Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation

Prem Prakash Jayaraman [a,*], Xuechao Yang [b], Ali Yavari [b], Dimitrios Georgakopoulos [a], Xun Yi [b]

[a] Swinburne University of Technology, Melbourne, Victoria, 3122, Australia

[b] RMIT University, Melbourne, Victoria, 3000, Australia

ARTICLE INFO

Article history:
Received 16 March 2016
Received in revised form
16 January 2017
Accepted 1 March 2017
Available online xxxx

Keywords:
Internet of Things
IoT privacy and security
Secure IoT platform

ABSTRACT

The Internet of Things (IoT) is the latest web evolution that incorporates billions of devices that are owned by different organisations and people who are deploying and using them for their own purposes. IoT-enabled harnessing of the information that is provided by federations of such IoT devices (which are often referred to as IoT things) provides unprecedented opportunities to solve internet-scale problems that have been too big and too difficult to tackle before. Just like other web-based information systems, IoT must also deal with the plethora of Cyber Security and privacy threats that currently disrupt organisations and can potentially hold the data of entire industries and even countries for ransom. To realise its full potential, IoT must deal effectively with such threats and ensure the security and privacy of the information collected and distilled from IoT devices. However, IoT presents several unique challenges that make the application of existing security and privacy techniques difficult. This is because IoT solutions encompass a variety of security and privacy solutions for protecting such IoT data on the move and in store at the device layer, the IoT infrastructure/platform layer, and the IoT application layer. Therefore, ensuring end-to-end privacy across these three IoT layers is a grand challenge in IoT. In this paper, we tackle the IoT privacy preservation problem. In particular, we propose innovative techniques for privacy preservation of IoT data, introduce a privacy preserving IoT Architecture, and also describe the implementation of an efficient proof of concept system that utilises all these to ensure that IoT data remains private. The proposed privacy preservation techniques utilise multiple IoT cloud data stores to protect the privacy of data collected from IoT. The proposed privacy preserving IoT Architecture and proof of concept implementation are based on extensions of OpenIoT - a widely used open source platform for IoT application development. Experimental evaluations are also provided to validate the efficiency and performance outcomes of the proposed privacy preserving techniques and architecture.

1. Introduction

"Internet of Things" (IoT) is the latest Internet evolution that involves (i) incorporating billions of internet-connected sensors, cameras, displays, smart phones, wearable, and other smart devices that communicate via the internet (which are collectively referred to as IoT things), and (ii) harnessing their data and functionality to provide novel smart services and products that benefit our society. A recent forecast made by the Gartner projects Internet of Things and the associated ecosystem to be a $1.7 trillion market by 2020 and include 28.1 billion connected things [1]. IoT is fuelling a paradigm shift of a truly connected world in which everyday objects become interconnected, able to communicate directly with each another, and capable of collectively providing smart services [2]. However, in many such applications [3] the data collected by IoT is sensitive and must be kept private and secure. Examples of sensitive IoT data include physiological data collected by (in some cases wearable) biomedical sensors, energy consumption data collected by smart meters, and location data collected by mobile phones to name just a few. The disclosure of such data may create opportunities for criminal activity, or result in serious harm or even death. Therefore from such a perspective, IoT presents a significant challenge for security, privacy and trust, which are considered to be among the remaining main barriers in IoT application development.

[*] Corresponding author.
E-mail address: prem.jayaraman@gmail.com (P.P. Jayaraman).

http://dx.doi.org/10.1016/j.future.2017.03.001
0167-739X/© 2017 Published by Elsevier B.V.

Most of existing solutions for protection of privacy-sensitive data in IoT focus on communication channels security and authorisation. Little work has been done to protect sensitive sensor data after they are collected, integrated and stored. This creates opportunities for both hackers and malicious administrators to steal and disclose privacy-sensitive data collected and distilled from IoT devices. To protect such privacy-sensitive data against hacking, we need to develop an IoT platform/infrastructure that ensure end-to-end privacy and security (i.e., starting from the point of data collection from IoT devices thought the point of data harnessing for delivering IoT applications and/or related services).

In this paper we introduce a novel privacy-preserving technique and a related IoT architecture that are designed to protect sensitive IoT sensor data from disclosure and hacking [4,5]. The basic idea in this technique involves two steps. First, each data item x that is collected from an IoT device is randomly transformed to a sum of n numbers, i.e., $x = x_1 + x_2 + x_n(n \geq 2)$, and each addend $x_1, \ldots, x_n$ is stored in a different data store. Therefore, our solution requires the use of n data stores (in a server or/and the cloud) and each data store D_i keeps only the addend x_i of $x(n \geq i \geq 1)$. Second, we also propose the introduction of a homomorphic encryption scheme that allows access to the data collected from IoT devices via the aggregation of their addends, and hence without the risk of exposing sensitive data to hackers or to data store administrators. Even in cases where all but one of the n data stores are compromised, the IoT data remains private. Specifically, this paper makes the following contributions:

- Introduces a privacy-preserving technique for controlling access to sensitive IoT data via decomposing sensitive data to addends that are stored in multiple data stores and then (re)aggregating the IoT data when is requested by a user without exposing any anything beyond meaningless addends.
- Proposes a blueprint for a privacy preserving IoT architecture that provides end-to-end privacy based on the proposed privacy-preserving data access scheme.
- Describes a proof-of-concept system prototype implementation and evaluates its efficiency.

The rest of the paper is organised as follows. In Section 2, we present a survey of the current state-of-the-art in IoT security. In Section 3, we propose and formulate the proposed security technique. In Section 4, we present the blueprint architecture the IoT system that implements the proposed security technique implementation. In Section 5, we present experimental evaluations of the developed system. Section 6 concludes the paper.

2. Related work

IoT is an important new internet technology with great potential for developing smart buildings and cities, assisted living and healthcare, precision agriculture and environmental monitoring, manufacturing, as well as for security and defence [6]. IoT systems and their applications must deal with malicious information disclosure and provide techniques that protect sensitive data, such as patient data in healthcare, energy consumption data from smart energy meters, and location data. IoT poses the following privacy challenges that define the need for novel privacy and data protection techniques [7,8]:

- Lack of control over IoT devices,
- Inferences derived from collected data,
- Pattern extraction from anonymous data, and
- Privacy loss across IoT layers, e.g., devices, infrastructure storage, applications, and related communications.

The need for security and privacy solutions is also highlighted by IoT forensic researchers [9–11].

Existing techniques for protecting sensitive data in IoT have mainly focused on securing the communication channel, as well as user authentication and authorisation. However, there is a significant gap in developing techniques that can ensure privacy in the collection, storage, and retrieval (providing computed aggregations without exposing the data) of IoT data.

2.1. Communication channel security mechanisms

To secure communication in IoT, it is important to encrypt data communicated between IoT devices, gateways and other IoT infrastructure due to the public nature of the Internet. Keys for encryption must be agreed upon by communicating nodes [12,13]. Due to resource constraints, key agreement in IoT is non-trivial. Many key agreement schemes used in general networks, such as Kerberos [14] and RSA [15], may not be suitable for IoT because there is usually no trusted infrastructure in IoT. Pre-distribution of secret keys for all pairs of nodes is not viable due to the large amount of memory used when the network size is large. To overcome this problem, a random key pre-distribution scheme [16] was proposed, where each sensor node receives a random subset of keys from a large key pool before deployment and any two nodes can find one common key within their subsets and use that key to secure the communication. Without requiring any key pre-distribution, data sensed within IoT has been used to establish the common secret key. For example, in [17], two sensors S1 and S2 in a Body Sensor Network (BSN) use the common electrocardiogram (EKG) signals of a patient to establish a secret key.

Roman et al. [18], Du et al. [19] and Camtepe et al. [20] analyse the applicability of several link-layer oriented key management systems (KMS), which establish keys for sensor nodes within the same WSN using techniques such as linear algebra, combinatorics and algebraic geometry. However, the authors mention not all mathematical-based KMS protocols can fulfil the IoT context, according to the analysis result, only [19,21] might be suitable for some IoT scenarios. At the end of the paper, the authors recommend to use a trusted third party to enable other key management mechanisms.

OSCAR [22] is a more recent approach for end-to-end security of IoT. It is based on the concept of object security and focuses on securing the message payload to enable secure M2M communication. The novel aspect of OSCAR is the use of cryptography techniques over CoAP protocol to ensure lightweight and scalable encryption. Similarly in [23], the authors propose a lightweight method using (Internet Protocol Security) IPsec for securing end-to-end communication channel between unconstrained peers and IoT devices (constrained). The proposed method makes it possible for an unconstrained node to set up an IPsec-ESP Transport Mode connection with an IoT device while moving the master session key generation and authentication processes from the IoT device to the trusted gateway. The ESP mode (that provides data encryption and authentication) allows the setup of an end-to-end secure connection between two peers by encrypting the payload, therefore, the proposed method relieves the IoT devices from the computational burden associated with the generation of cryptographic data. Under the proposed method, IoT devices can benefit from higher level of cryptosystem without executing the intensive computation. In [24], the authors propose a novel secure and scaled IoT storage system to tackle the aforementioned issues at both data and system levels, which is based on Shamir's secret sharing scheme. There are mainly four components in the proposed system: client, dispatcher, peer managers and regular peers, and the system is organised into three layers: (1) file saving and restoration, (2) connection setup and data transfer, and (3) share replication.

They provide a secret sharing mechanism in order to eliminate complex key management. Both [23,24] focus is on using encryption methods to protect the communication layer (device to cloud and cloud to user).

2.2. Authorisation and access control

As a large amount of sensed data are stored in sensor nodes or database, it is important to control access to the data [25]. Attribute-based encryption (ABE) [26] has been used to control access to sensor data in [27,28]. In traditional public-key cryptography, a message is encrypted for a specific receiver using the receiver's public-key. Identity-based encryption (IBE) [29] changed the traditional understanding of public-key encryption by allowing the public-key to be an arbitrary string, e.g., the email address of the receiver. ABE goes one step further and defines the identity not atomic but as a set of attributes, e.g., roles, and messages can be encrypted with respect to subsets of attributes (key-policy ABE-KP-ABE) or policies defined over a set of attributes (ciphertext-policy ABE-CP-ABE). The key issue is, that someone should only be able to decrypt a ciphertext if the person holds a key for matching attributes. User keys are issued by some trusted party.

In CP-ABE, a user's private-key is associated with a set of attributes and a ciphertext specifies an access policy over a defined universe of attributes within the system. A user will be able to decrypt a ciphertext, if and only if his attributes satisfy the policy of the respective ciphertext. Policies may be defined over attributes using conjunctions, disjunctions. For instance, let us assume that the universe of attributes is defined to be A = General, B = Nurse, C = Doctor, D = Specialist and User 1 receives a key to attributes A, B and User 2 to attribute D. If a ciphertext is encrypted with respect to the policy (AC)D, then User 2 will be able to decrypt, while User 1 will not be able to decrypt. In KP-ABE, an access policy is encoded into the users secret key, e.g., (AC)D, and a ciphertext is computed with respect to a set of attributes, e.g., A, B. In this example the user would not be able to decrypt the ciphertext but would for instance be able to decrypt a ciphertext with respect to A, C. Based on KP-ABE, a Fine-grained Distributed data Access Control scheme, namely FDAC, was proposed for IoT in [27]. FDAC is resistant against user collusion, i.e., the cooperation of colluding users will not lead to the disclosure of additional sensor data. Based on CP-ABE, another fine-grained access control scheme for IoT was proposed in [28]. This scheme allows AND-based policies only.

Hu et al. [30] propose an identity based system, which protects location information of IoT devices during emergency situations. In the approach, each user communicates with others using Virtual Identity (VID), which does not contains any real information about the user. Under this architecture, users' privacy can be protected well because they only send VID(s) to communicate, and VID is anonymous and un-linkable to users. The location information will finally be sent to the user making a request only after verification of their identities. In IoT, verifying identities of IoT devices are crucial to prevent unauthorised access to user's private data, and enable access to only legitimate users. Liu et al. [31] propose an authentication protocol for IoT systems. Under the proposed protocol, IoT devices are end nodes, and each node has a unique global address for connecting over the Internet. To establish a session key, both secrete-key cryptosystems (SKC) and public-key cryptosystems (PKC) have been considered for IoT environments, but they all suffers several problems, such as SKC requires large memory to store key chains and PKC suffers from high energy consumption. Kalra et al. [12] propose an ECC (Elliptic Curve Cryptosystem) based key establishment method suitable for IoT environment. The analysis results indicate the proposed protocol can prevent eavesdropping, man-in-the middle, key control attack, and replay attacks.

2.3. Privacy preservation

In existing sensor network-based privacy solutions, such as those provided by CodeBlue [32], ALARM-NET [33] and MEDiSN [34], the sensitive data collected by IoT sensors is stored in a database for users to access and analyse. All data is encrypted during the transmission and decrypted in the data store. However, such existing sensor network solutions require complete trust of the data store and cannot guarantee privacy in cases where malicious data store administrators and hackers compromise privacy by disclosing such sensitive IoT data.

IoT devices using existing biometric-based key agreement protocols, such as [17], establish cryptographic keys by using unique biometric data, such as EKG data, obtained from the data owner. So far, the security of such protocols has not been analysed under any formal security model. It is currently unclear if there is any security weakness in these protocols.

Existing authenticated broadcast protocols [35], require IoT sensors authenticate the broadcast data with the key disclosed by their gateway in next time interval. This causes a delay in the authentication, and each sensor has to keep all unauthenticated packets in its buffer. Even in case where a delay is acceptable, IoT devices usually have limited buffer space. Furthermore, there is no privacy protection for the broadcast messages and no formally proven security model.

In the existing ABE-based access control scheme [27,28], sensors need to encrypt IoT data with ABE schemes. The encrypted data can be decrypted only by the user who meets the access control policies. However, ABE schemes [26] typically produce a high computation and are difficult to implement in wireless sensors and IoT devices with limited power and computation capabilities.

Interaction-based privacy preservation frameworks, e.g., [36], are based on the strategies for restricting the non-authorised operations, and neutralising the execution of non-authorised operations. This privacy preservation approach uses privacy protection levels in order to restrict access to sensitive data. This prevents non-authorised operations on IoT data.

Utility-aware privacy preservation techniques, e.g., as proposed in [37], are based on a negotiation approach where the consumer and producer exchange privacy and utility preferences to jointly ensure user's privacy and utility of data for the producers. This approach is particularly interesting as IoT data producers benefit from negotiating a certain level of privacy with the users in order to derive utility from the data.

Public key solutions, e.g., as in the system proposed described in [38], enable data protection for IoT devices. They use IoT gateways to collect data from sensors and apply appropriate data encryption, user access control and secure transmission techniques for establishing the essential privacy and security required for sensitive data. Furthermore, there is also a growing interest to support and integrate conceptual forensic-by-design as reported in [39].

The privacy preservation technique proposed in this paper is different as it provides both privacy-preserving IoT data storage and access to such data for end-users without revealing the actual data (in the worst case malicious administrators and hackers can only get access of the meaningless addends that are used to "reaggregate" sensitive IoT data). The proposed solution takes advantage of the inherent characteristic of distributed computing i.e. avoids single point of attack/failure as compared to the current state-of-the-art presented earlier (which are primarily based on centralised data and privacy-preserving architectures). Moreover, to the best of our knowledge, none of the existing approaches provide privacy preserving access to IoT data without exposing the IoT data to a certain degree which is subject to user privacy

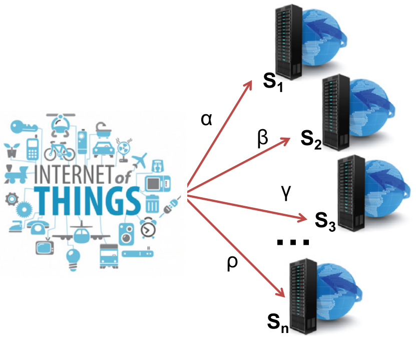

Fig. 1. IoT data ingestion model.

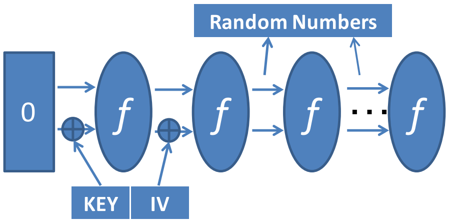

Fig. 2. Data ingestion—random number generation scheme.

settings. On contrast, the proposed technique allow operations such as aggregation, minimum, maximum, standard deviation etc., to be performed in a privacy-preserving way without exposing the raw data to the end user. Finally, no working IoT system currently uses any of the existing techniques we discussed above or currently provides data security and privacy across all IoT layers.

3. Integrated privacy protection scheme for end-to-end security

In our proposed approach [4,5], the IoT data store is composed of multiple servers. We assume that all servers are semi-honest and may try to learn about the data over time. The security requirements of our technique include:

1. Secure connection between gateway and IoT server
2. Secure persistence of data into the IoT data store
3. Privacy-preserving access to IoT data for data analysis without disclosing the data to other servers or the user.

3.1. Data ingestion scheme

Consider the IoT device produces a sequence of data (e.g. temperature, humidity) $d_1, d_2, d_3 \ldots d_m$. The proposed data ingestion scheme splits the IoT data $d_j (j = 1$ to $m)$ into n (number of servers) parts namely $\alpha_{1j}, \alpha_{2j}, \alpha_{3j} \ldots \alpha_{nj}$ termed data addends such that $d_j = \sum_{i=1}^{n} \alpha_{ij}$. Assume, the total number of server n is 3, the sensor produces a data point d_i and the three parts of the data are α, β and γ respectively. First, the sensor generates a sequence of random numbers $r_1, r_2, r_3 \ldots$ (each has 40 bits) with SHA-3 ($r = 40$ and $c = 160$) where *Key* is the random number generation secret key known only to the IoT device or the IoT gateway and the *IV* includes the current time stamp. Both *Key* and *IV* are 80 bits. Let $|\alpha_i|(|\beta_i|)$ be the first 32 bits of $r_1(r_2)$. The sign of $|\alpha_i|(|\beta_i|)$ is positive if $r_1(r_2)$ is even and otherwise negative. The sensor then computes $\gamma_i = d_i - \alpha_i - \beta_i$ for $i = 1, 2 \ldots$. The proposed data ingestion scheme is presented in Fig. 1 and the random number generation is presented in Fig. 2.

Remark. The communication channel (constrained) can be protected using any lightweight encryption scheme such as DES or approaches presented in [12]. The proposed data ingestion scheme uses minimalistic computation on the sensor node or the gateway node to split the data based on the server configuration. As long as two in a three server configuration is not compromised, the privacy of the IoT data is protected.

3.2. Data access control scheme

The data access control scheme uses a homomorphic encryption scheme provided by the Paillier cryptosystem [40] to ensure privacy-preserved access to IoT data. To access the data, the user is required to provide a public and private key pair (pk, sk) for the Paillier cryptosystem. The key could potentially be managed and issued by a Certificate Authority (CA). To request the data, the user sends a request that includes the identity of the user, the query (including time window), operation to be performed (aggregation) on the data and the public, private key pair. In our proposed scheme the focus is to provide analysed data to the end-user without exposing the actual data to both intermediate servers and the end-user. The communication channel between the end-user and the data access layer is assumed to be a secure channel and the data access layer uses some form of access control similar to [27,28]. If the user's request passes the signature verification and meets the access control policies, the data access layer will run Algorithm 1 to fetch the corresponding data.

Algorithm 1: Data access control scheme

```
Data: pk,identification,query
Result: $E(\rho)$-query result encrypted using users pk
The user provides $pk$ to the data access layer
The data access layer will pass the $pk$ to $n$ servers
For each server $S_i$ where $i = 1$ to $n$
    The server $S_i$ picks a random $r_i \in \mathbb{Z}_N^*$ and computes
    $C_i = Encrypt(\alpha_i, pk) = g_i^{\alpha} r_i^N (modN^2)$
    return $C_i$
End for
The data access layer will compute
    $E(\rho) = C_1 * C_2 * C_3 \ldots C_i$ where $i = 1$ to $n$
Return $E(\rho)$ to user
User Decryption
$\rho = Decrypt(E(\rho), sk)$
```

Due to the homomorphic properties of the Paillier cryptosystem, we have (assuming 3 servers and α, β, γ being the three parts of the data to illustrate the homomorphic property)

$$\begin{aligned} C_1C_2C_3 &= E(\alpha, pk)E(\beta, pk)E(\gamma, pk) \\ &= (g^{a}r_1^{N})(g^{b}r_2^{N})(g^{\gamma}r_3^{N})(\bmod N^2) \\ &= g^{\alpha+\beta+\gamma}(r_1r_2r_3)^N(\bmod N^2) \\ &= E(\alpha+\beta+\gamma, pk). \end{aligned}$$

Therefore, the result of this is,

$$\rho = Decrypt(C_1C_2C_3, sk) = \alpha + \beta + \gamma.$$

In order to perform statistical analysis on the IoT data, we present a scheme that allows fast privacy-preserved computation of average over collected IoT data. When the user queries the average of collected IoT data $d_1, d_2, d_3 \ldots d_x$ where $d_j = \sum_{i=1}^{n} \alpha_{ij}$ where $j = 1$ to m via the data access layer from n servers, the data access layer and the server will run Algorithm 2 to compute the average. The user query will include the identification of the user

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001

Algorithm 2: Data access - average computation

```
Data: pk,identification,query (window)
Result: AVG ($E(\rho)$)-query result encrypted using users pk
The user provides pk to the data access layer
The data access layer will pass the pk to n servers
For each server $S_i$ where $i = 1$ to n
    The server $S_i$ picks a random $r_i \in \mathbb{Z}_N^*$
    The server will query x data points based on the query window
    server computes
    $C_i = Encrypt(\sum_{i=1}^{n}{}_{k=1tox} \alpha_{ik}, pk)$
    $= g^{\sum_{i=1}^{n}{}_{k=1\to x} \alpha_{ik}} r_i^N$
    return $C_i$
End for
The data access layer will compute
    $E(\rho) = C_1 * C_2 * C_3 \ldots C_i\ where\ i = 1\ to\ n$
    $AVG(E(\rho)) = E(\rho)/x$
$Return\ AVG(E(\rho))touser$
User Decryption
    $x = Decrypt(E(\rho), sk)$
```

and the query (e.g. average over the last 10 min). The query can be over a single attribute or multiple attributes.

Due to the homomorphic properties of the Paillier cryptosystem (assuming 3 servers and α, β, γ being the three parts of the data to illustrate the homomorphic property), we have

$$\begin{aligned} C_1C_2C_3 &= (g^{\sum \alpha_k r_1^N})(g^{\sum \beta_k r_2^N})(g^{\sum \gamma_k r_3^N})(\bmod N^2) \\ &= g^{\sum(\alpha_k+\beta_k+\gamma_k)}(r_1r_2r_3)^N(\bmod N^2) \\ &= E\left(\sum d_k, |pk|\right). \end{aligned}$$

Therefore, the result of this is,

$$x = Decrypt(C_1C_2C_3, sk)/n = \sum_{k=1}^{x} d_x/n.$$

3.3. Security analysis

In the proposed privacy-preservation technique, there are four parts of communication namely, (1) communication between the IoT device and the gateway, (2) the communication between the gateway and the data store, (3) the communication between the data store and the data access layer and (4) the communication between the user and the data access layer. We assume that the communication between the IoT device, data store and the user are through secure channels (e.g. DES or AES) that can use any of the most recent light-weight key sharing schemes [24,23].

The security analysis of the proposed privacy-preserving technique is into two stages namely (1) the data ingestion stage and (2) the data access stage. In the data ingestion stage, the data split across n servers are generated by a SHA-3 with a secret key known only to the IoT device or the gateway as shown in Fig. 1. Any inside attack on the data store cannot guess the other random number to infer the actual data. The complexity to decode the actual data increases with increasing number of servers.

In the data access scheme, all the data and the intermediate data are encrypted by the user's public key. Neither the data access layer, the user nor the individual servers have access to the actual data stored in the distributed Privacy-Preserving IoT system. A more detailed analysis of the proposed approach is presented in [5]. The common attack models namely eavesdropping, impersonation, modification and data breach are addressed by the use of the proposed privacy-preserving data ingestion and analysis schemes.

4. Privacy preserving IoT architecture

To develop a Privacy Preserving IoT Architecture we will extend the blueprint IoT Architecture we designed for OpenIoT [41] to incorporate the proposed privacy preservation techniques for end-to-end IoT privacy. In the next section, we first provide an overview of the OpenIoT platform [41]. We then present the Privacy Preserving IoT Architecture IoT as an extension to the OpenIoT platform.

4.1. Overview of OpenIoT

In this paper we consider OpenIoT [41] as a representative of IoT platforms currently being available via the open source community. OpenIoT is a first-of-kind, award-winning, open source IoT platform that provide services for discovery and integration of IoT devices, IoT data integration, and cloud-based storage. OpenIoT also allows IoT application to request and process IoT data as needed to provide IoT services and related products. More specifically, OpenIoT's Architecture is comprised by the following components as depicted in Fig. 3:

- Sensor Middleware: collects filters and combines data streams from virtual sensors or physical devices. It acts as a hub between the OpenIoT platform and the physical world. The sensor middleware uses an extension of the Global Sensor Networks [42] namely x-GSN.
- Cloud Data Storage: is based on the Linked Sensor Middleware Light (LSM-Light) and enables the storage of data streams stemming from the Sensor Middleware thereby acting as a cloud database. The cloud infrastructure stores also the metadata required for the operation of the OpenIoT platforms (functional data).
- Scheduler processes: all the requests for on-demand deployment of services and ensures their proper access to the resources (e.g. data streams) that they require. This component undertakes the following tasks: it incorporates semantic discovery of sensors and the associated data streams that can contribute to service setup; it manages a service and selects/enables the resources involved in service provision.
- Service Delivery & Utility Manager (SDUM): performs a dual role. On one hand, it combines the data streams as indicated by service work flows in order to deliver the requested service. On the other hand, this component performs service metering to keep track of individual service usage.
- Request definition and Request Presentation: components enable on-the-fly specification and visualisation of service requests to the OpenIoT platform. The component selects mashups from an appropriate library in order to facilitate service definition and presentation.

4.2. System architecture

In our architecture, the sensor middleware component namely x-GSN is responsible for communicating with the IoT device. The data ingestion scheme is implemented as a wrapper in x-GSN component that is responsible to send the sensed IoT data to multiple (n) servers. The proposed privacy preserving IoT architecture is presented in Fig. 3.

Remark. For illustration, we make the following assuming (1) we have 3 lsm servers ($n = 3$) (2) x-GSN splits the data using the data ingestion scheme into three parts namely α, β, and γ and (3) the communication channels between the IoT device, x-GSN gateway, user and the data access layer and other internal components are through a secure challenges (all API access to the data is via HTTPS).

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001

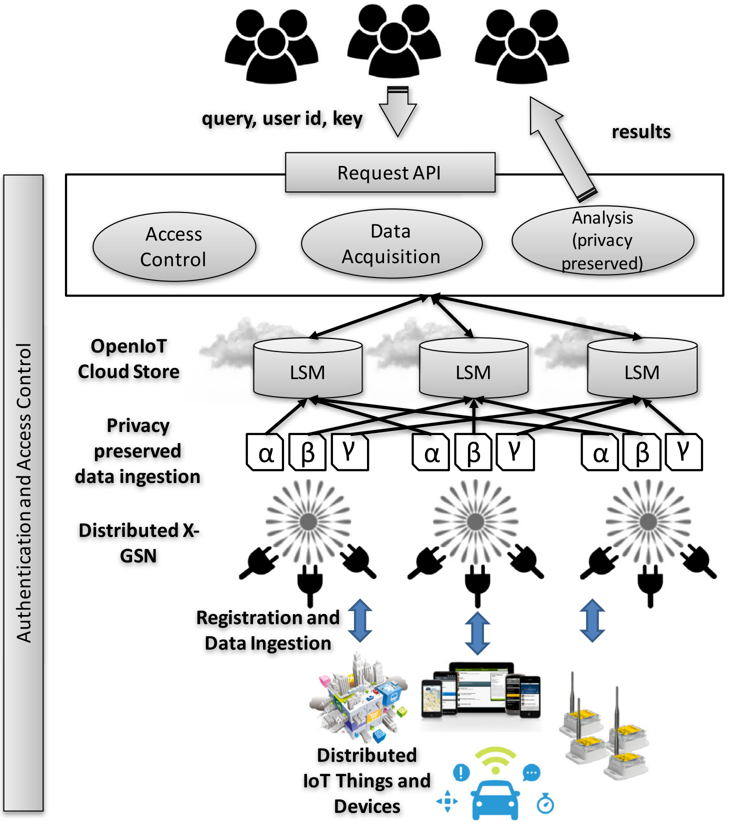
query, user id, key
results
Request API
Access Control
Data Acquisition
Analysis (privacy preserved)
OpenIoT Cloud Store
LSM
LSM
LSM
Privacy preserved data ingestion
α β γ α β γ α β γ
Distributed X-GSN
Registration and Data Ingestion
Distributed IoT Things and Devices
Authentication and Access Control

Fig. 3. Privacy preserving IoT system architecture (depicted using 3 servers).

4.3. Authentication and access control

OpenIoT implements a central authentication and access control service namely CAS. CAS is based on the principles of the OAuth authorisation framework [43]. The OAuth protocol describes methods for providing authorisation in a distributed environment, where distributed client applications get access to owner's resources using time-stamped tokens to avoid transmitting credentials. In the proposed architecture the CAS plays the role of the authorisation server, and all access to protected resources is governed through its OAuth-based authorisation mechanism. Resources could include IoT devices, IoT datasets, API services and web applications. The access control mechanism in the proposed architecture is based on RBAC (Role Based Access Control). A user in our system could be a person, an application or another service accessing services/data of the Privacy-Preserving IoT system (e.g. API's, data store etc.). The platform provides the flexibility to define custom roles enabling access restriction to certain kind of users/services. For example consider the users U = alice, bob and permission $P = ACCESS_{l}sm\&WRITE_{l}sm$. We can define roles such as $R = READ_{l}sm\&READWRITE_{l}sm$ with the permission assignments

$(ACCESS_{l}sm, READ_{l}sm)$,

$(WRITE_{l}sm, READ_{l}sm)$

$(READWRITE_{l}sm, WRITE_{l}sm)$

and define the user assignments as

$(alice, READWRITE_{l}sm), (bob, READ_{l}sm)$.

Then, *alice* will have the permission to both *READ* and *WRITE* to *LSM* (cloud data store) while *bob* will have only *READ* access. As depicted in Fig. 3, the authentication and access control is integrated right from the IoT device layer to the application layer (user/APIs).

4.4. Privacy preserving data storage and access

In this section, we describe the integration of the privacy preservation technique presented in Section 3. As mentioned earlier, x-GSN is responsible to implement the proposed data ingestion scheme. Prior to data ingestion, x-GSN will register the IoT device with the multiple LSM servers using the same sensor ID generated by x-GSN. Instead of maintaining a centralised store to keep track of sensor id, we introduce duplication of sensor id in order to avoid a central point of attack at the registration layer. The assumption is that x-GSN is generally not exposed to the internet and employs schemes to enable protection from inside attacks. In case of the LSM servers, x-GSN will split the data using the random number generation algorithm described in Section 3 into α, $\beta \& \gamma$. The sensor id, along with the split data points is ingested into LSM (e.g. http://lsm.deri.ei/sensor/23456123,10). In order to facilitate the computation of standard deviation and variance, we also store the randomly split squared value of the data.

The privacy-preserving data access is performed by the combination of Service Delivery and Utility Manager (SDUM) and linked sensor middleware (LSM). We have built a component that work with SDUM to integrate the proposed privacy-preserving data access control scheme presented in Section 3.2. The key parts of this component include:

- Request API: The request API is responsible for obtaining the user's request that includes the query (e.g. average temperature in the living room over the last 30 min), the user identification and the user's Paillier public key.
- Access Control: This access control is an interface to the OpenIoT access control mechanism. It is responsible for checking the access permissions of the user over the requested resources such as access to LSM, access to the IoT device producing the data (e.g. some devices could be used only

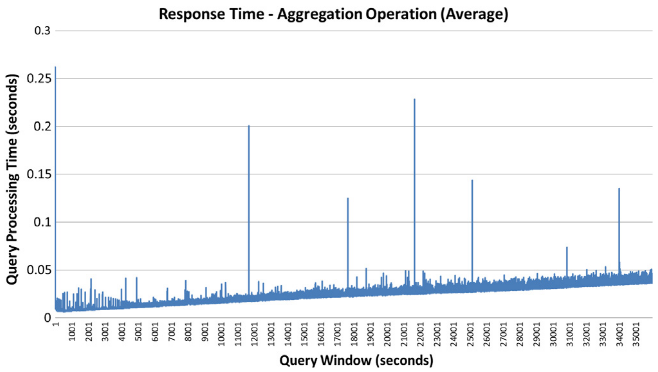

Fig. 4. Query processing (Average) performance with fixed key size and changing window size.

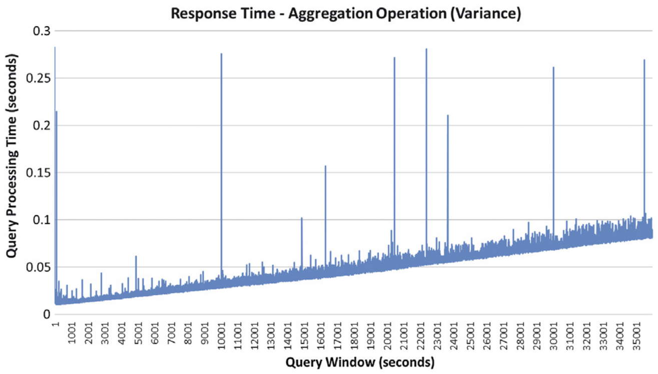

Fig. 5. Query processing (Variance) performance when encrypting individual data points.

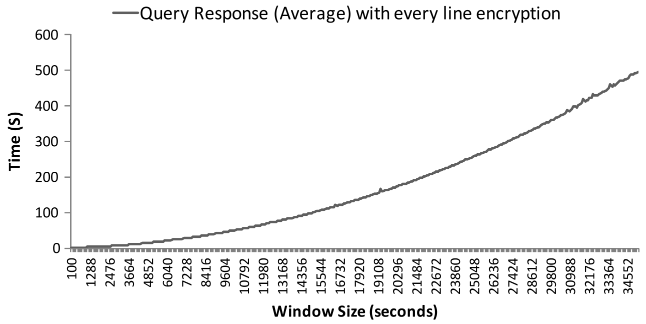

Fig. 6. Query processing (Average)—encrypting every line of data.

for private purposes) and other permissions to access various OpenIoT components (scheduler etc.).

- Data Acquisition: The data acquisition layer will either run Algorithm 1 or 2 (described in Section 3.2) depending on the type of query. The data acquisition is responsible to coordinate the key exchange between the servers, pass the queries to distributed LSM instances for execution and finally collect all the response (encrypted statistical data).
- Analytics (privacy-preserving): This component work with the data acquisition layer to compute the required statistics before returning the final result to the user. At this step, all the data is encrypted using the user's public key. Hence the SDUM never has access to the actual data.

The multiple LSM instances work independent of each other and only know parts of the sensor data. As long as no two LSM servers (in a three server configuration) are compromised, the privacy of the IoT data is preserved. As mentioned earlier, the higher the number of server, the higher the complexity to infer the actual data.

4.5. *Security analysis*

In our proposed method, each data is split into n ($n \geq 3$) parts and stored in different LSM servers in plain-text. Each authorised user owns a Paillier key pair, and public key and can use to retrieve the data (provided he/she is authorised). The extended SDUM and LSM provide a powerful computation layer between front-end users and back-end servers, which handles all computation and data transmission. When a user request a data, he/she can simply sends his request and his public key to SDUM, and then the layer broadcasts the public key to all LSM servers. Next, each LSM server extracts specific data according to the request, and returns encrypted data back to SDUM. The layer does the required statistical computation on the cipher-texts by taking advantage of the homomorphic properties discussed in Section 2. Finally, the user (requester) receives a cipher-text result, which can only be decrypted using his private key. As we use multiple servers to store the split data, we do not store the data in an encrypted form. Further all the servers are completely isolated from each other, and eavesdroppers/hackers cannot reveal any actual data unless all servers are compromised. Moreover, our approach allows the Privacy-Preserving IoT system to be deployed on multiple cloud platforms (e.g. a combination of Amazon and Windows Azure running Linux and Windows).

Furthermore, since we split the data into n parts, the need to encrypt individual data point is avoided which significantly increases data processing speed. We assume the communication channel between the IoT device, x-GSN and LSM are protected using lightweight encryption approaches such as DES. This protects the Privacy-Preserving IoT system from data breach. OpenIoT uses an OAuth-based system for authentication and authorisation. Hence, impersonation and modification of data is restricted as users need to have the right set of permissions and roles to access the Privacy-Preserving IoT system's services. Further CAS necessitates the need for a valid access token (with the correct validity) used to authenticate legitimate users. By incorporating access control across every component including SDUM, LSM, scheduler and x-GSN, the proposed architecture further reduce the risks of impersonation and data breach by allowing IoT application users to have fine-grained access control (e.g. control at the individual component level).

5. Experimental evaluations

In Section 3, we presented the proposed **privacy-preserving technique for IoT data**s and in Section 4, we presented the architecture of the Privacy-Preserving IoT system that incorporated. In this section, we present the result of our experiments evaluating the performance of the proposed Privacy-Preserving IoT system implemented by extending OpenIoT. In particular, we conducted the following experiments in order to determine the overheads introduced to the performance of a typical IoT system while incorporating end-to-end privacy.

1. The performance of query processing using the Privacy-Preserving IoT system incorporating the proposed scheme for different aggregation operations (e.g. average and variance)
2. The impact on query processing performance when using different Paillier key sizes to achieve privacy-preserving data access.

5.1. *Experimental setup*

For experimental evaluation, we deployed 3 ($n = 3$) independent containers of LSM server on amazon instances. We used

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001

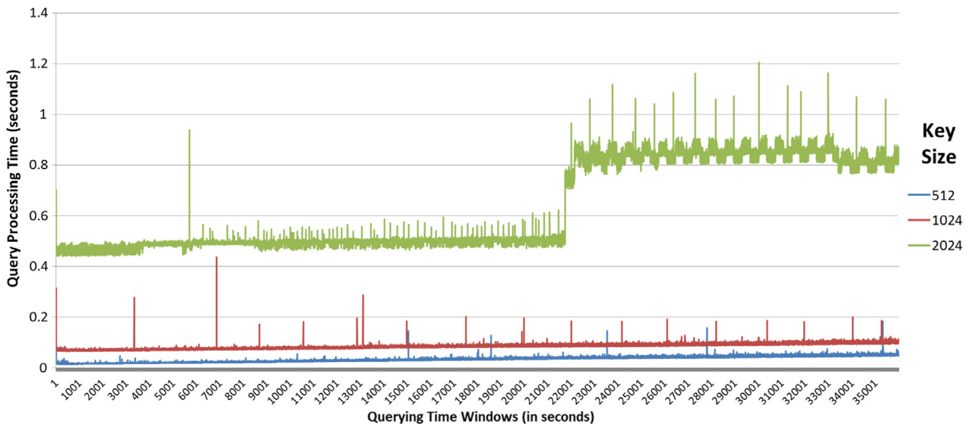

Fig. 7. Query processing (Average) performance with varying key sizes and window sizes.

t2.large instances on Amazon Elastic Cloud Computing (EC2) platform [44], running Ubuntu 14.04.4 LTS operating system. The hardware configuration of the servers were two vCPU 3.3 GHz Intel Xeon processors and 8 GiB of memory. The x-GSN instance was connected to a synthetic IoT data source that produced temperature and humidity data. The dataset was collected over a period of 9 h allowing us to run window queries ranging from 2 min to 9 h. For a given query, the following steps are performed (1) the query is passed to the SDUM data access layer, (2) SDUM presents the query with the user's public Paillier key to the 3 LSM servers, (3) LSM server execute the query, encrypt the data using the public key and returns the encrypted data and (4) SDUM computes the statistical analysis as described in the query (Average and Variance) using Paillier's homomorphic properties and (5) SDUM returns the encrypted statistical results to the user. The query computation time is computed as the sum of times required to perform steps 1–5. A *query window* in our experiment is the amount of data requested by the end-user/application. For example a typical query would be *average temperature for the past 30 min*.

5.2. *Query processing performance*

In this experiment, we test the performance of the proposed privacy-preserving data access scheme. The query provided by the user was to compute the average of the sensor data over a given window of time (2 min to 9 h). The result of this experiment is presented in Fig. 4. As the results indicate, with a time window of 9 h, the query response time is less than 1 s. This validates the efficiency of the proposed scheme and it is a applicability for real-world IoT applications.

To provide a benchmark of another common aggregation operation namely variance, we modified our scheme to store both the original sensor data d_i into three parts α, β, γ and the square of the sensor data namely d_i^2 into three parts namely $\widetilde{\alpha}, \widetilde{\beta}, \widetilde{\gamma}$. The result of this experiment is presented in Fig. 5.

In comparison, we conducted a similar experiment (Fig. 6) to compute privacy-preserving average of IoT data using the traditional approach i.e. encrypt every data point using Paillier key. As it can bee seen, the computation time grows exponentially with increasing amount of data. Given the plethora of IoT devices and the huge amount generated, the traditional approach will fail or will result in very poor query processing response time. However, the proposed data ingestion scheme ensures privacy by splitting the data into n parts while also handling large amounts of IoT data. In summary, the proposed approach introduces very less overheads with any significantly impact on the performance of the IoT system while ensuring end-to-end privacy.

5.3. *Impact of paillier key size*

In this experiment, we tested the impact of using various Paillier key sizes on the performance of the IoT system's query processing capability (while changing the query window size). This experiment is relevant as in the proposed scheme, no prior key exchange exists. Hence, the user is open to choose a Paillier key combination of any size. The results of this experiment are presented in Fig. 7. As noted from the experimental outcomes, the change in key size has insignificant impact on the performance of the query processing except when the key size is 2024. However performance when using a 2024 key size is still within 1 s while querying 9 h of sensor data.

The experimental evaluations clearly validate the feasibility and applicability of the novel IoT privacy preserving techniques and architecture for efficient support of IoT applications. Given the ever increasing IoT (an ecosystem of billions of IoT devices) and the amount of data contributed by these IoT devices, the proposed approach is effective and efficient as it ensures privacy-preserving storage and access to IoT data without compromising on the overall performance of the IoT system. In particular, the results presented in Fig. 6 indicate the exponential increase in processing overhead when using the traditional approach of storing individually encrypted IoT data. Given the nature of IoT and the plethora of IoT devices/things, this approach is not feasible to ensure security and privacy of large scale (in terms of IoT devices used and data points managed) IoT application. Moreover, note that our experimental results are based on floating point data which introduces additional processing overheads during the splitting, encryption and decryption process. However, the proposed privacy-preserving techniques, architecture and experimental system handles this very well without imposing any significant impact as it is evident from these experimental results.

6. Conclusion

In this paper, we present a novel techniques for end-to-end privacy and security of next generation IoT systems. The proposed techniques uses an innovative approach wherein each data item x that is collected from an IoT device is randomly expressed as a sum of multiple numbers (data addends), such that $x = x_1 + x_2 + x_n$ $(n \geq 2)$, and stored on n data stores that keep only one of the components number x_i. We also proposed a privacy-preserving data access scheme that uses the homomorphic properties of the Paillier cryptosystem to allow retrieval of analysed IoT data without exposing the actual data to any of the servers or the users. We implemented and demonstrated the feasibility and applicability of novel IoT privacy preserving techniques and architecture using

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001

the widely used OpenIoT platform. Experimental evaluations of the implemented Privacy-Preserving IoT system using data generated from IoT devices show that the proposed techniques has insignificant impact on the overall performance of the IoT system.

References

[1] Gartner, accessed: 2016-03-7. [link]. URL http://www.gartner.com/newsroom/id/3165317.

[2] D. Georgakopoulos, P.P. Jayaraman, Internet of things: from Internet scale sensing to smart services, Computing 98 (10) (2016) 1041–1058.

[3] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Internet of things: Vision, applications and research challenges, Ad Hoc Networks 10 (7) (2012) 1497–1516.

[4] X. Yi, J. Willemson, F. Nait-Abdesselam, Privacy-preserving wireless medical sensor network, in: Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on, IEEE, 2013, pp. 118–125.

[5] X. Yi, A. Bouguettaya, D. Georgakopoulos, A. Song, J. Willemson, Privacy protection for wireless medical sensor data, IEEE Trans. Dependable Secure Comput. 13 (3) (2016) 369–380.

[6] R. Ranjan, M. Wang, C. Perera, P.P. Jayaraman, M. Zhang, P. Strazdins, R. Shyamsundar, City data fusion: Sensor data fusion in the Internet of things, Int. J. Distrib. Syst. Technol. 7 (1) (2016) 15–36.

[7] B. Escribano, Privacy and security in the Internet of things: challenge or opportunity, accessed: 2016-03-7. URL http://www.olswang.com/articles/2014/11/privacy-and-security-in-the-internet-of-things-challenge-or-opportunity.

[8] H.L. Huansheng Ning, Cyber-physical-social based security architecture for future Internet of things, Adv. Internet Things 2 (1) (2012) 1–7.

[9] N.D.W. Cahyani, B. Martini, K.-K.R. Choo, A.M.N. Al-Azhar, Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study, Concurr. Comput. Practice Exper. (2016) http://dx.doi.org/10.1002/cpe.3855.

[10] C.J. DOrazio, K.K.R. Choo, L.T. Yang, Data exfiltration from Internet of things devices: ios devices as case studies, IEEE Internet Things J. PP (99) (2016) 1–1. http://dx.doi.org/10.1109/JIOT.2016.2569094.

[11] Q. Do, B. Martini, K.K.R. Choo, A cloud-focused mobile forensics methodology, IEEE Cloud Comput. 2 (4) (2015) 60–65.

[12] S. Kalra, S.K. Sood, Secure authentication scheme for iot and cloud servers, Pervasive Mob. Comput. 24 (2015) 210–223.

[13] C. Liu, X. Zhang, C. Liu, Y. Yang, R. Ranjan, D. Georgakopoulos, J. Chen, An iterative hierarchical key exchange scheme for secure scheduling of big data applications in cloud computing, in: 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013, pp. 9–16.

[14] B.C. Neuman, T. Ts' O, Kerberos: An authentication service for computer networks, IEEE Commun. Mag. 32 (9) (1994) 33–38.

[15] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[16] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM, 2002, pp. 41–47.

[17] K.K. Venkatasubramanian, A. Banerjee, S.K. Gupta, et al., Ekg-based key agreement in body sensor networks, in: INFOCOM Workshops 2008, IEEE, IEEE, 2008, pp. 1–6.

[18] R. Roman, C. Alcaraz, J. Lopez, N. Sklavos, Key management systems for sensor networks in the context of the Internet of things, Comput. Electr. Eng. 37 (2) (2011) 147–159.

[19] W. Du, J. Deng, Y.S. Han, P.K. Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, 2003, pp. 42–51.

[20] S.A. Camtepe, B. Yener, Combinatorial design of key distribution mechanisms for wireless sensor networks, in: Computer Security, ESORICS 04, Springer, 2004, pp. 293–308.

[21] D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS'03, ACM, New York, NY, USA, 2003, pp. 52–61.

[22] M. Vuini, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, R. Guizzetti, Oscar: Object security architecture for the Internet of things, Ad Hoc Networks 32 (2015) 3–16.

[23] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, M. Rossi, Secure communication for smart iot objects: Protocol stacks, use cases and practical examples, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, 2012, pp. 1–7.

[24] H. Jiang, F. Shen, S. Chen, K.-C. Li, Y.-S. Jeong, A secure and scalable storage system for aggregate data in iot, Future Gener. Comput. Syst. 49 (2015) 133–141.

[25] C. Perera, P.P. Jayaraman, A. Zaslavsky, D. Georgakopoulos, P. Christen, Sensor discovery and configuration framework for the Internet of things paradigm, in: 2014 IEEE World Forum on Internet of Things, WF-IoT, 2014, pp. 94–99.

[26] A. Sahai, B. Waters, Fuzzy identity-based encryption, in: Advances in Cryptology, EUROCRYPT 2005, Springer, 2005, pp. 457–473.

[27] S. Yu, K. Ren, W. Lou, Fdac: Toward fine-grained distributed data access control in wireless sensor networks, IEEE Trans. Parallel Distrib. Syst. 22 (4) (2011) 673–686.

[28] P. Picazo-Sanchez, J.E. Tapiador, P. Peris-Lopez, G. Suarez-Tangil, Secure publish-subscribe protocols for heterogeneous medical wireless body area networks, Sensors 14 (12) (2014) 22619–22642.

[29] D. Boneh, M. Franklin, Identity-based encryption from the weil pairing, in: Advances in Cryptology, CRYPTO 2001, Springer, 2001, pp. 213–229.

[30] C. Hu, J. Zhang, Q. Wen, An identity-based personal location system with protected privacy in iot, in: 2011 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT), IEEE, 2011, pp. 192–195.

[31] J. Liu, Y. Xiao, C.P. Chen, Internet of things' authentication and access control, Int. J. Secur. Netw. 7 (4) (2012) 228–241.

[32] D. Malan, T. Fulford-Jones, M. Welsh, S. Moulton, Codeblue: An ad hoc sensor network infrastructure for emergency medical care, in: International Workshop on Wearable and Implantable Body Sensor Networks, 2004.

[33] A. Wood, G. Virone, T. Doan, Q. Cao, L. Selavo, Y. Wu, L. Fang, Z. He, S. Lin, J. Stankovic, Alarm-net: Wireless sensor networks for assisted-living and residential monitoring, University of Virginia Computer Science Department Technical Report 2.

[34] J. Ko, J.H. Lim, Y. Chen, R. Musvaloiu-E, A. Terzis, G.M. Masson, T. Gao, W. Destler, L. Selavo, R.P. Dutton, Medisn: Medical emergency detection in sensor networks, ACM Trans. Embedded Comput. Syst. (TECS) 10 (1) (2010) 11.

[35] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, Spins: Security protocols for sensor networks, Wireless Netw. 8 (5) (2002) 521–534.

[36] A. Samani, H.H. Ghenniwa, A. Wahaishi, Privacy in Internet of things: A model and protection framework, Procedia Comput. Sci. 52 (2015) 606–613.

[37] A. Ukil, S. Bandyopadhyay, J. Joseph, V. Banahatti, S. Lodha, Negotiation-based privacy preservation scheme in Internet of things platform, in: Proceedings of the First International Conference on Security of Internet of Things, SecurIT'12, ACM, New York, NY, USA, 2012, pp. 75–84.

[38] C. Doukas, I. Maglogiannis, V. Koufi, F. Malamateniou, G. Vassilacopoulos, Enabling data protection through pki encryption in iot m-health devices, in: 2012 IEEE 12th International Conference on Bioinformatics Bioengineering, BIBE, 2012, pp. 25–29.

[39] N.H.A. Rahman, W.B. Glisson, Y. Yang, K.K.R. Choo, Forensic-by-design framework for cyber-physical cloud systems, IEEE Cloud Comput. 3 (1) (2016) 50–59.

[40] P. Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, Springer Berlin, Heidelberg, Berlin, Heidelberg, 1999, pp. 223–238.

[41] M. Serrano, H.N.M. Quoc, D. Le Phuoc, M. Hauswirth, J. Soldatos, N. Kefalakis, P.P. Jayaraman, A. Zaslavsky, Defining the stack for service delivery models and interoperability in the Internet of things: A practical case with openiot-vdk, IEEE J. Sel. Areas Commun. 33 (4) (2015) 676–689.

[42] J.-P. Calbimonte, S. Sarni, J. Eberle, K. Aberer, Xgsn: An open-source semantic sensing middleware for the web of things, in: 7th International Workshop on Semantic Sensor Networks, Riva del Garda, Trentino, Italy, 2014.

[43] D. Hardt, The oauth 2.0 authorization framework, accessed: 2016-03-11. URL https://tools.ietf.org/html/rfc6749.

[44] Amazon, Amazon elastic compute cloud (amazon ec2)Accessed: 2016-03-7. URL https://aws.amazon.com/ec2.

Prem Prakash Jayaraman is currently a Research Fellow at Swinburne University of Technology, Melbourne. His research areas of interest include, Internet of Things, cloud computing, mobile computing, sensor network middleware and semantic internet of things. He has authored/co-authored more than 50 research papers in international Journals and conferences such as IEEE Trans. on Cloud Computing, IEEE Selected areas in Communication, Journal of Computational Science, IEEE Transactions on Emerging Topics in Computing, Future Generation Computing Systems, Springer Computing, ACM Ubiquity Magazine, IEEE Magazine. He is one of the key contributors of the Open Source Internet of Things project OpenIoT that has won the prestigious Black Duck Rookie of the Year Award in 2013. He has been the recipient of several awards including hackathon challenges at the 4th International Conference on IoT (2014) at MIT media lab, Cambridge, MA and IoT Week 2014 in London and best paper award at HICSS 2016/2017 and IEA/AIE-2010. Previously he was a Postdoctoral Research Fellow at CSIRO Digital Productivity Flagship, Australia from 2012 to 2015.

Xuechao Yang is a Ph.D. candidate on cyber security at RMIT University. In 2013, he completed bachelor of information technology at the RMIT, and also he completed bachelor of computer science with honours in 2014. His research interests include cryptosystems, homomorphism and blockchain technology.

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001

Ali Yavari is currently pursuing his Ph.D. in School of Computer Science and Information Technology at Royal Melbourne Institute of Technology (RMIT), Australia. Prior to joining RMIT, Ali was Research Engineer at KTH University (Sweden) in Communication Systems department. Moreover, he has been working for more than ten years in IT industry as chief executive officer, project manager, system designer, and developer for aviation, bank, and IT companies in several countries. He serves as program committee member and designated reviewer in several international conferences and workshops. Ali is currently a committee member in Computer Society Chapter at IEEE Victorian Section. You can find more information about Ali at www.aliyavari.com.

Dimitrios Georgakopoulos is a Prof. in Computer Science and Director of the Key Lab for IoT at Swinburne University of Technology, Melbourne, Australia. Before that was Research Director at CSIRO's ICT Centre and Executive Director of the Information Engineering Laboratory, which was the largest Computer Science program in Australia. Before CSIRO, he held research and management positions in several industrial laboratories in the US, including Telcordia Technologies (where he helped found two of Telcordia's Research Centers in Austin, Texas, and Poznan, Poland); Microelectronics and Computer Corporation (MCC) in Austin, Texas; GTE (currently Verizon) Laboratories in Boston, Massachusetts; and Bell Communications Research in Piscataway, New Jersey. He was also a full Professor at RMIT University, and he is currently an Adjunct Prof. at the Australian National University and a CSIRO Adjunct Fellow. Prof. Georgakopoulos has produced 170+ journal and conference publications in the areas of IoT, process management, and data management, and has 10,500+ lifetime citations.

Xun Yi is currently a Professor with the School of Science, RMIT University, Australia. His research interests include applied cryptography, computer and network security, mobile and wireless communication security, and privacy-preserving data mining. He has published more than 150 research papers in international journals, such as IEEE Trans. Computers, IEEE Trans. Parallel and Distributed Systems, IEEE Trans. Knowledge and Data Engineering, IEEE Trans. Wireless Communication, IEEE Trans. Dependable and Secure Computing, IEEE Trans. Information Forensics and Security, IEEE Trans. Circuit and Systems, IEEE Trans. Vehicular Technologies, IEEE Communication Letters, IEE Electronic Letters, and conference proceedings. He has ever undertaken program committee members for more than 30 international conferences. Recently, he has led a few of Australia Research Council (ARC) Discovery Projects. Since 2014, he has been an Associate Editor for IEEE Trans. Dependable and Secure Computing.

Please cite this article in press as: P.P. Jayaraman, et al., Privacy preserving Internet of Things: From privacy techniques to a blueprint architecture and efficient implementation, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.001