Internet of Things 7 (2019) 100069

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

Fog-based local and remote policy enforcement for preserving data privacy in the Internet of Things

Abduljaleel Al-Hasnawi[a,*], Steven M. Carr[b], Ajay Gupta[c]

[a] Electrical Power Techniques Engineering Department, Al-Furat Al-Awsat Technical University, Kufa, Iraq
[b] Department of Computer Science, Western Michigan University, 1903 W. Michigan Ave., Kalamazoo, MI 49008-5466, USA
[c] Wireless Sensornets Laboratory, Western Michigan University, Kalamazoo, MI 49008-5466, USA

ARTICLE INFO

Article history:
Received 17 January 2019
Revised 7 May 2019
Accepted 2 June 2019
Available online 8 June 2019

Keywords:
Policy
Policy enforcement
Fog
IoT
Active data bundles

ABSTRACT

The pervasive nature of the Internet of Things has resulted in generating a huge amount of data about the lives of IoT users. This data includes Personally Identifiable Information (PII) that reflects people's behaviors, interests, lifestyles, and everyday routines. Protecting PII from privacy violations is a challenge since IoT data need to be handled by public networks, servers, and clouds, which are untrusted parties for data owners. In this paper, a solution called Policy Enforcement Fog Module (PEFM) is proposed for protecting sensitive IoT data whenever they are accessed throughout their entire lifecycle. PEFM uses the power of policy enforcement in the edge-fog infrastructures for protecting data accessed within users' local domains. For data that need to be sent to remote domains, PEFM uses Active Data Bundle (ADB); an executable and self-protecting construct that can run on any visited host and enforces privacy policies automatically for data accessed by these hosts. To test the feasibility of PEFM in realistic IoT systems, a framework of using PEFM as a privacy control for Foscam home security system is simulated. The experimental results show that PEFM assures data privacy via data minimization due to selective data disclosures. Better privacy controls with minimal overhead can be achieved if most PEFM processes are executed by the local fog nodes. Migrating parts of PEFM processes to remote fog nodes or the cloud incurs more overhead than using strictly local fog nodes. This overhead is the cost for a higher level of privacy regarding lifecycle data protection.

1. Introduction

1.1. Background

The Internet of Things (IoT) is an emerging paradigm that is connecting large collections of heterogeneous objects in the real world with the infrastructures in the cyber world to provide useful services to the end users, cooperatively [1]. It could provide connectivity for anything during any time and from anyplace for anyone. The IoT is generating an unprecedented volume and variety of data at unprecedented speeds. To keep up with IoT data sources, analysis of IoT data must be very rapid. A promising approach to handle the volume, variety, and velocity of IoT data is to use the computing model of Fog Computing (FC). FC is a highly-virtualized computing platform that provides compute, storage and networking services

* Corresponding author.
E-mail addresses: abduljaleelmoh.alhasnawi@wmich.edu (A. Al-Hasnawi), steve.carr@wmich.edu (S.M. Carr), ajay.gupta@wmich.edu (A. Gupta).

https://doi.org/10.1016/j.iot.2019.100069
2542-6605/

between edge devices and traditional Cloud services [2]. FC is implemented via fog nodes that consist of physical and virtual components that work together to perform fog computing for IoT. A fog node performs several functions, including, collecting data from IoT sensors, analyzing the most time-sensitive data, virtualizing remote cloud services for fog applications, providing temporary data storage, and sending pre-processed data to the cloud for non-real-time applications and long-term storage [3].

Raw data fed to fog nodes include *Personally Identifiable Information (PII)* [4]. According to a report [5], 33% of IoT data are sensitive since they include PII. PII is the category of data that can potentially identify a specific individual and, if disclosed, could harm that individual. For many IoT applications, PII can be automatically obtained from sensors on or in clothes, devices, cars, homes, offices, even bodies, revealing many aspects of data owners' lives. Handling sensitive IoT data without sufficient and appropriate privacy controls makes them vulnerable to unauthorized disclosures by attackers and malicious insiders.

Privacy controls are all kinds of methods, algorithms, mechanisms, and tools that can be used in a system to prevent adversaries from violating the privacy of the system's users [6]. One of the most powerful approaches that can be considered as a privacy control for data is the enforcement of privacy policies. Privacy policies are a set of privacy rules that determine the authorized operations on the given data, such as allowing or denying read, write, or delete operations [7]. The main idea of policy enforcement can be outlined as follows: (a) associating data to be protected with a set of privacy policies that determines usage rules for these data; (b) for each data use request, evaluating applicable policies for the requested data, and determining if and the way in which these data may be used; and (c) using a policy enforcement mechanism to enforce the results of policy evaluation. Typically, the Extensible Access Control Markup Language (XACML) is used to specify and evaluate access control policies [8]. XACML is an open standard-based language developed to standardize fine-grained access control using eXtensible Markup Language (XML).

The policy enforcement process can be done at the data source host where it has sufficient computational power to perform the enforcement or at a trusted third party, or the destination host [9]. However, for a high level of privacy protection, it is highly desired to perform policy enforcement for sensitive data wherever they are accessed throughout their entire lifecycle. This is our approach. Data Lifecycle is a set of sequential stages for data starting from data generation, among different intermediate stages including aggregation, dissemination, analyzing, and storing; till data destruction. The intermediate stages vary according to the environments and data applications. For data lifecycle protection, we utilize a well-investigated scheme called *Active Data Bundle (ADB)* [10]. The ADB is a self-protecting construct that inseparably bundles data (which is to be protected), privacy policies for these data, and a policy enforcement engine (executable code). The ADB is created at the source node, disseminated as an executable packet through multiple nodes, and enabled at the destination node. When the destination node executes the received ADB, the ADB starts the policy enforcement process. ADBs perform two major protection operations, namely evaporation and apoptosis. Evaporation is the process of destroying the part of ADB data that the host is not allowed to access (because of policy enforcement). Apoptosis is the process of complete self-destruction of an ADB when either the host is not authorized to access any data or the retention period for the carried data has expired.

1.2. Research challenge

According to a recent IoT privacy forum [11], IoT technologies extend the data collection practices of the online world to the offline world by enabling and normalizing the tracking of users' preferences and behaviors in their offline local domains. Hence, IoT makes it easier for third parties to identify people in public and private spaces, especially when they can collect people's emotional states over long periods. The forum's report states that most IoT services are user attractive and when more IoT-like features are released, users will have less ability to control them. Furthermore, IoT devices are not neutral; they are constructed with a commercial logic by embracing and extending the logic of social media — with intentional disclosure, participatory crowdsensing, and continued investment in interaction. The IoT market shifting toward smart features that are intentionally unobtrusive leads to less understanding of data collection, and less ability to decline those features. This motivates us to investigate privacy issues in IoT and propose a solution to address the identified issues.

From the Privacy Policy (PP) point of view, signing up for any IoT service requires that a user should go through a PP and gives consent to the Service Provider (SP) regarding the collection of personal data. The collected data includes (but is not limited to) account information, contact information, payment information, technical information, social media information (if the user chooses to connect the service to his or her social account), as well as, time-based usage information; which are PII. Usually, an SP collects PII to provide a certain service to the end user, to manage users' accounts and profiles, and to identify and authenticate the users for the service usage. Typically, a PP may state that the SP implements a variety of security measures to assure security and safety of user's data. However, this does not guarantee that the privacy of the user is assured by the PP. For instance, when security is breached in the SP's system, the users' PII is consequently leaked and become available to unauthorized parties. The PP also may state that the SP does not sell or trade customers' information to outside parties. However, the outside parties do not include the trusted third parties who assist SP in managing and supporting the system. For instance, SP may use cloud storage to store the collected customers' data, in this case, the cloud storage is a trusted third party to the SP. To get user consent for the SP's PP, a PP may state that by using the service, the user consents to the PP. At this point, a user does not have a choice to deny the PP if he decides to use the service. Hence, the only choice is accepting the stated PP. Once, a user accepts the PP, this door is open to many privacy threats [12], including (but not limited to) *Identification, Tracking, Profiling, Monitoring*, and *Linkability*.

The privacy problem that needs to be addressed by the proposed research, therefore, can be stated as:

Sensitive IoT data are highly vulnerable to privacy violations, which include exploitation of security threats by the attackers to access PII, a collection of data by malicious insiders more than what is needed for specific purposes as well as retaining data longer than needed. Furthermore, there is a lack of data owners' control over their sensitive data when these data leave the owner's private domain and are disseminated to public IoT domains. Lastly, data shared to untrusted public domains as passive entities are not able to protect themselves from privacy violations.

1.3. Our objectives

The research problem specified in the problem statement above can be addressed by satisfying the following objectives:

- Enforcing a PP on raw IoT data as soon as they are generated to reduce the privacy risks associated with processing, sharing, and storing data without privacy preservation.
- Assuring fine-grained access control for sensitive IoT data throughput their entire lifecycle.
- Increasing data owners' control over their data when data leaves their private spaces.
- Assuring data minimization based on *need-to-know* and *selective data disclosure*.
- Ensuring remote data destruction when data is no longer needed.
- Transforming data from passive entities to active entities that can protect themselves in untrusted public domains.

1.4. Review of related work

There are different state-of-the-art privacy solutions proposed in the literature to address privacy issues in IoT. These solutions can be classified, based on the mechanism used to achieve privacy, into *cryptography*-based solutions, *identity management*-based solutions, *policy enforcement*-based solutions, and *self-protecting data* solutions.

Cryptography can be used as a basis for assuring privacy in IoT by encrypting data and protecting their confidentiality. Different cryptography-based solutions are proposed such as Public-Key Encryption (PKE) [13], Attribute-Based Encryption (ABE) [14], Homomorphic encryption [15], and Point-to-Point Encryption (P2PE) [16]. Cryptography is usually computationally expensive and does not assure fine-grained access control. Identity management is another approach for assuring IoT privacy by protecting the identity of the communicating parties in IoT networks. There are many identity management-based privacy solutions in the literature such as k-anonymity [17], Pseudonymity [18], Attribute-Based Signature (ABS) [19], group signatures [20], and ring signatures [21]. Most of these solutions are limited to protecting a user's identity rather than protecting a user's data that include PII, which is our approach. Policy enforcement is a powerful approach to protecting data privacy. However, the power of this approach relies on the way how it is performed and the location where it is done. Some solutions rely on a Trusted Third Party (TTP) to perform policy enforcement such as mandatory enforcement of privacy policies (PRECIOSA) [22]. Other approaches rely on fog computing, like us, to perform policy enforcement such as policy-based management [23], and cloudlet-based privacy mediator [24]. However, these approaches do not set up a mechanism for remote policy enforcement, like our approach. Self-protecting data solutions rely on the idea of attaching data with a mechanism that assures no access can be done for these data unless the privacy protection is done. For that, either a container used to carry data and their policy such as in the DigiBox scheme [25] or using a mobile agent such as in the agent-based informed consent [26] solution and Personal Digital Rights Management (PDRM) [27]. These solutions have some limitations; DigiBox is limited to protecting data during transmission only, agent-based informed consent is a user-centric approach rather than data-centric, and PDRM is limited to *allow* or *deny* decisions only and does not support partial data disclosures as our approach does.

2. Our approach

The proposed solution, called *Policy Enforcement Fog Module (PEFM)*, is a software module that performs local and remote policy enforcement for sensitive IoT data. PEFM is designed to be integrated into the fog node's middleware to be the first processing point for raw sensing data that acts on data as close as and as soon as they are data generated. From the point of view of a given fog node *N*, we distinguish two kinds of IoT applications. First, an application running within *N* is its *local application*. Second, any application running on any node other than *N* is a *remote application* for *N*. Hence, we also distinguish two forms of data processing in IoT from the fog computing point of view: (a) processing data completely at the edge, which is required for most of the real-time applications that are run locally; and (b) preprocessing data at the edge and disseminating them to a remote server or Cloud for more processing and long-term storage, which is applicable to non-real-time applications that are run remotely. To fulfill the privacy requirements of the abovementioned forms of data processing, PEFM is designed to have two major submodules: *Local Policy Enforcement Module (LPEM)* for protecting data accessed by local applications; and *Remote Policy Enforcement Module (RPEM)* for protecting data accessed by local applications. LPEM and RPEM are work collaboratively to assuring a lifecycle privacy preservation for IoT data. PEFM enables data owners to participate, collaboratively with system administrators, in specifying privacy policies for their sensitive data through policy negotiation. Data user who requests the data or whom the data collected for. The following two subsections outline LPEM and RPEM in detail.

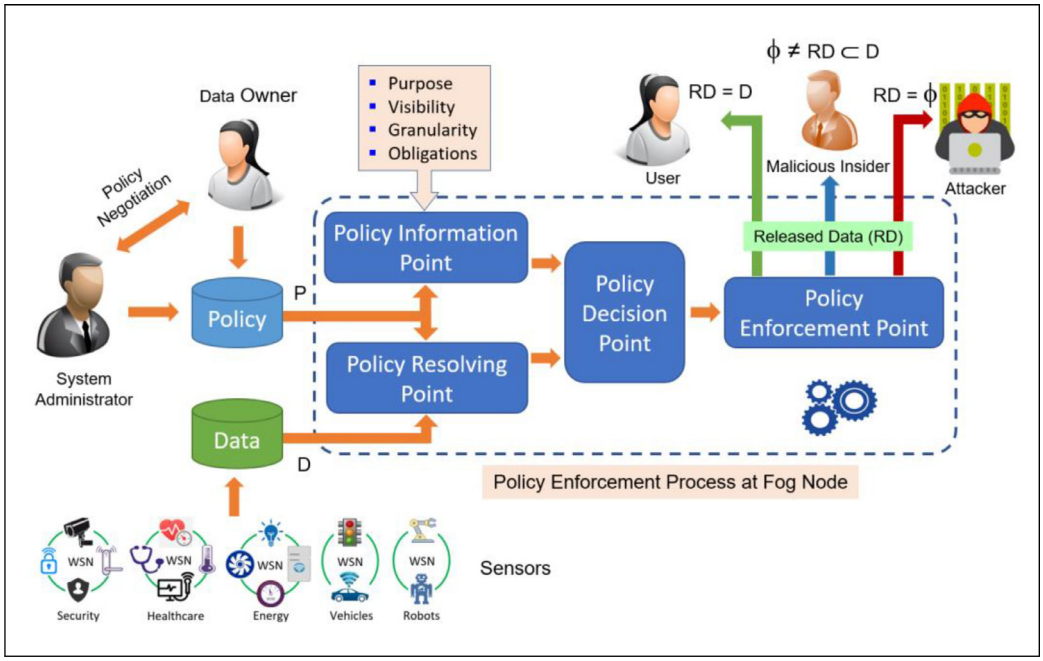

Fig. 1. Structure and process flow of the Local Policy Enforcement Module (LPEM).

2.1. Local Policy Enforcement Module (LPEM)

LPEM is developed on the basis of the XACML reference architecture [8]. The LPEM consists of four major components, as shown in Fig. 1, *Policy Resolving Point (PRP), Policy Information Point (PIP), Policy Decision Point (PDP)*, and *Policy Enforcement Point (PEP)*. PRP is designed to get a set of data items D from the fog node's Data Repository (DR) and obtain the set of applicable privacy policies P from the fog node's Policy Repository (PR). PIP provides information on *purpose, visibility, granularity*, and *obligations* for the D based on P. Based on the information from PRP and PIP, PDP makes data access decisions to be enforced by a PEP, which in turn passes only released data (RD) allowed by a PDP decision to local applications.

LPEM assures selective data disclosure by offering different levels of disclosures, based on data user: *full*, which means RD = D; *partial*, which means RD $\subset$ D and RD $\neq \phi$; and *null*, which means RD = ϕ, as shown in Fig. 1. In this way, sensitive data fed from LPEM to the users via their local IoT applications are guaranteed to satisfy privacy policies defined for these data. Depending on the policies, some of these data are never shown to some local applications, while others might be shown to some local applications after privacy-preserving transformations such as data masking or de-identification.

2.2. Remote Policy Enforcement Module (RPEM)

For extending privacy protection for data accessed by remote IoT applications, we propose a module for enforcing privacy policies remotely called *Remote Policy Enforcement Module (RPEM)*. RPEM is developed based on the XACML reference architecture [8] and the Active Data Bundle (ADB) scheme [10]. RPEM is integrated into the fog node's middleware as a *mediator* for setting up a mechanism for protecting sensitive IoT data whenever they accessed by remote nodes. The RPEM process is activated to start when data preprocessed by a fog node need to be sent to a remote node for more processing or long-term storage. As shown in Fig. 2, D from DR constitutes the ADB's data to be protected. P from PR bundled with D, and PDP & PEP constitute the ADB's policy enforcement engine. A fog node performs the ADB creation process and then disseminates the created ADB over the IoT network to a remote node including a remote fog node or cloud. ADB is transmitted as a packet with a self-protecting mechanism so no access to the ADB's data can be done without satisfying the carried privacy policies. At the destination node, an ADB starts the remote policy enforcement process by using PDP to evaluate P on D for requested user and decides whether D should be fully disclosed (RD = D), partially disclosed (RD $\subset$ D and RD $\neq \phi$) or denied (not disclosed) (RD = ϕ), as shown in Fig. 2. Then, PDP passes to PEP data allowed by its decision. Lastly, PEP passes RD to users via remote IoT applications.

In this way, sensitive data carried by an ADB and disseminated to remote IoT applications are guaranteed to satisfy privacy policies defined for these data. Depending on the policies, some of these data are never shown to some remote applications, while others might be shown to some remote applications after privacy-preserving transformations such as data masking or de-identification.

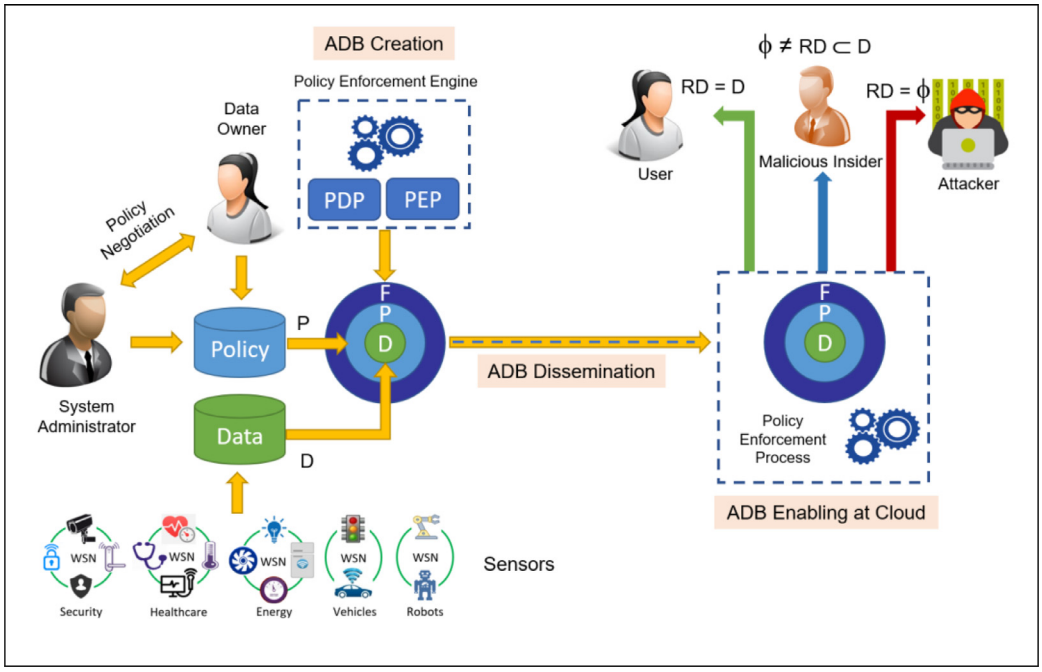

Fig. 2. Structure and process flow of the Remote Policy Enforcement Module (RPEM).

3. Framework for evaluation

3.1. Case study

To test the capabilities of the proposed PEFM modules, namely, LPEM and RPEM, we select the Foscam Home Surveillance System (FHSS) [28], as a proof-of-concept case study. FHSS is an automation system able to monitor a home environment (indoor and outdoor) using intelligent hardware and software subsystems with limited (or without) human intervention. FHSS is a commercial home security system that is recognized as the best home surveillance system of 2018 [29]. It is integrated with Smart Home (SH) technology [30], an IoT-enabled home equipped with sensing, actuating, monitoring, and controlling devices that can be controlled remotely by a smartphone, tablet, or computer. The major task of FHSS is providing safety services for home residents as well as assuring their security through erroneous event detection. FHSS includes setting up a network of real-time, high-definition surveillance cameras in the home to be monitored either by third parties or by homeowners themselves using smartphone apps.

The FHSS consists of two major components: IP Camera (IPC) and Base Station (BS). The major data source in FHSS is the IPC which monitors a given surveillance area within its fields of view; the IPS acts as a sensor in our case study. Multiple IPCs are distributed over multiple surveillance areas within an SH. The Foscam E1 IPC [31], is a 1/2.9″ CMOS sensor type that generates H.264 video format with 1080P (1920×1080) and 720P (1280×720) quality. It provides 25 ft IR range with a 110° view angle. The E1 IPC uses IEEE 802.11b/g/n wireless standard with data rates of 11 Mbps, 54 Mbps, and 150 Mbps for IEEE 802.11b, IEEE 802.11 g, and IEEE 802.11n, respectively. An IPC sends a *Raw Video Stream (RVS)* to the SH's BS, which is a central hub connecting the IPCs to the Internet. The BS acts as a *fog node* in our case study that is a part of the local fog and connecting the SH's IPCs with the cloud. The Foscam E1 BS [31] has 720P &1080P recording resolution and H.264 compression format. It provides a single channel 720P & 1080P synchronous video playback. The E1 BS also uses the IEEE 802.11b/g/n wireless standard with $1 \times 10/100$ Mbps RJ45 port network interface. It supports IOS, Android, and any 3G/4G Smartphones.

3.2. Privacy threats in FHSS

The FHSS' IPCs continuously watch home residents. Data collected by these IPCs reflect lifestyle, everyday routine, and all other personal aspects related to the people within an SH. These data are sensitive, and if they disclosed to unauthorized parties or authorized parties in an unauthorized manner, the privacy of home residents would be violated. FHSS applications such as security, child monitoring, elder monitoring, etc. can be used in combination with other SH applications such as an alarm system, door lock system, light system, etc. In this way, FHSS increases the risks of a purpose drift: information collected for one purpose is used for another purpose [32]. Christin et al. [33] investigated the privacy risks associated with sharing environmental surveillance data. They mention that the contents of the captured images are most likely revealing

personal (sensitive) information about the people in that environment. Therefore, FHSS and SH applications together can collect data that are highly vulnerable to privacy attacks.

3.3. The framework of using PEFM for FHSS

To address privacy threats in the FHSS, we integrate the proposed PEFM system in the FHSS as a privacy control for protecting sensitive FHSS data from privacy violations. The PEFM is placed on the BS's software system (fog node) to act on the RVS packets generated by IPCs connected to the BS. Only sensitive RVSs are saved on the PEFM's DR to be ready for policy enforcement. In the case of sending data to the user within a local home domain, the PEFM's LPEM sends local notifications after performing local policy enforcement on them as *Local Released Data (LRD)* to the local applications. In the case of sending data to the user within a remote cloud domain, the PEFM's RPEM sends remote notifications in the form of ADBs which are disseminated to the cloud (among intermediate fog nodes) and then released to the user as *Remote Released Data (RRD)*. Note that each RRD is already targeted for a user U who requested data. This is because all policies for RRD and U were enforced in the cloud using ADB-VM. An RRD targeted for U needs to be delivered to U securely, for example, using strong encryption. We do not need to use ADB for the last hop on the route to U since policies for data are already enforced, and RRD is no more sensitive.

3.4. Experimental scenarios

To test the feasibility and efficiency of the PEFM system for assuring privacy in the FHSS case study, we consider different scenarios regarding privacy risk of having malicious insiders, attackers, or both in the system. We categorize our scenarios into two major groups: Scenarios for local Foscam system—where LPEM is the privacy actor for them; and Scenarios for remote Foscam system—where RPEM is the privacy actor for them.

3.4.1. Local Foscam scenarios

We consider four different scenarios for the local Foscam system. First is the *Home No-Risk Scenario (HNRS)* where home residents Bob and Michelle are at home, and they can access surveillance data locally using a Foscam mobile app. This scenario is free from attackers or malicious insiders, so it is considered as a baseline for a local Foscam system. In this scenario, LPEM enforces privacy policies even for normal users to maintain data access authorization by home residents who have different levels of authority in accessing the Foscam data. Second is the *Home Malicious Insider Scenario (HMIS)* where Bob and Michelle have a technical issue in their Foscam system. They have called the Foscam customer service which in turn sends a technician to fix the issue. Now, the technician is considered in this scenario to be a data user which may or may not be a malicious insider. In both cases, access to home data should be controlled by PEFM's LPEM. The third is the *Home Attacker Scenario (HATS)* where Bob and Michelle are at home, and there is an attacker who attempts to access the home surveillance data. PEFM's LPEM assures that this kind of access is denied by enforcing privacy policies on the attacker's access. Fourth is the *Home High-Risk Scenario (HHRS)* where Bob and Michelle have a technical issue in their Foscam system, and a technician comes to fix the issue. Now, the technician is considered in this scenario to be a data user which may or may not be a malicious insider. At the same time, there is an attacker who attempts to access the home surveillance data. PEFM's LPEM assures that access by the technician is controlled and access by the attacker is denied by enforcing privacy policies.

3.4.2. Remote Foscam scenarios

We also consider four different scenarios for the remote Foscam system. First is the *Cloud No-Risk Scenario (CNRS)* where Bob and Michelle are away from home, and they can access surveillance data remotely via the cloud using a Foscam mobile app. This scenario is free from attackers or malicious insiders, so it is considered as a baseline scenario for the remote system. Second is the *Cloud Malicious Insider Scenario (CMIS)* where a technician accesses home data remotely to fix an issue in the Foscam system. This kind of access should be controlled by RPEM. The third is the *Cloud Attacker Scenario (CATS)* where an attacker attempts to access the home surveillance data via the cloud. This kind of access should be denied by RPEM via enforcing privacy policies on the attacker's access using the ADB. Fourth is the *Cloud High-Risk Scenario (CHRS)* where Bob and Michelle have a technical issue in their Foscam system, and a technician accesses home data remotely via the cloud to fix the issue. Now, the technician is considered in this scenario to be a data user which may or may not be a malicious insider. At the same time, there is an attacker who attempts to access the home surveillance data remotely via the cloud. RPEM assures that access by the technician is controlled and access by the attacker is denied by enforcing privacy policies.

3.5. Experimental setup

Simulation experiments are conducted to evaluate PEFM performance in achieving privacy and the corresponding FHSS performance regarding latency (LAT) and throughput (T). LAT is the period from the instant when the *first* bit of RVS is sent by a given IPC till the instant when either the *last* bit of LRD is delivered to a local application or the *last* bit of RRD is delivered to the remote application. T is the maximum number of jobs that can be completed by a system in a unit of time. The system performance for a scenario with a certain level of privacy risk is compared with the system performance for the

baseline scenario, with no risk, to measure the resulted overhead of using PEFM as privacy control for FHSS. The simulation is conducted using SimPy, a process-based discrete event simulation framework based on standard Python, running in the PyCharm, IDE environment. The simulation runs on an Intel Core i7-4710 HQ 250 GHz processor with 8 GB RAM. To test PEFM performance on different network topology sizes, we have varied the number of smart homes keeping the number of IPCs connected to each BS constant. Five network configurations based on realistic Foscam specifications have been simulated: *Config 1, Config 2, Config 3, Config 4*, and *Config 5*—having 8, 32, 72, 128 and 200 BSs. Each BS is connected to 4 IPCs generating RVSs and three smartphones receiving RD.

3.5.1. Assumptions

The assumptions for the simulation can be stated as follows:

1) Foscam BS is the only IPC gateway: BS serves all incoming and outgoing data for the IPCs. This means that we do not consider the cases when IPCs communicate their data directly with the Cloud without going through the smart home's fog node.
2) Single-packet messages: Each message is sent as a single packet. In other words, we consider packet switching [34], in our simulation networks instead of message switching where each message includes multiple packets.
3) An Active Data Bundle is sent as a packet: An ADB is sent in the simulation as a packet with a payload including sensitive data, metadata (including privacy policies), and an executable code as a policy enforcement engine [10].
4) All generated data are sensitive: RVS, generated by IPCs, as well as data sent by BS to other BSs in the system are all sensitive. However, these data have different sensitivity levels.
5) Processing nodes: Fog node (BS), intermediate fog node, and Cloud are the only processing nodes. IPCs perform sensing and actuating only. User's smartphones are just receiving LRD and RRD from BS and Cloud, respectively.
6) Reliable links: The BS-to-local user device and the Cloud-to-remote user device links are so reliable that no acknowledgments from the local device to BS and from the remote device to Cloud are needed.
7) XACML privacy policy evaluation: In general, evaluating XACML privacy policies produces four possible results: Permit, Deny, Indeterminate, or Not Applicable [8]. In our simulation, we consider only Permit (partial or full disclosure) and Deny (null disclosure).

3.5.2. Limitations

There are a few limitations in our research that are beyond the scope of this work and is left for future investigations. We limit our scope as follows:

1) Data in IoT might be public and private (sensitive). We consider only sensitive data. Public IoT data are outside of the scope of this research.
2) We rely on the data owner to identify which data are sensitive, so identifying sensitive data is outside of the scope of this research.
3) We focus on protecting sensitive IoT data. Protecting other aspects of privacy for IoT entities is outside of the scope of this research.
4) We investigate the privacy of data. Security is outside of the scope of this research. However, some security issues, intertwined with privacy issues, are addressed by our research.
5) We limit our investigation to the selected privacy threats (as discussed in Section 1.2). Other threats are outside of the scope of this research.
6) We limit our investigation to fog computing rather than any similar terminologies—including cloudlets, Mobile Edge Computing, Mobile Cloud Computing, and so on.

3.5.3. Simulation parameters

We consider constant parameters for network nodes and network links. The parameters values are chosen based on the values commonly used in the literature [35–37]. However, some values are chosen somewhat arbitrarily but within a certain range when no significant impact is envisioned. For each processing node, we define CPU speed, which is the maximum number of instructions that a node's CPUs can process per second, defined in MIPS (Million Instructions Per Second) [38]; and RAM size, which is the size of the short-term memory of the network node defined in GBs [39]. The values of CPU speed are 4000 and 44,000 for BS and Cloud, respectively. The values of the RAM size parameter are 4, and 40 for BS and Cloud, respectively. The network link is defined by its source and destination nodes. For each link type in the simulation, we define *bandwidth* (Mbps), *propagation delay* (ms), and *link parallelism* (simultaneous packets). The values of these parameters are 150, 1, 4 and 300, 100, 10 for IPC to BS and BS to Cloud, respectively.

We consider random parameters (variables) for packets, as shown in Table 1. Packets are the major processing and communication units in the simulated FHSS system, which has four types: RVS, LRD, ADB, and RRD. We model IPC as a discrete event generation sensor, which generates RVS packets when an event occurred in a certain surveillance area. This means IPCs are not continuously sending raw data to the system network as a normal (not intelligent) camera. Hence, there are different RVS arrival times, so we model RVS arrivals in our simulation as random variables with Poisson distribution [40], with Lambda equal to 10. This value is chosen arbitrarily. The Poisson distribution is particularly appropriate if the arrivals are from several independent sources in a certain time interval [41], which is the case in our simulated system. Each IPC

Table 1
Random variables for packets.

Packet type	Packet size		Processing effort (MI)		
	Distribution	Value	Distribution	Value	
		λ		n	p
RVS	Poisson	20	Binomial	300	0.75
LRD	Poisson	10	Binomial	150	0.85
ADB	Poisson	15	Binomial	200	0.75
RRD	Poisson	10	Binomial	150	0.85

generates a certain number of RVS packets. Hence, we model the number of RVS packets generated by each IPC as a random variable with discrete uniform distribution [42], in the range value from 5 to 15; these values are also chosen arbitrarily. The discrete uniform distribution is typically used when it is believed that the value is equally likely over a bounded interval [41]. In a realistic FHSS, packets vary regarding the payload (content data) they carry. Therefore, we model packet size (for each packet type) as a random variable with Poisson distribution, as shown in Table 1.

Consequently, if the packets sizes are random, the efforts required for processing these packets are also random. Therefore, we model packet processing efforts (for each packet type) as a random variable with Binomial distribution [43], as shown in Table 1. The Binomial distribution is typically used to model the number of successes in a sequence of n independent and identical Bernoulli trials [41]. In our simulation, we need to model several identical instructions that successfully handle the processing of a certain type of packet. Hence, we believe that binomial distribution is the best fit for this case.

4. Experimental results

This section presents the experimental simulation results of using the proposed PEFM solution as a privacy control for protecting sensitive Foscam data from privacy violations. In our experiments, PEFM must assure that no data is delivered to any party (incl. data owner) without enforcing privacy policies on these data. Hence, the simulation results are collected for risk scenarios and non-risk scenarios as well. Since executing PEFM for non-risk scenarios does not include data minimization, the results for running PEFM for these scenarios is considered to be the baseline results. The following two subsections present the system performance results in terms of latency and throughput for executing the two major PEFM's modules—LPEM and RPEM, for local and remote Foscam scenarios, respectively. For each PEFM's module, the absolute and relative latency and throughput results are presented for using that module as a privacy control for certain Foscam scenarios. The results are obtained for five different system workloads (Configurations). For each workload, the absolute values show the system performance results of all (local or remote) scenarios. While the relative values show the system performance differences results of each scenario compared with the results of the baseline scenario.

4.1. Results of using LPEM for local Foscam scenarios

4.1.1. LPEM latency

The experimental results, shown in Fig. 3, show that the absolute LPEM latency for all scenarios increases with configuration complexity: linearly for *Config 1* to *Config 4*, and less than linearly for *Config 4* to *Config 5*. For each network configuration, the LPEM latency for risk scenarios (HMIS, HATS, and HHRS) is lower than the baseline scenario (HNRS). Among all configurations, LPEM latency has the highest values for HATS and the lowest values for HMIS. The graph also shows the standard deviation of LPEM latency values for all scenarios and all configurations. Fig. 4 demonstrates that, compared to the baseline scenario (HNRS), privacy enforcement in LPEM causes an average latency *improvement* of 21.65% for HMIS, 12.95% for HATS, and 17.30% for HHRS, averaged over five configurations. For each configuration, the graph shows the percentage of LPEM latency gain for this configuration for the three types of scenarios. Among different configurations, the latency gain values are close to each other. However, we observe a dip in the latency gain for *Config 3* for HATS and a slight dip for HHRS. This is an unexpected case in our results which may be due to the randomization of packet size values. Among different scenarios, LPEM latency gain is different: higher for HMIS than for HATS where the ratio of LPEM latency gain for HHRS is approximately in the middle between the two.

4.1.2. LPEM throughput

Fig. 5 shows LPEM throughput for the local scenarios for five different network configurations. For each scenario type (including the baseline), the LPEM throughput increases more than linearly with configuration complexity. Increasing the number of SHs with x-factor increases the throughput with x-factor too because more SHs means more output data packets generated. For each network configuration, the LPEM system throughput values are different for each type of scenario. HMIS increases the system throughput for all configurations. HATS decreases the system throughput for all configurations. However, for HHRS, the LPEM system throughput increases for all configurations. Fig. 6 demonstrates that, compared to the baseline scenario, privacy enforcement in LPEM causes an average throughput *improvement* of 18.45% for HMIS, and causes

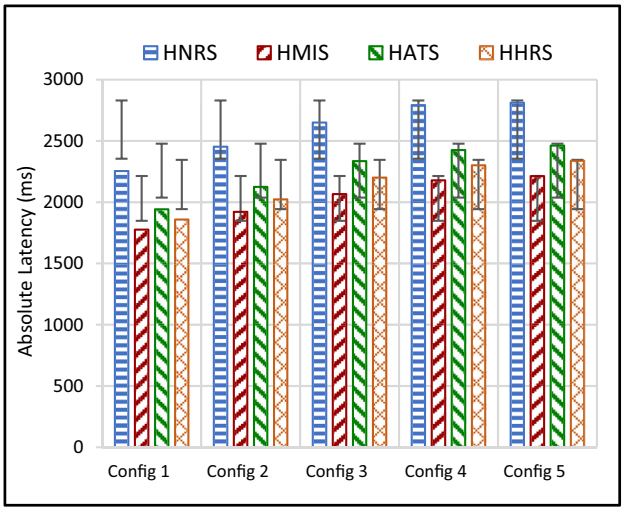

Fig. 3. Absolute LPEM latency for local scenarios.

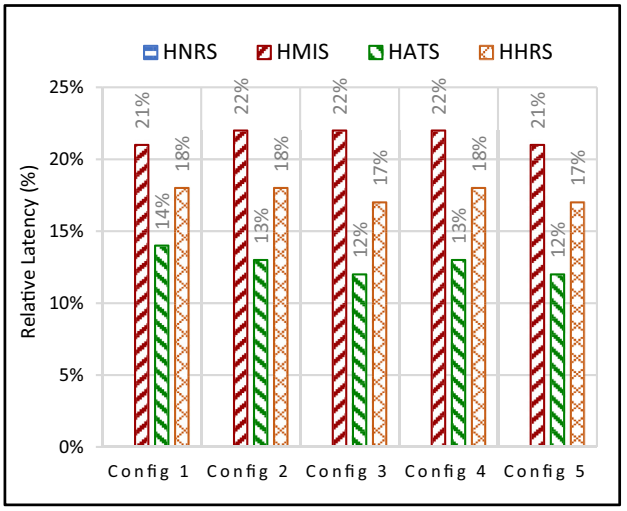

Fig. 4. Relative LPEM latency for local scenarios.

an average throughput *overhead* of 9.74% for HATS. However, LPEM causes an average throughput *improvement* of 6.06% for HHRS. Among different configurations, the throughput improvement or overhead values are close to each other.

4.2. Results of using RPEM for remote Foscam scenarios

4.2.1. RPEM latency

The experimental results, shown in Fig. 7, show that the absolute RPEM latency for all scenarios increases linearly with configuration complexity. For all network configurations, the RPEM latency for risk scenarios (CMIS, CATS, and CHRS) is higher than its latency for the baseline scenario (CNRS). However, among all configurations, RPEM latency values for risk scenarios are close to each other. In general, increasing the system workload over many configurations results in an observable increase in RPEM latency for all remote scenarios. The graph also shows the standard deviation of RPEM latency values for all scenarios and for all configurations. Fig. 8 demonstrates that, compared to the baseline scenario (CNRS), privacy enforcement in RPEM causes the average latency *overhead* of 11.51% for CMIS, 11.09% for CATS, and 11.30% CHRS, averaged over the five configurations. For each configuration, the graph shows the percentage of RPEM latency overhead for the risk scenarios. The overhead values slightly decrease with configuration complexity for all risk scenarios.

4.2.2. RPEM throughput

Fig. 9 shows that, for all remote scenarios, RPEM throughput increases more than linearly with configuration complexity. However, for each configuration, the RPEM throughput values are different for each type of scenario. Compared to

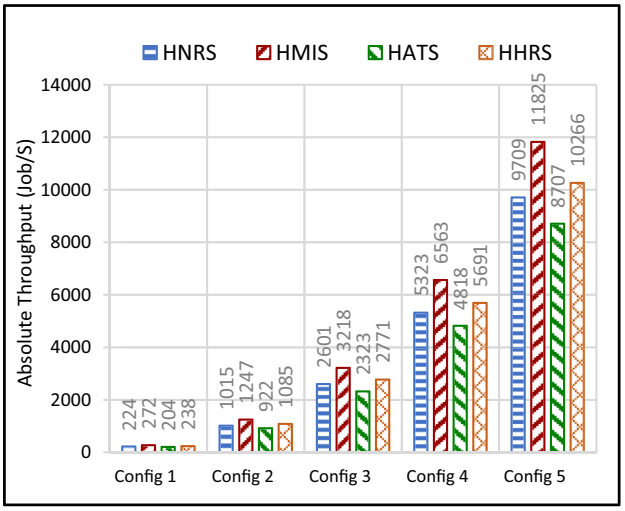

Fig. 5. Absolute LPEM throughput for local scenarios.

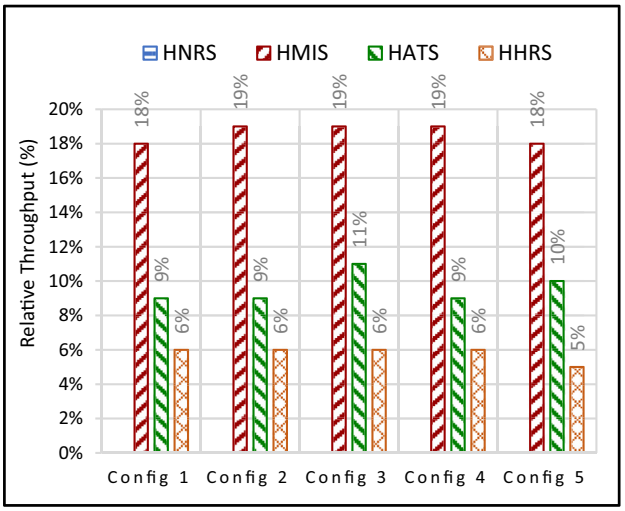

Fig. 6. Relative LPEM throughput for local scenarios.

the throughput of the baseline scenario (CNRS); CMIS slightly decreases the throughput, CATS significantly decreases the throughput, and CHRS decreases the throughput too. Fig. 10 demonstrates that, compared to the baseline scenario, privacy enforcement in RPEM causes the average throughput *overhead* of 15.33% for CMIS, 32.81% for CATS, and 24.07% for CHRS, averages over five configurations. Among all configurations; CMIS has the highest overhead values, CHRS has the lowest overhead values, and CATS has overhead values slightly higher than for CHRS and lower than for CMIS.

5. Results discussion

5.1. LPEM privacy and the corresponding latency and throughput

Our experiments show that LPEM can assure mandatory enforcement of privacy policies for all outgoing data of the FHSS system. In other words, no data can be delivered to a local application without enforcing privacy policies for these data by LPEM. The enforcement process, in turn, assures fine-grained access control for data and selective data disclosures (our privacy objectives discussed in Section 1.3). Since LPEM adds extra processing time to the basic fog computing processing time done by the fog node (Foscam BS), overhead in terms of latency and throughput is expected. However, our experimental results show that comparing risk scenarios (HMIS and HHRS) with the baseline scenario (HNRS), assuring data privacy by LPEM decreases system latency and increases system throughput. This is mainly due to data minimization performed by

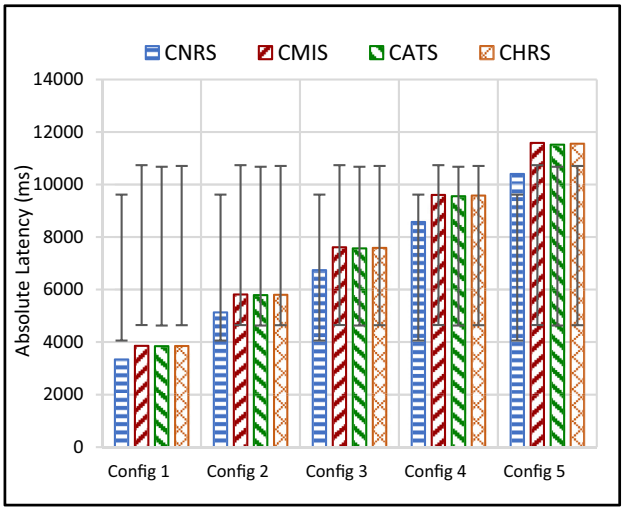

Fig. 7. Absolute RPEM latency for remote scenarios.

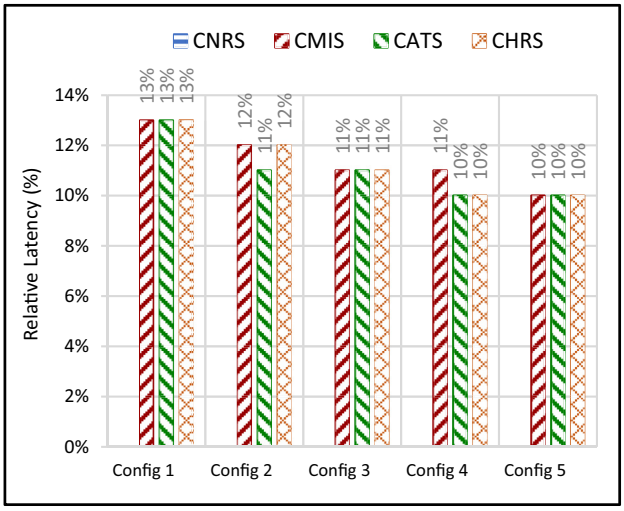

Fig. 8. Relative RPEM latency for remote scenarios.

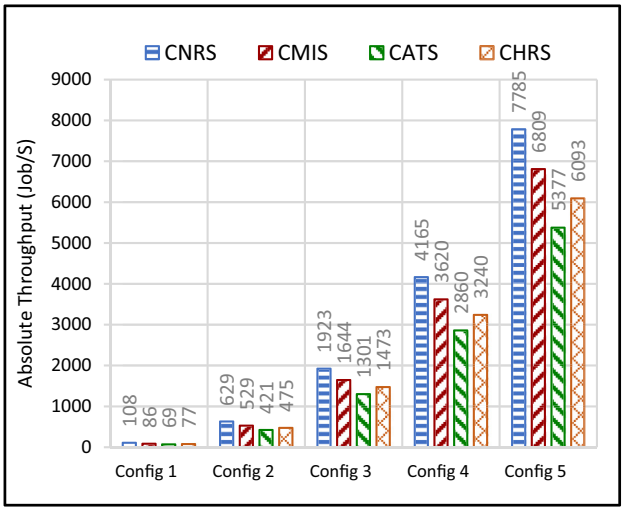

Fig. 9. Absolute RPEM throughput for remote scenarios.

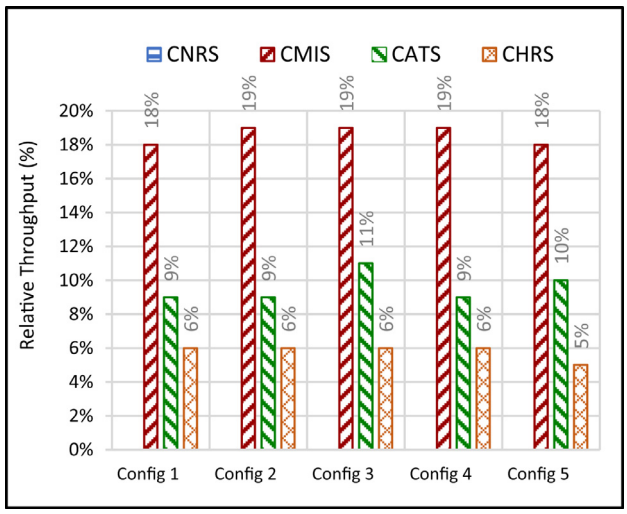

Fig. 10. Relative RPEM throughput for remote scenarios.

LPEM during selective data disclosures. For HATS, assuring data privacy by LPEM also decreases system latency, but it also decreases system throughput as well since the number of delivered packets decreases by null disclosures. All packets in the baseline scenario (HNRS) are fully disclosed to the local applications and hence they take the full time for transmission, and no reduction can happen in terms of packet size and the number of transmitted packets. While for HMIS, some packets are partially disclosed to the local applications that might include malicious insiders. This means there is a reduction in packet size, but there is no reduction in the number of transmitted packets. In this case, latency is decreased, and throughput is increased because the same number of packets as for the baseline is transmitted with less latency. For HATS, some packets are null disclosed to the local applications that might include attackers. This means there is a reduction in the number of transmitted packets. In this case, latency is also decreased since the time required to transmit fewer packets is probably lower than the time required for all packets, and clearly, the throughput is decreased. Among different configurations, the LPEM latency gain for a certain scenario is close to each other. Among different scenarios, LPEM latency gain is different: higher for HMIS than for HATS and HHRS.

Regarding LPEM latency, we observe that even with increasing system workload, the LPEM latency is not increased proportionally, but its values stay close to each other with only a slight increase. This is due to the distribution feature of fog computing where each SH has an independent fog node (BS) responsible for processing and transmitting the data of this SH. However, increasing system workload is proportionally increasing system throughput. In other words, increasing the number of smart homes (SHs) with x-factor increases the throughput with x-factor too because more SHs means more output data packets generated.

In general, since LPEM policy enforcement is handled at the local fog node and each SH has a fog node, the complexity should not increase, and our simulation results support that hypothesis. The results show that running LPEM for FHSS improves both privacy and system performance. This is a promising result for the feasibility and efficiency of PEFM system for local FHSS scenarios.

5.2. RPEM privacy and the corresponding latency and throughput

The experimental results show that RPEM can assure mandatory enforcement of privacy policies for FHSS data whenever these data are accessed in remote domains. All data packets sent by RPEM are Active Data Bundles (ADBs) which are active packets able to protect themselves from privacy violations. ADB assures that no data can be delivered to a remote application without enforcing privacy policies for these data by the ADB's VM. This assures portable fine-grained access control for disseminated data, selective data disclosures, remote data destruction, and increasing data owners' control for their transmitted data (our privacy objectives discussed in Section 1.3). For all configurations, the RPEM latency for risk scenarios is higher than the baseline because assuring privacy by LPEM requires sending data as a form of ADB. Each ADB carries the original data in addition to the policies and the policy enforcement engine (ADB's VM) which requires more processing and transmission times than the original data packets. For all configurations, RPEM throughput for risk scenarios is lower than the baseline due to ADB apoptosis. Therefore, we have an overhead in terms of latency and throughput and this overhead is expected in our model since we should have a cost for a high-level of privacy protection. However, the experimental results show that the overhead is acceptable; 11.3% average latency overhead and 24.07% throughput overhead. This overhead is mainly due to the remote scenarios, where ADBs are sent over all hops from IPCs to Cloud among intermediate fog nodes with less data minimization than for local scenarios. This is because the enforcement is done only at the final hop—from

Cloud to remote devices, so all ADBs are sent with their full payloads and take full time for transmission and processing on their journey from source BSs to Cloud. This distinguishes RPEM from LPEM where data minimization is done at the local fog level for the later.

Regarding RPEM latency, we observe that increasing system workload, among different configurations, proportionally increases system latency for remote scenarios. This is different than the case for local scenarios because using Cloud is more centralized than using distributed fog computing. This highlights the advantage of using fog computing for our solution. Among different scenarios, the RPEM latency overhead values for a certain configuration are close to each other because for all scenarios ADBs are sent and take the full time, as explained earlier. However, the overhead values decrease with configuration complexity. The reason for this is that when the system workload increases, the computations also increase. Therefore, the time required for enforcing privacy policies by RPEM is proportionally low concerning the highest computations in *Config 5*. While for *Config 1*, the computations of the system are low so the time for RPEM processing will be more observable than for *Config 5*.

Regarding RPEM throughput, increasing system workload proportionally increases system throughput. In other words, increasing the number of smart homes (SHs) with *x*-factor increases the throughput with *x*-factor too because more SHs means more output data packets generated. For all configurations, RPEM throughput decreases with configuration complexity due to latency overhead. CMIS slightly decreases the throughput since the number of processed packets is the same as for the baseline (CNRS), but the latency for them is slightly higher. CATS significantly decreases the throughput since the number of processed packets is much less for the baseline due to the null disclosure for all attackers in the scenario.

Finally, since RPEM policy enforcement is not handled at the local fog node, the complexity should increase, and our simulation results support that hypothesis. The average RPEM overhead is the price paid for privacy protection. So, we have a trade-off of overhead versus privacy. For some applications, this price might be acceptable while for others it might not. For example, safety applications like fire detection or gas leak detection do not accept any extra overhead. However, entertainment applications might accept the introduced overhead. The results show that running RPEM for FHSS improves privacy with reasonable overhead for system performance. This establishes the feasibility and efficiency of PEFM system for remote FHSS scenarios.

6. Conclusions

The rapid evolution of IoT regarding services and applications, results in collecting a huge amount of data about people and their surroundings. IoT data most likely have PII since they reflect, directly or indirectly, people's activities in continuously and timely manners. We recognize these data as major threats that are vulnerable to attackers and malicious insiders to violate the privacy of IoT users, so preserving data privacy is key for many IoT applications that include collecting PII.

In this paper, we propose a privacy-preserving software module to be placed at the fog level as close as possible to IoT sensors, called Policy Enforcement Fog Module (PEFM). It performs mandatory enforcement of privacy policies for all outgoing IoT data to assure that no data can be accessed without preserving the privacy for them. From a fog computing point of view, we have distinguished two types of IoT applications. For local applications, Local Policy Enforcement Module (LPEM), a part of the PEFM, is proposed to enforce policies directly within local fog node based on XACML access control standard. For remote applications, Remote Policy Enforcement Module (RPEM), the second part of the PEFM, is proposed to set up a self-protecting mechanism called ADB that is designed to be migrated to remote nodes and enforces privacy policies for the carried data whenever it accessed throughout the entire data lifecycle.

To test the performance of PEFM in a realistic IoT environment, we have introduced a proof-of-concept case study named Foscam Home Surveillance System (FHSS). The privacy threats of FHSS are identified, and the framework for using the PEFM as a privacy control for these threats is presented. A comprehensive simulation is implemented for eight different experimental FHSS scenarios; four local scenarios, where LPEM is the privacy actor, and four remote, where RPEM is the privacy actor.

The experimental results show that PEFM's LPEM assured privacy for data accessed within local domains with improved system performance regarding latency and throughput. PEFM's RPEM assured privacy for data accessed within remote domains with a reasonable system performance overhead regarding latency and throughput. In general, the results show that better privacy controls with minimal overhead can be achieved if most PEFM processes are executed by the local fog nodes. Migrating parts of PEFM processes to remote fog nodes or the cloud incurs more overhead than using strictly local fog nodes. This overhead is the price to be paid for a higher level of privacy in terms of lifecycle data protection. So, there is a tradeoff between overhead and the desired level of privacy. The overhead should be acceptable by applications that are not time-sensitive with hard deadlines.

Declaration of interests

The authors declare that they have no known competing financialinterestsor personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Leszek Lilien for valuable comments during the execution of this research work. This research was supported in part by the NIGMS of NIH under award number R15GM120820. Authors thank the reviewers for their valuable comments.

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, S. Woelfflé, Vision and challenges for realising the Internet of Things, in: Proceedings of Cluster of European Research Projects on the Internet of Things (CERP-IoT), European Commission, 2010, doi:10.2759/26127.
[2] Cisco, Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are, White paper, San Jose, CA, 2017 Accessed on March 26from: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf .
[3] E.M. Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G.J. Ren, J. Zhu, and J. Farre, "What is a Fog Node? A Tutorial on Current Concepts Towards a Common Definition," vol. arXiv:1611.09193v1, 2016.
[4] E. McCallister, T. Grance, S. Kent, Guide to Protecting the Confidentiality of Personally Identifiable Information (PII): Recommendations of the National Institute of Standards and Technology, National Institute of Standards and Technology, NIST Special Publication, 2010 800-122.
[5] "Trends in Encryption and Data Security" (Slides), Cloud, Big Data and IoT Edition, Vormetric Data Threat Report, Vormetric Data Security, San Jose, CA, 2016.
[6] A. Al-Gburi, A. Al-Hasnawi, L. Lilien, et al., Differentiating security from privacy in Internet of Things—a survey of selected threats and controls, in: K. Daimi, et al. (Eds.), Computer and Network Security Essentials, Springer, 2018, pp. 153–172.
[7] L. Ben Othmane, Active Bundles for Protecting Confidentiality of Sensitive Data Throughout Their Lifecycle, Department of Computer Science, Western Michigan University, Kalamazoo, MI, Dec. 2010.
[8] D. Ferraiolo, R. Chandramouli, R. Kuhn, V. Hu, Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC), in: Proceedings of ACM International Workshop on Attribute-Based Access Control (ABAC), New Orleans, LA, 2016, pp. 13–24.
[9] R. Ranchal, Cross-domain Data Dissemination and Policy Enforcement, Purdue University, West Lafayette, Indiana, 2015.
[10] L. Ben Othmane, L. Lilien, Protecting privacy in sensitive data dissemination with active bundles, in: Proceedings of the Seventh Annual Conference on Privacy, Security and Trust (PST), Saint John, New Brunswick, Canada, 2009, pp. 202–213.
[11] Internet of Things Privacy Forum, "Clearly Opaque: pPrivacy Risks of the IoT," 2018.
[12] A. Al-Gburi, A. Al-Hasnawi, L. Lilien, et al., Differentiating security from privacy in Internet of Things—a survey of selected threats and controls, in: K. Daimi, et al. (Eds.), Computer and Network Security Essentials, Springer, 2018, pp. 153–172.
[13] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, SPINS: security protocols for sensor networks, Wirel. Netw. 8 (5) (2002) 521–534.
[14] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), Alexandria, VA, 2006, pp. 89–98.
[15] L. Malina, J. Hajny, R. Fujdiak, J. Hosek, On perspective of security and privacy-preserving solutions in the internet of things, Comput. Netw. 102 (2016) 83–95.
[16] PCI Security Standards Council, Initial roadmap: point-to-point encryption technology and PCI DSS compliance, Emerg. Technol. (2010) 1–16. Whitepaper, Version 1.0, Accessed on Sep. 10, from: https://www.pcisecuritystandards.org/documents/pci_ptp_encryption.pdf .
[17] L. Sweeney, k-anonymity: a model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (05) (2002) 557–570.
[18] A. Pfitzmann and M. Hansen, "A Terminology for Talking About Privacy by Data Minimization: Aanonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," Version v0.34, 2010.
[19] J. Li, M.H. Au, W. Susilo, D. Xie, K. Ren, Attribute-based signature and its applications, in: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS), Beijing, China, 2010, pp. 60–69.
[20] D. Chaum, E. Van Heyst, Group signatures, in: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques, Advances in Cryptology, EUROCRYPT, Lecture Notes in Computer Science (LNCS), Springer-Verlag, Berlin, Germany, 1991, pp. 257–265. LNCS 547.
[21] Y. Zheng, Digital signcryption or how to achieve cost (signature & encryption) cost (signature) + cost(encryption), in: Proceedings of International Cryptology Conference on Advances in Cryptology, CRYPTO, Springer, Heidelberg, 1997 LNCS 1294.
[22] F. Kargl, F. Schaub, S. Dietzel, Mandatory enforcement of privacy policies using trusted computing principles, in: Proceedings of AAAI Spring Symposium (Intelligent Information Privacy Management), Stanford University, Stanford, CA, 2010, pp. 104–109.
[23] C. Dsouza, G.J. Ahn, M. Taguinod, Policy-driven security management for fog computing: preliminary framework and a case study, in: Proceedings of 15th IEEE International Conference on Information Reuse and Integration (IRI), Redwood City, CA, 2014, pp. 16–23.
[24] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, B. Amos, Privacy mediators: helping IoT cross the chasm, in: Proceedings of the 17th ACM International Workshop on Mobile Computing Systems and Applications (HotMobile), St. Augustine, FL, 2016, pp. 39–44.
[25] O. Sibert, D. Bernstein, D. Van Wie, The DigiBox: a self-protecting container for information commerce, in: Proceedings of the First USENIX Workshop on Electronic Commerce, New York, NY, 1995, p. 15.
[26] R. Neisse, G. Baldini, G. Steri, Y. Miyake, S. Kiyomoto, A.R. Biswas, An agent-based framework for informed consent in the Internet of Things, in: Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT), Milan, Italy, 2015, pp. 789–794.
[27] A. Tchao, G. Di Marzo, J.H. Morin, et al., Personal DRM (PDRM) – a self-protecting content approach, in: F. Hartung, et al. (Eds.), Digital Rights Management: Technology, Standards and Applications, CRC Press, Taylor & Francis Group, New York, NY, 2017.
[28] "Foscam home security." 2007–2019. Accessed on July 28, 2018 from: https://www.foscam.com/.
[29] J. Carlsen, "Best Home Surveillance Systems of 2018." 2018. Accessed on Sep. 28, 2018 from: https://www.toptenreviews.com/home/smart-home/best-home-surveillance-systems/.
[30] A. Jacobsson, M. Boldt, B. Carlsson, A risk analysis of a smart home automation system, Futur. Gener. Comput. Syst. 56 (2016) 719–733.
[31] "Foscam Home Security- FOSCAM E1." 2007–2019. Accessed on Sep. 28, 2018 from: https://www.foscam.com/E1.html.
[32] A. Jacobsson, M. Boldt, B. Carlsson, A risk analysis of a smart home automation system, Futur. Gener. Comput. Syst. 56 (2016) 719–733.
[33] D. Christin, A. Reinhardt, S.S Kanhere, M. Hollick, A survey on privacy in mobile participatory sensing applications, J. Syst. Softw. 84 (11) (2011) 1928–1946.
[34] M.J. O'Mahony, S. Dimitra, K.H. David, T. Anna, The application of optical packet switching in future communication networks, IEEE Commun. Mag. 39 (3) (2001) 128–135.
[35] H. Gupta, A. Vahid Dastjerdi, S.K. Ghosh, R. Buyya, iFogSim: a toolkit for modeling and simulation of resource management techniques in the Internet of Things, edge and fog computing environments, Softw.: Pract. Exp. 47 (9) (2017) 1275–1296.
[36] A. Khanna, T. Ravi, IoT based interactive shopping ecosystem, in: Proceedings of the 2nd IEEE International Conference on Next Generation Computing Technologies (NGCT), 2016, pp. 40–45.
[37] M. Taneja, D. Alan, Resource aware placement of IoT application modules in fog-cloud computing paradigm, in: Proceedings of the IFIP/IEEE Symposium on Integrated Network and Service Management (IM), 2017, pp. 1222–1228.
[38] "Instructions per second," Wikipedia. Accessed on May 6, 2018, from: https://en.wikipedia.org/wiki/Instructions_per_second.
[39] J. Kurose, K. Ross, Computer Networking: A Top-Down Approach, 7th ed., Pearson, Boston, MA, 2016.
[40] "Poisson Distribution." 2008–2018. Accessed on Sep. 19, 2018, from: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.poisson.html/numpy.random.poisson.

[41] J. Raj, Art of Computer Systems Performance Analysis Techniques for Experimental Design Measurements Simulation and Modeling, Wiley Computer Publishing, John Wiley & Sons, Inc., 1991.
[42] "Discrete Uniform Distribution." 2008–2018. Accessed on Sep. 19, 2018, from: https://docs.scipy.org/doc/scipy/reference/tutorial/stats/discrete_randint.html.
[43] "Binomial Distribution." 2008–2018. Accessed on Sep. 19, 2018, from: https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.binomial.html#numpy.random.binomial.