Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

Securing the Smart Home: a real case study

Sabrina Sicari*[1] | Alessandra Rizzardi[1] | Daniele Miorandi[2] | Alberto Coen-Porisini[1]

[1]Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Varese, Italy
[2]U-Hopper, , Trento, Italy

Correspondence
*Corresponding author Sabrina Sicari, Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, via G. Mazzini 5, 21100 Varese (Italy). Email: sabrina.sicari@uninsubria.it

Abstract

Both people and organization are widely accepting and adopting the functionalities offered by the smart home or smart building applications. This is due to the many advantages, in easing users' every-day life and work, provided by the emerging Internet of Things (IoT) technologies and devices, equipped with sensors, cameras, or actuators, and able either to acquire information from the environment or to perform proper tasks. The main features of smart homes/buildings include real-time monitoring, remote control, safety from intruders, gas/fire alarm, and so on. Since within smart homes/buildings sensitive and private information are managed, security and privacy solutions must be put in place, in order to protect users/businesses' data against violation attempts as well as to guarantee the provision of reliable services. To this end, rules, in the form of policies, associated to the smart home/building resources, must be defined and correctly enforced, by means of a robust framework for handling the huge amount of IoT data managed. In this paper, the effectiveness and potentialities of a strategy based on sticky policies, integrated into a security and privacy-aware IoT middleware, are demonstrated within a smart home scenario. A test-bed is developed using real data-sets in order to conduct analysis on the execution times, response times to detected attacks, and memory occupancy of the proposed approach.

KEYWORDS:
Smart Home; Internet of Things; Security; Sticky Policy; Enforcement

1 | INTRODUCTION

The growing diffusion of smart home or smart building applications arises not only scalability and interoperability issues among the involved Internet of Things (IoT) technologies, devices, and protocols, but also serious concerns as regards security and privacy. In fact, data are usually exchanged via wireless communications and remotely transmitted to the homes' inhabitants or businesses' owners, thus increasing the risk of violation and leakage of information. Hence, the access to the smart homes/buildings' resources by unauthorized entities must be prevented and blocked by associating proper policies to the data themselves. To this end, the sticky policy paradigm (1) perfectly fits the need of monitoring the data flow in a capillary way. Such policies are called "sticky" in that they travel together with the associated data during the whole information flow, and also across multiple domain. Such a feature is very important in presence of the remote controlling functionality, which is typical of smart home/building applications. Moreover, they state the access control rules, the scope, and other relevant aspects related to the associated data. As a consequence, an enforcement system, managing sticky policies, coupled with a security and privacy-aware IoT middleware, has been defined (2) and applied, in this paper, to a real smart home scenario.

The system presented in (2) is called *NetwOrked Smart object (NOS)*. Each NOS has been conceived following the IoT general-purpose approach. Therefore, NOS is able to handle every kind of information in a dynamic way, thanks to the non-relational data model adopted. Data provided by different IoT contexts, such as smart home/building, smart transportation system, e-health environment, and so on, can be managed. In (2), NOS platform has been integrated with an enforcement framework based on sticky policies, thus allowing the owners of the data to express their preferences as regards the access to their resources. It is a matter of fact that often users are not fully aware of the constraints applied to their information or are not aware of the level of consent that has been granted by themselves for the use of the data (3). Such limitations must be overcome by designing efficient solutions able to support users in their decisions and to automatically protect sensitive or private data, as proposed in (2).

In this paper, the strategy adopted in (2) is further validated by means of a real-world case study in the smart home domain. On the basis of the authors' knowledge, this is the first application of sticky policies in the smart home scenario. More in detail, a real data-set, obtained from existing smart homes, is used in order to show the behavior of NOS's enforcement system and put in light its effectiveness and potentialities. Note that the smart home scenario is very relevant with respect to security and privacy issues in the IoT, because sensitive data, related to the people who live in the home, are managed. Such information can reveal people's habits and movements during the day, thus allowing malicious people to take actions against the inhabitants. Therefore, it is fundamental to guarantee adequate levels of security and privacy, which still represents an open issue in the smart home domain (4).

The remainder of this paper is organized as follows: Section 2 describes the background on smart home's security challenges and on the work presented in (2); Section 3 details the smart home scenario and presents the performance results obtained from the NOS prototype; Section 4 ends the paper.

2 | BACKGROUND

Security and privacy in the smart homes still present several challenges (5), in terms of access control, data integrity, confidentiality, and availability. Attacks that may occur can be distinguished in passive and active ones. The former consists in attempting to learn or make use of information from the smart home without affecting its resources (e.g., eavesdropping, traffic analysis); while the latter includes alteration of smart home's resources and functioning, as well as introduction of fraudulent information (e.g., DoS, replay, data integrity violation) (4). Some available solutions regard security alert systems, which, in response to particular or unusual events or changes within the smart home environment, notify the users (e.g., on their smartphones or tablets) to take countermeasures, possibly with the minimum delay (6). The authors of (7) propose a lightweight key management scheme to cope with encryption tasks in presence of power-constrained smart home's devices. What emerges is the need of a more robust and comprehensive system able to couple an efficient management and transmission of the data generated in the smart home with a security framework able to protect information and users' privacy. Such a requirement is the goal addressed in this paper, by means of the approach presented in (2).

With this regard, when considering the general IoT context, the level of maturity of solutions in the enforcement by means of sticky policies is still low. Some security solutions that use sticky policies in the cloud, such as (8) (9) (10) (11), have been proposed. Other application case studies for sticky policies are related to information exchanges among mobile devices (12) or digital ecosystems (13). Such works present various limitations, mostly regarding how to keep users in control of the access to their personal (or even sensitive) information, as well as how to react against violation attempts. Many approaches, proposed in the literature, are based upon the usage of robust encryption mechanisms, which are associated to sticky policies. Yet, the literature lacks a suitable and realistic approach for providing a secure, customizable, and cross-domain solution. Moreover, few real tests have been conducted in order to validate the proposed solutions in terms of effectiveness, efficiency, delay, storage, and overhead. The approach presented in (2) tried to cope with such issues in the context of IoT. However, its innovative contribution and its potentialities for a possible every-day or industrial application must be examined in depth. Therefore, in this paper, a real-world case study is analyzed and the outcomes of the solution proposed in (2) is deeply validated in a relevant smart home scenario.

2.1 | NOS platform

The solution proposed in (2) is based on NOS middleware. A network of NOSs is responsible for the storage and the processing of the information in a distributed manner. The goal is minimizing latency, providing support for user/service mobility, and

improving the resilience of the whole system avoiding single points of failure. NOSs are also able to evaluate, by means of well-defined automatic algorithms (14), the security and data quality of the information transmitted in order to satisfy user requirements, while providing a lightweight and secure information exchange process, based on an authenticated publish and subscribe mechanism that use Message Queue Telemetry Transport (MQTT) (15). NOSs essentially handle two main entities: (i) the nodes, conceived as heterogeneous devices (e.g., RFID, NFC, actuators, sensors, etc.) which generate data for the IoT platform and transmit them to NOSs through protocols that depend on the specific interacting devices (e.g., HTTP, CoAP, and so on); (ii) the users, who interact with the IoT system through services making use of such IoT-generated data, accessing them by means of mobile devices (e.g., smartphone, tablet) connected to the Internet (e.g., through WiFi, 3G, or Bluetooth). Since different data types and formats are received, NOSs initially put them in the *Raw Data* storage unit. In such a collection data are periodically processed, in a batch way, by the *Data Normalization* and *Analyzers* units, in order to obtain a uniform representation and to add metadata on security aspects (i.e., level of confidentiality, integrity, privacy and robustness of the authentication mechanism) and data quality ones (i.e., level of accuracy, precision, timeliness and completeness). Then, data are put in *Normalized Data* collection. Therefore, data, once received by NOS, are temporarily stored, processed, and, finally, shared with users. In the scenario considered in this paper, data do not flow across different realms. Regarding the nodes, they may decide, in an early registration phase, to agree with NOSs on a specific encryption mechanism for exchanging information in a secure way (i.e., registered sources), or may decide to transmit the data in clear (i.e., non-registered sources); whereas, users must register themselves to NOSs in order to establish proper credentials for accessing to resources and for future communications.

2.2 | NOS Sticky Policy Enforcement

A sticky policy-based enforcement framework has been integrated in the NOS platform presented in Section 2.1 (2). Policies are expressed by means of a flexible and interoperable specification language, based on *JSON* syntax. Due to its flexibility and effectiveness, *Attribute Based Access Control (ABAC)* (16) has been chosen as access control model. In such a solution, NOSs own no policies/credentials, while a trust authority is responsible for their management. The registered owners/sources of the data send them in an encrypted way along with the associated sticky policy; then each NOS can contact the trust authority in order to obtain the access decision related to the current sticky policy; finally, decryption keys for accessing the data themselves are released to eligible requestors, in a secure way (i.e., such keys may be further encrypted following the encryption algorithm agreed among requestor and NOS during the registration phase, as said in Section 2.1). Note that *ABAC* scheme is used since both requestors' credentials and related specific attributes are used to grant or not the access to the IoT platform and to the IoT resources. Note that such attributes have to match those eventually included in the sticky policy.

For each incoming data, the following information are gathered: (i) the kind of data source, which describes the kind of node; (ii) the communication mode, that is, the way in which the data are collected (e.g., discrete or streaming communication); (iii) the data schema, which represents the type (e.g., number, text) and the format of the received data; (iv) the data itself. Moreover, a sticky policy, containing the following information, is also associated: (i) the owner of the data (e.g., in the form of a unique identifier); (ii) one or more purposes for which the data can be used (e.g., statistical or analytical purpose, sharing in social networks, private use within organizations, etc.); (iii) a timestamp that points out the validity (i.e., the lifetime) of the data within the IoT/smart home system; (iv) one or more constraints which represent the rules to be applied to data.

After reception, a topic is assigned to each data; access to topics is regulated on demand by NOSs through proper requests to the trust authority, by means of the associated sticky policy. When a user/device is notified of a new published data, he/she/it has to demonstrate to own the correct attributes, compliant with the purpose, the validity, and the constraints contained in the sticky policy. Otherwise, the received data cannot be decrypted. Figure 1 summarizes NOS's components and interactions.

3 | CASE STUDY AND VALIDATION

The framework, just presented in Section 2.2, is evaluated in the smart home scenario and aims to demonstrate its feasibility for improving the reliability of a smart home system, while maintaining its efficiency. The proposed approach aims to prevent and counteract most of the passive and active attacks, described in Section 2. More in detail, simulated attacks concerns: data integrity violation attempts, false access requests injection, and eavesdropping. Note that attacks such as DoS or DDoS cannot be counteracted by the proposed mechanism, but an effective solution will be available in (17).

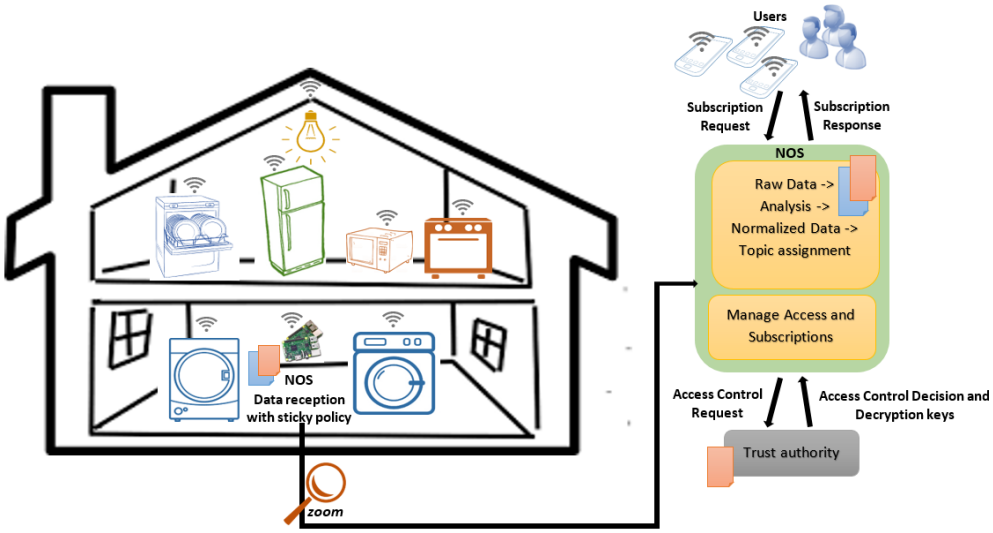

FIGURE 1 Overview of NOS's components and interactions in the smart home scenario

In the validation setup, data from real-world smart homes test-bed are used[1]. Details about the real deployments are available in (18). In this experiment, the behaviour of a set of nodes that send to a NOS the data gathered from seven smart homes is emulated by means of a laptop. A WiFi network is adopted to allow the communication between the laptop and the Raspberry Pi, where NOS is deployed. The same WiFi connection is also used for information exchange with the MQTT broker and with the trust authority module, implemented as separate components.

The goal of the proposed validation is to analyze the performance of the system in presence of a huge amount of data in real-time and in presence of malicious entities, by simulating: (i) the association of the data acquired by the smart homes with the sticky policies, as defined in (2); (ii) the access requests to data both by authorized and unauthorized/malicious users. We considered the data gathered in a period of one month in the year 2016 from the various smart meters, installed into the seven smart homes. Information collected include the electricity consumption of rooms' lights, washer, dryer, dishwasher, fridge, freezer, microwave, duct heater, furnace, and so on. Data fetch rate from data sources to the NOS corresponds to one minute. The sticky policy associated to each data has the format sketched in Listing 1.

```
{ "data content": {
    "timestamp": "dd/mm/yyyy hh:mm:ss",
    "data": "the data value",
    "datatype": "the type of the data",
    "owner": "the smart meter that acquired the data",
    "sticky policy": [{
      "purposes": [{
      "purpose1": "for what purpose the data can be used",
      "purpose2": "for what purpose the data can be used",
    }],
    "validity": "the lifetime of the data",
    "constraints": [{
      "constraint1": "who can access the data and for what purpose",
      "constraint2": "who can access the data and for what purpose"
      }],
}}
```

Listing 1: Sticky policy format

Three kinds of user are considered in the experiment: (i) the owners of the various smart homes, who wants to know the actual conditions of their home also in order to remotely control appliances (e.g., turn up the heating, turn off the light, etc.); they also have administration rights, since they can grant access permissions to other people; (ii) the inhabitants of the smart homes, who, with respect to the owners, have no administration rights (e.g., children, babysitters, housekeepers); (iii) the intruders who have no access permissions granted and want to get information about the habits of the home-dwellers and/or compromise the

[1] http://traces.cs.umass.edu/index.php/Smart/Smart

correct behavior of the smart home for malicious purpose. Access requests as well as malicious attacks by the aforementioned kinds of users are randomly simulated for the seven smart homes at a variable rate in the interval [5:10] minutes. Note that, in case of violation attempt, the system is robust, because the intruder, in order to gain access to the resources, must to: (i) know the credentials distributed a priori by authorized users; (ii) go up to the decryption keys; (iii) satisfy the defined sticky policy. However, invalid/failed access attempts can cause delays in the system's responses to valid requests, thus revealing the presence of an attack.

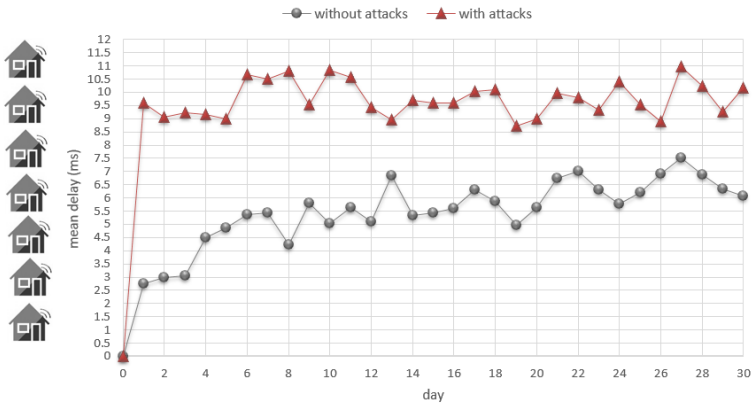

FIGURE 2 Query response times

In the presented scenario, relevant metrics to be analyzed are: (i) query response times, which represents the latency of access requests to the trust authority both in presence and without malicious attacks (see Figure 2); (ii) the memory occupancy required on NOS (see Figure 3), taking in mind that *Raw Data* and *Normalized Data* collections are not persistent, and that the malicious attacks considered in this case study do not directly affect such a metric; (iii) time required for detecting false access requests injections and decide to not release the requested information (see Figure 4). As reported in the figures, we can observe that NOS is able to manage data acquisition, processing, and provision maintaining stable levels of memory occupancy (on average 19 MB) and delay (on average 6 ms without attacks, and 10 ms in presence of attacks). Such an aspect represents an important indicator of the scalability and efficiency of the solution proposed in (2). In presence of malicious attacks and access violation attempts, the smart home's resources are protected and the delay is maintained under an acceptable threshold. Furthermore, the time required by NOS and trust authority for discarding bad data access requests is very promising (on average 2.5 ms).

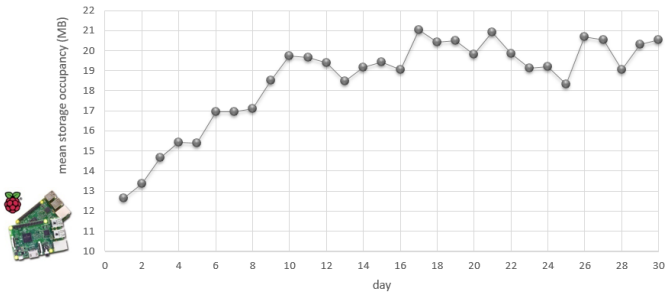

FIGURE 3 Storage occupancy

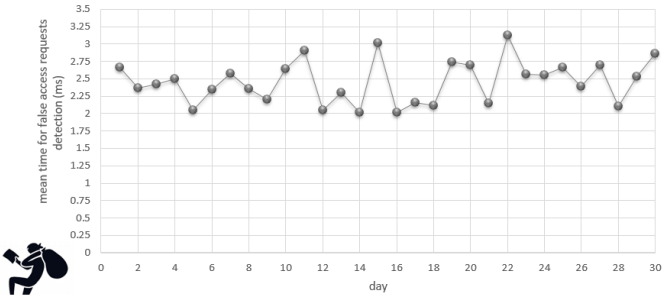

FIGURE 4 False access requests injection attack's detection

Further investigation, left as a future work, regards the deployment of more than one NOS belonging to the same smart home, in order to analyze the systems' efficiency, also considering NOSs' use as gateways for reducing consumption of power-constrained devices. Moreover, the development of a mobile application with a user-friendly GUI could be considered for testing the proposed approach with people in an existing smart home.

4 | CONCLUSION

In this paper, a sticky policy-based enforcement framework, which has been integrated into the NOS distributed and cross-domain IoT middleware, has been adopted within a smart home scenario. As demonstrated by means of a real-world case study, the designed solution provides advantages in terms of resilience, latency, and storage requirements, also in presence of malicious attacks, which represent critical aspects in the actual smart home/buildings applications. As a future work, we plan to expand our analysis to real complex scenarios, possibly with strict real-time requirements.

References

[1] Karjoth Günter, Schunter Matthias, Waidner Michael. Privacy-enabled services for enterprises. *IEEE 13th International Workshop on Database and Expert Systems Applications.* 2002;:483-487.

[2] Sicari Sabrina, Rizzardi Alessandra, Miorandi Daniele, Coen-Porisini Alberto. Security towards the edge: Sticky policy enforcement for networked smart objects. *Information Systems.* 2017;71:78-89.

[3] Mont Marco Casassa, Pearson Siani, Bramhall Pete. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. *IEEE 14th International Workshop on Database and Expert Systems Applications.* 2003;:377-382.

[4] Komninos Nikos, Philippou Eleni, Pitsillides Andreas. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials.* 2014;16(4):1933-1954.

[5] Robles Rosslin John, Kim Tai-hoon, Cook D, Das S. A review on security in smart home development. *International Journal of Advanced Science and Technology.* 2010;15.

[6] Tanwar S, Patel P, Patel K, Tyagi S, Kumar N, Obaidat MS. An advanced Internet of Thing based Security Alert System for Smart Home. *IEEE International Conference on Computer, Information and Telecommunication Systems (CITS).* 2017;:25-29.

[7] Naoui Sarra, Elhdhili Mohamed Elhoucine, Saidane Leila Azouz. Lightweight Enhanced Collaborative Key Management Scheme for Smart Home Application. *IEEE International Conference on High Performance Computing & Simulation (HPCS).* 2017;:777-784.

[8] Trabelsi S., Sendor J.. Sticky policies for data control in the cloud. *10 th Annual International Conference on Privacy, Security and Trust (PST).* 2012;:75-80.

[9] Leng Chunxia, Yu Huiqun, Wang Jingming, Huang Jianhua. Securing personal health records in the cloud by enforcing sticky policies. *Indonesian Journal of Electrical Engineering and Computer Science.* 2013;11(4):2200-2208.

[10] Li Shuyu, Zhang Tao, Gao Jerry, Park Younghee. A sticky policy framework for big data security. *IEEE First International Conference on Big Data Computing Service and Applications (BigDataService).* 2015;:130-137.

[11] Pearson Siani, Charlesworth Andrew. Accountability as a way forward for privacy protection in the cloud. *IEEE International Conference on Cloud Computing.* 2009;:131-144.

[12] Di Cerbo Francesco, Trabelsi Slim, Steingruber Thomas, Dodero Gabriella, Bezzi Michele. Sticky Policies for Mobile Devices. *Proceedings of the 18th ACM Symposium on Access Control Models and Technologies.* 2013;:257-260.

[13] Koshutanski Hristo, Ion Mihaela, Telesca Luigi. Distributed identity management model for digital ecosystems. *IEEE International Conference on Emerging Security Information, Systems, and Technologies (SECUREWARE 2007).* 2007;:132-138.

[14] Sicari Sabrina, Rizzardi Alessandra, Miorandi Daniele, Cappiello Cinzia, Coen-Porisini Alberto. A secure and quality-aware prototypical architecture for the Internet of Things. *Information Systems.* 2016;58:43-55.

[15] Rizzardi Alessandra, Sicari Sabrina, Miorandi Daniele, Coen-Porisini Alberto. AUPS: An Open Source AUthenticated Publish/Subscribe system for the Internet of Things. *Information Systems.* 2016;62:29 - 41.

[16] Goyal Vipul, Pandey Omkant, Sahai Amit, Waters Brent. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. *Proceedings of the 13th ACM Conference on Computer and Communications Security.* 2006;58:89-98.

[17] Sicari Sabrina, Rizzardi Alessandra, Miorandi Daniele, Coen-Porisini Alberto. Reacting to Denial of Service Attacks in the Internet of Things. *Tech. Rep.*. 2017;.

[18] Barker Sean, Mishra Aditya, Irwin David, Cecchet Emmanuel, Shenoy Prashant, Albrecht Jeannie. Smart*: An open data set and tools for enabling research in sustainable homes. *SustKDD.* 2012;111:112.