# HomePad: Guardian of a Smart Home Galaxy

Igor Zavalyshyn, Nuno O. Duarte, Nuno Santos
INESC-ID / Instituto Superior Técnico, Universidade de Lisboa
{igor.zavalyshyn,nuno.duarte,nuno.m.santos}@tecnico.ulisboa.pt

***Abstract*—We are gradually moving to the reality in which every appliance in our homes is connected to the Internet. We already have smart fridges, TVs, lights and thermostats and there are more coming. But the whole design of these devices and the services they provide is centered around collection and further processing of sensor data in the cloud outside of users' control. In order to benefit from the desired service, users are forced to blindly trust the service provider with their sensitive information. As a result, this raises a lot of concerns about the private data being collected and used in ways it was not supposed to.**

**In this position paper we report the ongoing development of HomePad, a framework that aims to address the privacy problem of modern smart home services. It acts as a data hub and a processing unit for all the smart devices owned by the user. Apps provided by the service providers run in HomePad and process raw sensor data under users' control. With HomePad smart home owners can verify the privacy guarantees of each app at install time and eliminate any possible data leaks.**

## I. Introduction

Numerous smart home devices, from smart lights and locks to thermostats and cameras, rely on Internet connection in order to provide useful services. These devices constantly stream their sensor data to the service provider's cloud for processing, backup, remote access and control. End users, however, have little or no knowledge of what kind of data is collected and how it is later used, nor can they control the granularity of the data exposed. In order to benefit from a desired service, users are forced to deliberately share the sensitive data they would not normally need to for the purpose of the service being offered. As a result, 87% of US consumers [1] *are concerned about their personal information being collected and used in ways they were unaware of*. At the same time, 27% mentioned this concern as the *main reason why they do not currently own a smart device*, affecting not only the sales but also the overall trust in smart home technologies.

Various solutions have been proposed to solve the privacy problem of smart home services. Some of them proposed a centralized data storage for the user to keep all his personal data and manage its access [2][3]. With this approach, users are able to selectively share their personal data with third parties in exchange for useful services or even money reward. However, once the personal data is released from the storage, users have no way to know how it is later used and who it is shared with.

Instead of providing direct access to the raw sensor data, there are alternative solutions that opted for *privacy mediators* to perform data obfuscation and anonymization before releasing it from users' control [4][5]. Such mediators allow to minimize the risks of a possible privacy breach. However, none of the mentioned solutions provide evidence of applicability of this approach or implementation details.

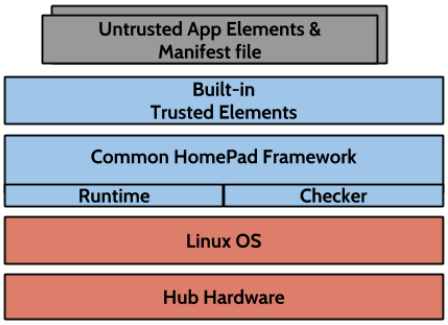


Fig. 1: HomePad architecture.

Following this idea, other solutions proposed to inspect the data flow of the service providers' apps for potential privacy leaks before granting them access to the sensitive data [6][7]. While this approach appears to be reasonable, it can be too strict for some of the smart home use-cases. By restricting the data flow from the devices to the cloud of service provider, it might sometimes make it impossible for these apps to send sensor data to the outside world even when the user needs this.

To address these challenges, we propose HomePad, a framework in which we try to combine two of the aforementioned approaches. We implemented the idea of privacy mediators as system-wide trusted functions that allow app developers not only to access the sensor data but also to process it in predefined ways, i.e. with known generated data types and in a trusted environment. These functions cover common smart home scenarios and aim to satisfy most of app developers' needs. The latter ones thus use these trusted functions as part of their apps. HomePad can then check which trusted functions the apps rely on and therefore detect what data types they operate with. By knowing this, HomePad can generate a graph representing a complete data flow of the apps and then use it to validate their privacy properties. This graph clearly states what data types the apps have access to and what they try to send outside. Unlike other solutions, such approach allows for more precise network access control and provides a flexible interface to the user.

## II. Overview of HomePad

We envision HomePad to be a central point in smart home environment. As Figure 1 shows, HomePad runs on a dedicated hardware on top of Linux OS. The framework itself consists of two main components: Runtime and Checker. The Runtime component provides a sandbox environment to execute the code written by app developers and communicate with trusted functions. Sandboxing is necessary to ensure that sensitive data cannot be sent outside bypassing the internal mechanisms

of Homepad. For the same reason, network connections can only be established through HomePad trusted functions and for particular destinations and data types.

The Checker component is responsible for the privacy analysis of the apps being installed in HomePad. It inspects their structure based on the manifest files provided by the developers. These manifests files list the trusted functions the apps depend on, and the sensor data they require access to. HomePad then uses these manifest files to validate the apps' privacy properties.

For the privacy validation, Checker relies on Prolog - a powerful declarative programming language that allows to model an app as a set of facts specifying its interactions with the sensor data. Checker then executes several queries over the app model to test its privacy properties and discover potential violations. After that, Checker generates a report with the results of each query and the final decision about the app, i.e. whether it is safe to install it. The report also contains complete data flow graph of a given app showing its structure and data flow.

The apps are analyzed for privacy leaks before they get access to the sensitive data. Unlike previous solutions, this is done at install time, which introduces a zero performance loss. Thanks to this analysis, HomePad is able to generate a user-friendly report informing the user regarding possible privacy violations. We argue that this approach is more clear than common permission-based approaches that require users to understand the applications being installed.

## III. Expected Contributions

We expect this work to produce the following contributions and results: (i) the design of a privacy-preserving framework for smart home owners with support for various devices and use cases; (ii) the implementation of HomePad prototype; and (iii) its thorough evaluation with real hardware and use cases. We also plan to compare the performance of HomePad with state of the art solutions available in terms of developer effort and user comfort.

For now, we have implemented a HomePad prototype with basic functionality and ported several existing smart home apps into it. The initial experiments showed promising results: malicious apps were correctly identified and reported. In the following months, we plan to complete the implementation of the framework and move on to its extensive evaluation. After that, we plan to submit a paper describing our work.

We also plan to explore the following ideas as future directions: (i) investigate how to leverage HomePad to perform compute-intensive operations at the client side without relying on the cloud; (ii) extend the privacy properties validation that is currently performed by HomePad to the cloud provider itself, i.e., ensure that in the cases where the data needs to be released to the cloud, the cloud continues to process the data according to the privacy policy of the user; and (iii) investigate ways to ensure that smart home devices communicate with HomePad only – whether this is possible to do without imposing changes to today's sensors or devices.


## Acknowledgment

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/50021/2013.


## References


[1] TRUSTe, "US IoT Privacy Infographics," https://www.truste.com/resources/privacy-research/us-internet-of-things-index-2015/. Accessed February 2017.

[2] A. Chaudhry, J. Crowcroft, H. Howard, A. Madhavapeddy, R. Mortier, H. Haddadi, and D. McAuley, "Personal data: thinking inside the box," in *Proceedings of Aarhus*, 2015, pp. 29–32.

[3] D. McAuley, R. Mortier, and J. Goulding, "The dataware manifesto," in *Proceedings of COMSNETS*, 2011, pp. 1–6.

[4] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, and B. Amos, "Privacy mediators: helping IoT cross the chasm," in *Proceedings of HotMobile*, 2016, pp. 39–44.

[5] Y.-A. Ya de Montjoye, E. Shmueli, S. S. Wang, and A. S. Pentland, "OpenPDS: Protecting the privacy of metadata through safeanswers," *PloS one*, 2014.

[6] E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "Flowfence: Practical data protection for emerging iot application frameworks," in *Proceedings of USENIX Security Symposium*, 2016.

[7] R. Herbster, S. DellaTorre, P. Druschel, and B. Bhattacharjee, "Privacy capsules: Preventing information leaks by mobile apps," in *Proceedings of MobiSys*, 2016, pp. 399–411.