, 2013.

[7] J. Dean. Designs, lessons and advice from building large distributed systems. *Keynote from LADIS*, page 1, 2009.

[8] C. Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[9] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and i will tell you who you are. In *SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41. ACM, 2010.

[10] J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-blending privacy. In *Advances in Cryptology*, pages 479–496. Springer, 2012.

[11] O. Hasan, J. Miao, S. B. Mokhtar, and L. Brunie. A privacy preserving prediction-based routing protocol for mobile delay tolerant networks. In *AINA*, AINA, pages 546–553. IEEE, 2013.

[12] M. Matos, H. Mercier, P. Felber, R. Oliveira, and J. Pereira. EpTO: An Epidemic Total Order Algorithm for Large-Scale Distributed Systems. In *Proceedings of the 16th Annual Middleware Conference (Middleware'15)*, pages 100–111. ACM, 2015.

[13] M. Matos, V. Schiavoni, P. Felber, R. Oliveira, and E. Riviere. Brisa: Combining efficiency and reliability in epidemic data dissemination. In *IPDPS*, pages 983–994. IEEE, 2012.

[14] M. Matos, V. Schiavoni, E. Riviere, P. Felber, and R. Oliveira. Laystream: composing standard gossip protocols for live video streaming. In *14th IEEE International Conference on Peer-to-Peer Computing*, pages 1–10. IEEE, 2014.

[15] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling churn in a DHT. In *Proceedings of the USENIX Annual Technical Conference*, pages 127–140, 2004.

[16] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. Chord: a scalable peer-to-peer lookup protocol for internet applications. *IEEE/ACM Transactions on Networking (TON)*, 11(1):17–32, 2003.

[17] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[18] V. Turner, J. F. Gantz, D. Reinsel, and S. Minton. The digital universe of opportunities: rich data and the increasing value of the internet of things. *IDC Analyze the Future*, 2014.

[19] L. Wang and J. Kangasharju. Measuring large-scale distributed systems: case of bittorrent mainline dht. In *IEEE International Conference on Peer-to-Peer Computing*, pages 1–10. IEEE, 2013.

[20] R. H. Weber. Internet of things–new security and privacy challenges. *Computer Law & Security Review*, 26(1):23–30, 2010.

[21] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel. A classification of location privacy attacks and approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, 2014.

[22] A. Whitmore, A. Agarwal, and L. Da Xu. The internet of things—a survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274, 2015.

[23] G. Zhong, I. Goldberg, and U. Hengartner. Louis, Lester and Pierre: Three protocols for location privacy. In *7th International Conference on Privacy Enhancing Technologies*, PET, pages 62–76, 2007.