Lightweight Privacy-Preserving Scheme using Homomorphic Encryption in Industrial Internet of Things

Shancang Li, Shanshan Zhao, Geyong Min, Lianyong Qi, and Gang Liu

***Abstract*—The emerging technologies, such as *smart sensors, 5G/6G wireless communication, artificial intelligence, etc.*, have being maturing the future Internet of Things (IoT) by connecting massive number of devices, which are expected to consistently collect and transmit real-time data to support business intelligence in an efficient and privacy-preserving way. The IoT can afford businesses predictive maintenance, improve field service, asset tracking, and further enhance customer satisfaction and facility management in industrial sectors. However, the privacy concern in IoT is a big challenge in IoT applications and services. This work proposed a lightweight privacy-preserving scheme based on homomorphic encryption in the context of the IoT, in which we investigated and analysed the privacy issues between the data owners, untrustworthy third-part cloud servers, and the data users. Meanwhile, computationally-efficient homomorphic algorithms are proposed to guarantee the privacy protection for the data users. Experimental results demonstrates that the proposed scheme can effectively prevent privacy breaches in IoT.**

***Index Terms*—Security, Provenance, IIoT, Lightweight Privacy, IoT security**

I. Introduction

THE Internet of Things (IoT) technology is being broadly used and can have huge impact in our daily lives [1]. In industry setting, the Industrial IoT (IIoT), or Industry 4.0, is expected to improve the user experiences and create new business streams, taking advantages of the new capability of IIoT device and secure data analytics [2], [3]. The IIoT connects smart machines and sensors to form automatic systems that collect, exchange, and analyse real-time data, and deliver valuable insights to improve the performance, safety, reliability, and energy consumption of industry sectors [4], [5], [6], [7].

The IIoT are promising in many industrial areas, including healthcare, smart manufacturing, smart city, smart grid, *etc.* As a vast and more complex system, the IIoT is expected to offer huge potential benefits to existing industry systems by automating projects, optimizing digital transformation goals, improving productivity, reducing costs, *etc.* Meanwhile, the IIoT is expected to make far-reaching impact on the operation of industries around world. The IIoT can provide business predictive maintenance, which uses real-time data to accurately predict defects in smart devices, and enabling industrial sectors to take action to address those issues before apart fails or a device goes down. The IIoT can improve field service by identifying potential issues in industry systems before they become major problems. It can also provide asset tracking to monitor the real-time location, conditions, and storages, which is important in healthcare, supply chain, manufacturers, logistics, *etc.* [5], [8].

The IIoT connects industry devices to the Internet and can bring many benefits, however, it also leaves devices in IIoT vulnerable to hacking and solutions to prevent IIoT devices from being exploited are necessary. In many IIoT applications, such as healthcare, life-threatening cyberattacks are targeting medical devices (such as insulin pumps attack, baby monitor) and can cause serious security issues [9], [10]. Actually, the IIoT systems are facing many challenges and becoming the targets of cyberattacks, as summarised below,

1) Security in IIoT: In the past few years, a number of security solutions have been proposed for IIoT, including cryptographic techniques, data encryption algorithms, secure communication channels, strong anonymous authentication, and access controls in IIoT. A number of promising technologies, such as attributed-based access control, or even blockchain technologies have been developed in ensuring IIoT security.

2) Privacy in IIoT: The information leakage is a big concern in IIoT in many critical IIoT systems, such as smart grid, industrial critical systems (ICS)/supervisory control and data acquisition (SCADA) systems, *etc.*, in which the need for privacy assurance of data collected, as well as the privacy issues associated with critical infrastructures and IoT services, has been raised in the literature. The data aggregation in key IIoT applications is the core of privacy preserving [11], [12].

3) Safety: Many IIoT devices suffer from intermittent defects that can cause device on fault. In safety of IIoT, the following issues need to be clearly addressed: device malfunction, devices communications, device incompatibilities and errors, unintended accident, malicious intents, *etc.*

This paper aims at developing secure solution by balancing the security and efficiency in IIoT. A lightweight privacy-preserving scheme will be detailed for IIoT system. An air quality monitoring use case will be given to address the challenges mentioned above. The main contributions of this paper include:

- This work proposes a secure data processing framework for applications in IIoT that well address the privacy is-

Shancang Li and Shanshan Zhao are with University of the West of England, Bristol BS16 1QY, UK. Email: shancang.li@ieee.org.

Geyong Min is with the University of Exeter, Exeter EX4 4PY, UK.

Lianyong Qi is with Qufu Normal University, Qufu 56650, China.

Gang Liu is with Xidian University, Xi'an 710071, China.

sues between data owners, untrusted cloud server, and the data users, and a data labeling scheme in IoT environment for resource-constrained devices, which allows data users to learn a label ℓ_i at each node i while keeping low communication complexity. A label is typically a unique index of the data created by a node and could be designed as small size integer.

- A lightweight privacy-preserving scheme for both data owner and data user in IIoT is proposed that by dividing computational costs into a fixed and a dynamic part, and in data processing only dynamic part is updated.
- We improved upon the labHE of [13] by leveraging a pre-processing phase, where the computational cost is reduced. The data users (applications) no longer need to perform an expensive evaluation on the resulting ciphertexts. This allows apps on IoT devices to utilise more efficient HE and further improve the performance.

The remainder of this work is organized as follows: Section II elaborates the recent works in the field of lightweight privacy-preserving in IIoT; Section III and Section IV present the proposed lightweight privacy-preserving scheme for IIoT systems in detail. An use case is introduced in Section V. Finally, Section VI concludes this work.

II. Related Works

The privacy preserving at endpoint devices in IIoT has been discussed as a core security issues in the past few years [1], [5]. The privacy issues in the IoT have been well studied and a number of privacy preserving protocols have been developed for IoT devices and applications [4], [5]. With the advent of new techniques in industry, the privacy concerns in IIoT (or Industry 4.0) are on the rise as well [1]. The IIoT is able to generate, collect, store, and process huge amounts of data, it has become a main target of cyberattacks that can cause large-scale system failures and massive destruction [7]. The security and privacy challenges in IIoT systems are very complex than the existing industry automation systems [14], in which data generated by IIoT device must be protected against cyber attacks. Efficient ways are needed to identify, secure remote remediaiton [15].

In general, an IIoT system consists of a large number of IIoT devices, which are vary widely in terms of computational capabilities, security levels, connectivities, *etc*. In IIoT, device/service/user authentication is one of key security issues and that is significantly varied as the applications [16]. For example, in e-commercial system, digital certificates are widely used to gurantee the trustworthness of an IIoT device, which can provide secure access control for IoT devices. In recent, the emerging decentralised ledger technologies, (such as blockchain), have been used in IIoT to enhance the PKI and ensure digital cert management [1] by providing good auditability. In IIoT, two-factor authentication is sufficient. Since the devices are very different and the authentication schemes must consider the variation of protocols and standards. In the past few years, many research efforts have been conducted on authentication solution for applications in IIoT, such as healthcare, social networks, personalised privacy, cloud privacy, *etc*.

The IIoT must be able to provide industrial facilities and systems, including ICS, cyber physical systems (CPS), industrial automation and control system (IACS), *etc*., with ability to defend against new cyberthreats that take advantage of weaknesses and other attack vectors that come with the adoption of new technology [16], [17], [18]. To facilitate modern security architectures into existing industry systems, following fundamental questions must be addressed: 1) to address the security architecture for existing systems; 2) to extend the modeled architecture artefacts to include security. To make it happen, the new systems must consider the business process, technologies, system architectures, and integration with existing cyber systems. The number of IIoT devices is increasingly growing, which presents new threats, vulnerabilities, and attack surface is significantly expanded. As shown in Figure. 1, the increased connectivity in IIoT also increases the attack surface for the software, access control, and critical processes.

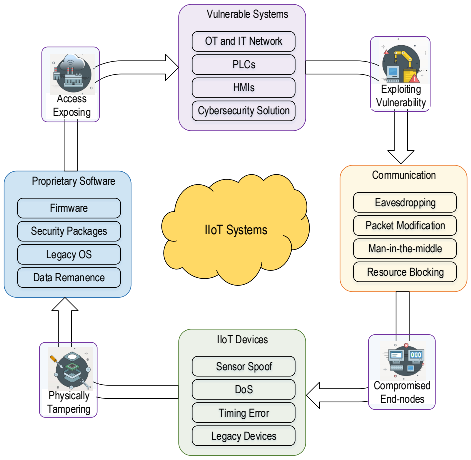

Figure 1. Attack surfaces of IIoT

Personal privacy have been well defined in information systems [19], [7], however, in industrial environment (IIoT or Industry 4.0), privacy is still an open question. As discussed above, the smartization of devices, products, and operation technologies make the industry systems are exposing maximum information the world has ever seen. Specifically, in the past few years, the information leaking incidents, such as `Miral, DDoS, Stuxnet`, have make it very concern about the data privacy and personal data privacy in industry environment. However, the IIoT faces difficulties from following aspects: (1) It is unclear for data privacy in IIoT applications; (2) No privacy standization for IIoT applications; (3) Consent gathering from the user is inefficient; and (4) Industry facilities profiling is very difficult to monitor. In this works, we summarise that the privacy issues in IIoT and its application areas (such as real-time facial recognition systems, healthcare systems) covers:

- The raw data storage;
- Insecure devices, applications, and users;
- Side channel attack for IoT devices;
- Data encryption over resource-constrained devices;

- Insufficient security standardization;
- Diversity of IoT devices;
- Insecure data collection and sharing protocols;

In recent, a number of research works have been done on lightweight privacy-preserving, including lightweight credentials (domain credentials, generic credentials) management, authentication, multi-user authentication, and privacy-preserving solutions. Figure 2 shows the data flow between data owners and data users, in which the cloud server(s) always provided by untrusted third-party.

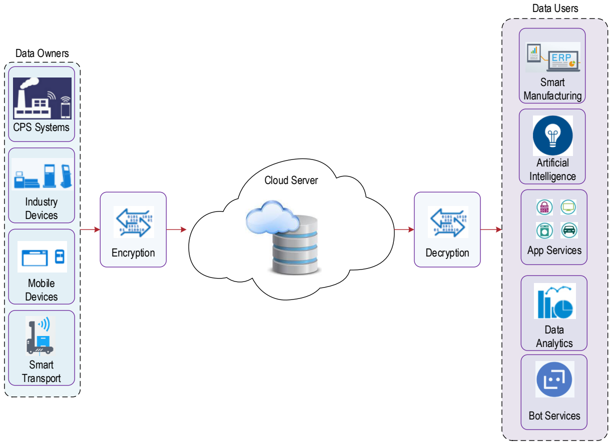

Figure 2. Data Encryption/Decryption in IIoT

In the past few years, many research efforts have been done on homomorphic encryption [20], [14], [21]. Most of these works focused on the computational data outsourcing to insecure cloud services [20]. The increasingly use of IoT technologies in industry makes it is important to leverage the security, privacy, and computing power. The computational expensive traditional data encryption way cannot match the needs of security and privacy in Industry 4.0 environment. The homomorphic encryption (HE) allows computation over encrypted data, including multiplication (mul), addition (add), constant-multiplication (cmul), when decrypted, matches the result of the operations that performed over plaintext. This features can hide the queries from the cloud servers and can further improve the privacy protection between cloud servers and data users. In [2], Lu developed an communication-efficient privacy preserving solution for IoT applications using BGN homomorphic encryption techniques. Meanwhile researchers investigated the privacy preserving for IIoT devices from the access linkability for roaming service aiming at providing multilevel privacy preservation [22].

To solve this problem, a number of new lightweight homomorphic schemes for IoT were developed. Lu *et al.* developed a communication-efficient secure query scheme in fog environment [15], in which both the data user (*e.g.*, application) and data owner (*e.g.*, an IoT device) can be privacy-preserved using HE. In [20], Fiore, *et al.* developed a multi-key homomorphic authenticator from the view point of data outsourcing [20]. The HE can make ciphertexts the same size after operations, which is suitable for some resource constrained devices. HE contains addition and multiplication operations. Gentry proposed the full homomorphic encryption (FHE) by looking for an encryption scheme with low decryption complexity. However, the FHE still has some limitations: (1) The boostrapping is computational expensive and cannot be performed by resource-constrained IIoT devices; (2) The size of ciphertext is too big that needs lots of storage space; (3) The size of public key is too big as well. FHE has been widely used in functional encryption, verifiable computing, secure multi-party computation, and attribute-based encryption (ABE).

In [23], a policy-based management system was developed to manipulate devices in fog computing environment, which considers the user requests of resources, supports, *etc.* Jana *et al.* [6] developed an on-device sensor abstract that can avoid data leakage from accessing the raw data by data users. However, it is unable to deal with multiple data sources of data (multiple IoT devices) in IoT environment.

However, most existing privacy preserving solutions still face challenges, such as unclear data privacy, expensive computational costs, inefficiencies, *etc.* In next Section, we will introduce a new privacy preserving scheme by refining the data privacy in IIoT.

III. Privacy-Preserving IIoT System

It is reported that more than 90% of organizations dependent upon OT (such as those in the manufacturing, healthcare, transportation industries) experiences at least one major cyberattack in the past two years [24], [25]. As discussed above, most IIoT system contains large amount of facilities, locations, machines exchanging data through cloud platforms and various applications. It is really difficult to map the complete attack surface of IIoT, Figure 1 summarises the critical attacks in IIoT systems. It can be seen that the surface areas includes IIoT devices, vulnerable systems, proprietary software, and communication protocols. This work focus on the privacy preserving in an IIoT system, which covers a data owner, a cloud server (or applications), and a data user (application).

A. System Model

In our IIoT system, we consider three roles as shown in Figure 5, which consists of three main entities, namely,

1) Data Owner (DO): It can be a set of IoT devices $D = \{D_1, D_2, \ldots, D_n\}$ that can generate and process data with limited resources in terms of computation, speed, storage space, and memory size;

2) Cloud Server: The untrustworthy cloud servers can be a third-party hosting server that can store data generated by IoT devices; Meanwhile, the cloud server can also offer computing services for both data owners and data users;

3) Data User (DU): It can be the IoT applications that use the data stored over the cloud server; it could be the data owner, *e.g.*, an IoT device D_i could can be both data owner and data user. In this work we use app denote the a data user.

4) Sensitive Data: A large part data in IIoT must be protected against cyberattacks, including personal data, individual data, and operational data that created by IoT devices. Sensitive data needs particular security encryption solutions.

5) Privacy Policies and Enforcement: The privacy policies contains the rules that determine the authorised operations on sensitive data, which featured on *purpose, visibility, granularity*, and *obligations* of data [26]. Meanwhile, the policy enforcement processes are the usages of policies to protect associating data, access request, evaluation of policies, and the control access of data.

6) Privacy Threats in IIoT: Mainly include the identifiability threat, linkability, unauthorised data disclosure, excessive data disclosure, and profiling to infer interests and habits of individuals from their data and metadata.

B. Privacy Model

For an IIoT with n nodes, we have device set as $D = \{D_1, D_2, \ldots, D_n\}$. Each node D_i can generate data $\mathbf{x}_i$. For privacy reason, node D_i will not leak individual data $\mathbf{x}_i$ to others. For the cloud server, it should not know the $\mathbf{x}_i, \forall i \in \{1, n\}$. To solve this problem, a straightforward way is to use encryption, to guarantee the privacy of data user, we will use homomorphic encryption $\mathbf{c}_i = \mathsf{Enc}(\mathbf{x}_i)$ here $\mathbf{c}_i$ will be transmitted to cloud server by node D_i.

Definition 1. *Data Owner Privacy (DOP).* For all IoT devices, each of them does not learn anything from other IoT devices, *e.g.*, D_i generated data $\mathbf{x}_i$ and D_j generated data $\mathbf{x}_j$, D_i can only transmit encrypted data $\mathbf{c}_i = \mathsf{Enc}(\mathbf{x}_i)$ and D_i cannot learn anything from $\mathbf{c}_j$. Device D_i might unable to aware if D_j is active or not. Attacks from an external adversary are beyond the scope of this paper. In this case, we say D_i is privacy-preserving for D_j. DOP means that no information is leaked about device's set elements to a malicious server, except the upper bound on the device's set size.

Definition 2. *Data User Privacy (DUP).* In IIoT environment, a data user $\mathbf{U}$ might be an IoT device, or an application from user. In this work, we assume that application (app) is data user that can conduct query to cloud server $\mathbf{S}$. From the view point, a data user U_k might use data from one or more data owners. Different data users do not have learn anything from each other. For example, U_k does not have any connection with U_τ even they might consume the data from the same data owner D_i. In some scenarios, the data user also includes extra user privacy, such as preferences, location, *etc.*.

Definition 3. *Cloud Server Privacy.* For a cloud server $\mathbf{S}$, it might not be trustworthy for both DOs and DUs. $\mathbf{S}$ does not conduct data encryption or data decryption, so it can only be aware who and when upload or download encrypted data but unable to know the contents of the data. In this work, $\mathbf{S}$ can also offer computing services for the queries from $\mathbf{U}$. The cloud server privacy-preserving aims at prevent the data bleaches such as insider theft, malware and ransomware, DDoS, *etc.* Or misuse, disclose, modify, deny access, *etc.* If the IoT node D_i learns no information (except the upper bound on size) about the subset of elements on the server that are NOT in the intersection of their respective sets.

Definition 4. *Label Privacy*. At a data owner node, each piece of encrypted data is associated with a unique label ℓ (*e.g., timestamp, ID, etc.*). The class labels $\ell_{i,j}|\{j \in [1, M+i]\}$ may associate with the participated devices. In this work, we assume that the data users are happy to share the labeled data and do not consider the label privacy.

Definition 5. *Differential Privacy (DP).* The DP aimed at protecting the privacy against learning by statistical queries on a database. For a randomized algorithm $Alg : \mathbf{x} \to \mathbf{o}$ gives $\epsilon - DP$ if for all adjacent datasets $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{X}$ differing on at most one element. If the algorithm Alg satisfies Eq.(1), the algorithm Alg will satisfy $\epsilon - DP$ protection.

$$P_r\left[Alg\left(\mathbf{x}\right) \in \mathbf{o}\right] \le exp\left(\varepsilon\right) \times P_r\left[Alg\left(\mathbf{x}'\right) \in \mathbf{o}'\right] \tag{1}$$

in which $P_r[\cdot]$ denotes the randomness of Alg on the data $\mathbf{x}$ and $\mathbf{x}'$.

In this work, the third part cloud servers can only provide on-demand self-service. In [27], [28], Freedman *et al.* proposed a single-server private information retrieval (PIR) protocol using keywords based on additive HE. In this work, we use the same interpolation polynomial. The communication per keywords is $O(\sigma \log |X| + \ell)$ and the size of the entire database is about $|\mathbf{X}| \cdot \ell$ [29]. It also works for multiple keywords handling and can also be used for resource-constrained IIoT device.

IV. Proposed Scheme

As discussed above, in the past few years, a number of HE and its variants (such as FHE [29], SHE[14], Addititively HE [30]) have been developed in verifiable computation outsourcing in cloud based systems. This work aims at providing the data users different level of privacy preserving over data and protect data collected by the data owners from following four aspects:

1) Provide settings that allow the data users to disable access to sensitive information;
2) At cloud server, use the strongest data protection level for app. Use transport security when sending user or device data over the network;
3) The access for encrypted data over cloud server must be authorized by both data user and the cloud server;
4) Use the minimum amount of data required.

The conventional protocols such as CoAP (Constrained Application Protocol) [31] can provide IoT applications with secure communication, suitability, scalability, and privacy preservation. In typical IoT scenarios as shown in Figure 2, the IoT mediators, including `IoT gateway, router, Internet, Servers` can mainly provide by third part that cannot guarantee the security. In this work, we use server to present IoT mediators. Meanwhile, the resources limitations of IoT devices might restrict the security solutions over `IoT devices`. To address the privacy preservation in IIoT, we need to clear following major privacy problems:

- Privacy awareness, which provides data users with discovery of services' privacy properties; this concerns how IoT services might open communication channel between app and devices; Data users are not always aware when a device (data owner) present and collect data. A data user can be informed to the presence of the data owners (*e.g.*, PIR *sensor* ↔ *cloud servers* ↔ *app*); The data user do not have to clear if the data owners are alive or not.
- Privacy preferences, for cloud server, it might be able to learn from the data owner what data the device is collecting and what inferences might be possible; meanwhile, by analyse the queries from the data users, it can learn what does the data user actually care about.
- Privacy notification, IoT devices may not willing to interact to an IoT service in IIoT environment, *e.g.*, a temperature sensors might do not have to interact with an app (data user); However for some specific services, *e.g.*, product tracking in smart manufacture, the data user want to be alerted by the location sensor.

In IIoT, good security solution can ensure data is secure both in transit and at rest, including the confidentiality, integrity, and availability of data. From the viewpoint of privacy, good privacy means appropriate collection and use of information; the data collection/usage are transparent for data owners, cloud server, and data applications, respecting rights and choose of individuals. This work aims at addressing following key challenges:

A. Data Pre-processing

Placing obligations and restrictions on the collection and use of 'individual data'. The raw form of data at each IIoT device is the participant's privacy to be protected from both data users and third-part cloud server. The data links between two IIoT devices can be used to measure the privacy degree.

$$y = \mathcal{P}\left(\mathbf{x}_1, \ldots, \mathbf{x}_t\right) \tag{2}$$

in which $\mathbf{x}_i$ denotes the data at node D_i. A labeled program $\mathcal{P}$ is a tuple $(f, \ell_1, \ldots, \ell_n)$, and f is a homomorphic encryption allowed function of n variables and $\ell_i \in \{0,1\}^*$ is a label for the i-th input of f.

In this work, we introduce a lightweight data transmission scheme that includes following five algorithms: $\mathsf{KeyGen}(), \mathsf{Enc}(), \mathsf{Eval}()$ and $\mathsf{Dec}()$.

1) Key Generation Algorithm: this stage uses $\mathsf{keyGen}(1^\lambda)$ algorithm to create keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{vk})$, in which pk denotes a public key, sk denotes a secret authentication key, and vk denotes a verification key, respectively;

2) Encryption Algorithm: the $\mathsf{Enc}()$ algorithm creates ciphertext c, as $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x})$, in which $\mathbf{x}$ is the dataset, and ℓ is the label of $\mathbf{x}$.

3) Evaluation Algorithm: at the cloud server, when received a request for using data from an app (data user), the server will invoke the $\mathsf{Eval}(\mathsf{pk}, f, \mathsf{c})$ to create a new ciphertext based on the request $\mathcal{P}$.

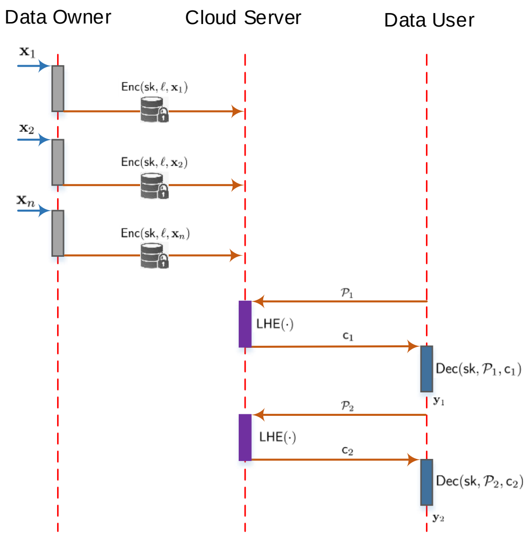
Data Owner
Cloud Server
Data User
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_n$
$\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_1)$
$\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_2)$
$\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_n)$
$\mathcal{P}_1$
$\mathsf{LHE}(\cdot)$
c_1
$\mathsf{Dec}(\mathsf{sk}, \mathcal{P}_1, \mathsf{c}_1)$
$\mathbf{y}_1$
$\mathcal{P}_2$
$\mathsf{LHE}(\cdot)$
c_2
$\mathsf{Dec}(\mathsf{sk}, \mathcal{P}_2, \mathsf{c}_2)$
$\mathbf{y}_2$

Figure 3. Single user data enc

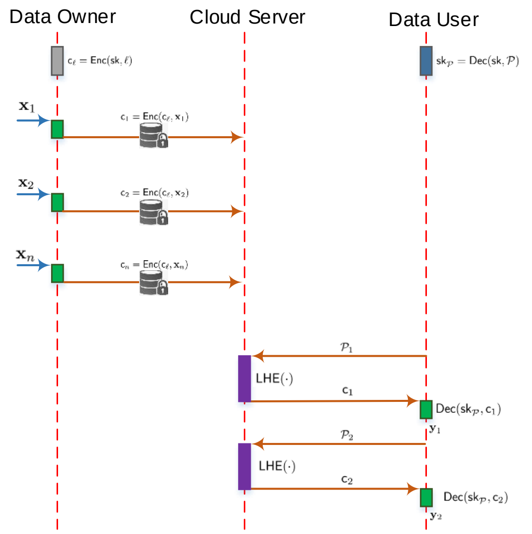
Data Owner
Cloud Server
Data User
$\mathsf{c}_\ell = \mathsf{Enc}(\mathsf{sk}, \ell)$
$\mathsf{sk}_\mathcal{P} = \mathsf{Dec}(\mathsf{sk}, \mathcal{P})$
$\mathbf{x}_1$
$\mathsf{c}_1 = \mathsf{Enc}(\mathsf{c}_\ell, \mathbf{x}_1)$
$\mathbf{x}_2$
$\mathsf{c}_2 = \mathsf{Enc}(\mathsf{c}_\ell, \mathbf{x}_2)$
$\mathbf{x}_n$
$\mathsf{c}_n = \mathsf{Enc}(\mathsf{c}_\ell, \mathbf{x}_n)$
$\mathcal{P}_1$
$\mathsf{LHE}(\cdot)$
c_1
$\mathsf{Dec}(\mathsf{sk}_\mathcal{P}, \mathsf{c}_1)$
$\mathbf{y}_1$
$\mathcal{P}_2$
$\mathsf{LHE}(\cdot)$
c_2
$\mathsf{Dec}(\mathsf{sk}_\mathcal{P}, \mathsf{c}_2)$
$\mathbf{y}_2$

Figure 4. Multiple user data enc

4) Decryption Algorithm: using the sk, with $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, \mathsf{c})$ the data user is able to verify and decrypt the message and label ℓ and decide reject or accept the data retrieved from the server.

B. Lightweight Privacy-Preserving Protocol

In IIoT, many applications (apps) may take computation tasks over encrypted data and looking for statistically relevant features across the encrypted data. Typically, such tasks are carried by computational powerful devices and each data query takes computational expensive algorithms such as $\mathsf{Enc}, \mathsf{Eval}, \mathsf{Dec}$, *etc.*, which stops the resource-constrained IIoT devices to perform this tasks in many applications, such as ehealthcare, smart home, *etc*. The proposed solution can shift the computation from IIoT devices to untrusted cloud server and offers a trade-off between privacy protection in terms of *data owner, cloud server, data user* and the computational efficiency.

Assume we have data space $\mathbf{x} \in \mathcal{X}$, label space $\ell \in \mathcal{L} \subset \{0, 1\}$, and an admissible circuits $\mathcal{F}$, including *multiplication, addition, and constant-multiplication* [13].

1) Single User Protocol: In a cloud system, for a data owner $\mathbf{A}$. The system generates keys $(\mathsf{pk}, \mathsf{sk}, \mathsf{ek})$ using $\mathsf{KeyGen}(1^\lambda)$. As shown in Figure 3, a data owner takes as input as the sk, a label $\ell \in \mathcal{L}$, and a message $\mathbf{x}_i \in \mathcal{X}$ to encrypt message using $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x})$ and outputs a ciphertext c, which will be uploaded to the cloud server $\mathbf{S}$. When a data user wants to use the data, it will sent a request of allowed arithmetic circuit f. The f takes as input the ek, ciphertexts $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_t$, the cloud server will perform the evaluation algorithm algorithm $\mathsf{Eval}(\mathsf{ek}, f, \mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_t)$ and return a ciphertext c to data user.

The data user takes as input the sk, a label program $\mathcal{P} = (f, \ell_1, \ell_2, \ldots, \ell_t)$, and the ciphertext c, using decryption function $\mathsf{Dec}(sk, \mathcal{P}, \mathsf{c})$ can output a message $f(\mathbf{x}')$, as

$$Pr[\mathbf{x}'] = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t) \tag{3}$$

is a negligibly close to 1, then the solution is said to correctly evaluate a $f \in \mathcal{F}$ for all keys.

In IIoT environment, a cloud server is more powerful than data owner and data users regarding computing capabilities and storage capacity. However, for an IIoT node, it has to run $\mathsf{Enc}(sk, \ell, \mathbf{x}_i)$ when each time the node create a message $\mathbf{x}_i$, which is computation expensive for resource-constrained nodes. Meanwhile, for the data user (always not powerful devices) it needs to run computational expensive $\mathsf{Dec}(sk, \mathcal{P}, c)$ for each data query.

A straightforward idea is to outsource the computation at data owner and data users to the cloud servers. Using the labeled homomorphic encryption scheme [13], it is possible to reduce computation of $\mathsf{Enc}(\cdot)$ and $\mathsf{Dec}(\cdot)$, as shown in Figure 4. It can not only provide lightweight data encryption, but also match the privacy requirements addressed in Section III.

2) Computation Split at Data Owners and Data Users: It is noted that the computation cost at data owner comes from $\mathsf{Enc}(\cdot)$, for each piece of data $\mathbf{x}_i$, the data owner has to perform $\mathsf{Enc}(\mathsf{sk}, \ell, \mathbf{x}_i)$. Since the (sk, ℓ) are always the same, a straightforward way is to divide the $\mathsf{Enc}(\cdot)$ into two parts

$$\mathsf{c}_\ell = \mathsf{Enc}(\mathsf{sk}, \ell) \tag{4}$$

$$\mathsf{c}_i = \mathsf{Enc}(\mathsf{c}_\ell, \mathbf{x}_i) \tag{5}$$

Eq.(4) takes as input a label and the secret key and output ciphertext c_ℓ, which can be reused by messages created by the same node. Both Eq.(4) and Eq.(5) is homomorphically correct in the sense that $\mathsf{Enc}(sk, \ell, \mathbf{x}_i)$ [13]. As shown in Figure 4, a data user must be able to correct decrypt the ciphers. Similarly, at a data user, given sk and an allowed circuits, the decryption algorithm can be divided into two parts

$$\mathsf{sk}_\mathcal{P} = \mathsf{Dec}(\mathsf{sk}, \mathcal{P}) \tag{6}$$

$$\mathbf{x}'_i = \mathsf{Dec}(\mathsf{sk}_\mathcal{P}, \mathsf{c}) \tag{7}$$

Here both Eq.(6) and Eq.(7) are homomorphically correct in the sense that $\mathsf{Dec}(\mathsf{sk}, \mathcal{P}, \mathsf{c})$ [13]. This splits the computation into a fixed function and a dynamic function with dynamic computation complex.

In IIoT environment with multiple IIoT devices, the key generation algorithm will generate a master public key and a master secret key $\mathsf{KeyGen}(1^\lambda) \rightarrow (\mathsf{msk}, \mathsf{mpk}, \mathsf{uek})$. The master public key will be used to generate a public key associated with its encrypted data. Each data owner can encrypt using its public keys and user's master public key (mpk) to guarantee the data encrypted by a data user can not be decrypted by another data user. Decryption requires the msk along with pk of data owner/users involved.

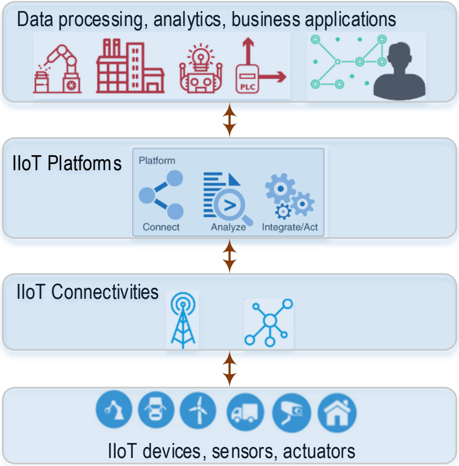
Data processing, analytics, business applications
PLC
IIoT Platforms
Platform
Connect
Analyze
Integrate/Act
IIoT Connectivities
IIoT devices, sensors, actuators

Figure 5. IIoT Infrastructure

For an IIoT node D_i, the encryption algorithm $\mathsf{Enc}(\mathsf{mpk}, \mathsf{usk}, \ell_i, \mathbf{x}_i)$ outputs a ciphertext c_i. At server $\mathbf{S}$, evaluation algorithm will perform $\mathsf{Eval}(\mathsf{mpk}, f, \mathsf{c}_1, \ldots, \mathsf{c}_t)$ and returns a ciphertext c_s. A data user D_j, for example, can use $\mathsf{Dec}(\mathsf{sk}, \mathsf{upk}, \mathcal{P}, \mathsf{c}_s)$ returns a message $\mathbf{x}'$. For all honestly generated keys $(\mathsf{mpk}, \mathsf{msk})$, all data user key pairs $(\mathsf{upk}_i, usk_i), \ldots, (\mathsf{upk}_t, \mathsf{usk}_t)$, if we have

$$Pr[\mathbf{x}'] = f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t) \tag{8}$$

is negligibly close to 1, then we can say the protocol is privacy secure for data owner, server, and data users.

Correctness: To guarantee the correctness of above algorithms, the system applies $\mathbf{Eval}(\cdot)$ on data and label $\mathsf{c}_1, \ldots, \mathsf{c}_n$ corresponding to the label set ℓ respectively, the result will be a ciphertext $\mathsf{c}_{(f,\mathbf{x})}$ that verifies against function f, labels ℓ, and message $f(\mathbf{x}_1, \ldots, \mathbf{x}_n)$.

Theorem 1. In semi-honest model, the proposed protocol can solve the above privacy-preserving problems between multiple data owners and multiple data users, in which a data owner or data user can following the lightweight protocol to keep privacy without data leaking.

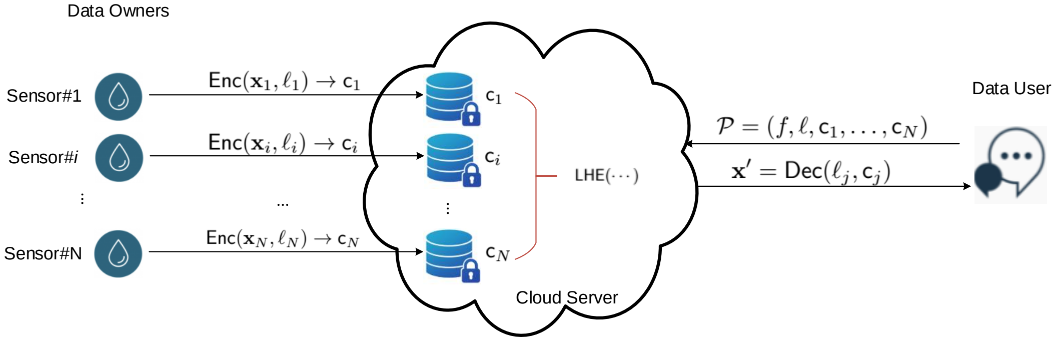
Data Owners
Sensor#1
Sensor#i
⋮
Sensor#N
$\mathsf{Enc}(\mathbf{x}_1, \ell_1) \to \mathsf{c}_1$
$\mathsf{Enc}(\mathbf{x}_i, \ell_i) \to \mathsf{c}_i$
...
$\mathsf{Enc}(\mathbf{x}_N, \ell_N) \to \mathsf{c}_N$
c_1
c_i
⋮
c_N
LHE(···)
Cloud Server
Data User
$\mathcal{P} = (f, \ell, \mathsf{c}_1, \ldots, \mathsf{c}_N)$
$\mathbf{x}' = \mathsf{Dec}(\ell_j, \mathsf{c}_j)$

Figure 6. Lightweight Privacy Data Encryption Protocol

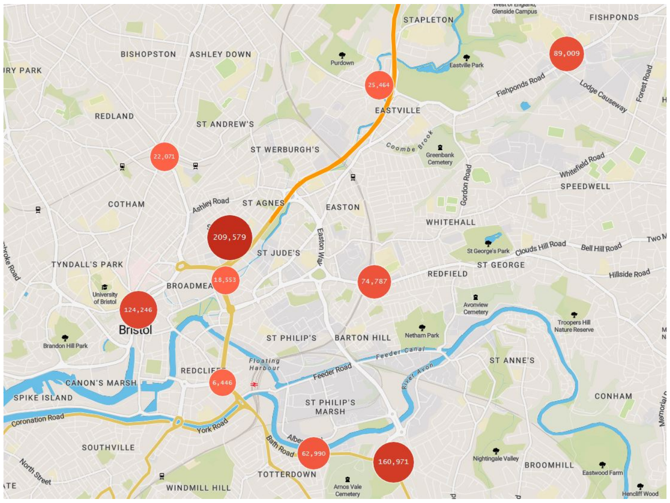

Figure 7. Air condition acquisition sites in Bristol

Proof. Security: suppose the protocol is insecure. Then there is a probabilistic polynomial time-real adversary $\mathcal{A}$ that does not have corresponding PPT ideal adversary $\mathcal{A}'$ exists that makes $(P_1(\mathbf{x}_1), \ldots, P_n(\mathbf{x}_n), \mathcal{A})$ and $(P_1'(\mathbf{x}_1), \ldots, P_n'(\mathbf{x}_n), \mathcal{A}')$ are computational indistinguishable. □

Basically, an IIoT system consists of four main entities: *intelligent assets*, IIoT *infrastructure*, *analytics* and *applications, users*. Figure 5 shows an architecture of an IIoT system, in which the *intelligent assets* could be the IoT sensors, actuagors, industry machineries, *etc.*, that can generate and store data; The IIoT infrastructure provide connectivities between components in IIoT. Applications or devices that can consume data are data users.

- A smart sensor or actuator, *for example*, can generate, collect, process, transmit, or store data and an actuator is able to conduct physical functions, such as closing or opening a door when a command is conducted.
- IIoT infrastructure, includes IoT gateway, IIoT platforms, can provide connectivities between components in IIoT and allow data real-time analysis. The most popular IoT platforms include Azure IoT hub, where device data is sent to the cloud.
- Analytics and applications, it provides detailed analysis of remaining data and relies on the applications that an IIoT can provide, where data-intensive processing and analysis takes place.
- IIoT entities, include IIoT users, service providers, customers, *etc.*

V. Evaluation and Discussion

The increasing popularity of IIoT shows great promises both new conveniences and new privacy concerns. In this section, we will introduce an air quality monitoring applications in Bristol. We examine an IIoT scenario as shown in Figure 6 to test proposed solutions that allow the data owner and data users to protect the themselves from smart home privacy vulnerabilities. The *sensors* are deployed at different sites as shown in Figure 7 and Table I. The dataset collected by these sensors are provided by the Bristol council and this case studies we have performed motivated components of a general IIoT solution. To evaluate the proposed scheme, we use the air quality dataset to test above data. The data is acquired from 15 sites in Bristol area as shown in Table I. The location of these sites are highlighted in Figure 7.

In the test, the data captured by sensors at*AURN St Pauls* were encrypted and uploaded to a third part cloud server at UWE. We define these sensors are *data owner*, the `data user` is an app over mobilephone that can send query to the server. The raw data was labeled using the timestamp, the query $\mathcal{P} = \{f_1, f_2\}$, f_1 is a $const - multiplication$ algorithm, and f_2 is an $addition$. When enquiring the air quality data at *AURN st Pauls*, the `data user` app sends a query $\mathcal{P} = (f_1, 05082019)$, in which f_1 is the cost-multiplication and constant was defined as 1, and the $\ell = 05082019$ is the time-label. When send a query for asking the quality data of 05 Aug 2019.

We tested the proposed scheme over a server with Ubuntu 18.04, Intel Core i7 7700K Quad Core Dedicated Server ($4.2GHz \times 8$), 32GB RAM. In the test, the size of sk, pk, ℓ, and $Nounce$ is 128 bits. Privacy analysis: (1) The data owner ID_{AURN} learns nothing from other data owners (such as $ID_{Fishpond}, etc.$; (2) The cloud server learns nothing from ID_{AURN}, and learns nothing from the data user as well; (3)

Table I
AIR QUALITY DATA CONTINUOUS (ug/m^3)

ID	Data Onwer	Records	NOx	NO_2	NO	PM10	PM2.5
203	Brislington Depot (BLD)	160,970	74.9	30.0	29.2	–	–
215	Parson Street School (PSS)	152,188	116.2	32.3	54.1	–	–
270	Wells Rd (WLR)	140,021	161.7	49.5	73.1	–	–
206	Rupert Str (RPS)	113,951	280.3	90.0	120.8	–	–
452	AURN St Pauls (ASP)	113,171	55.8	5.9	32.6	20.3	6.0
375	N. Road Station (NRS)	96,407	65.0	29.8	23.0	–	–
463	Fishponds Rd (FPR)	91,347	65.9	38.3	18.0	–	–
395	Shiner's Garage (SSG)	74,787	174.8	73.3	101.8	–	–
447	Bath Rd (BAR)	62,990	91.5	47.3	44.3	–	–
271	Trailer Portway P (TPP)	43,824	61.5	35.7	16.8	–	–
209	IKEA M32 (IKM)	25,464	159.0	77.6	51.1	–	–
459	Cheltenam Rd (CTR)	22,071	72.5	38.3	22.3	–	–
500	Temple Way (TPW)	18,552	16.8	12.0	3.1	14.5	–
501	Colston Avenue (CSA)	10,294	52.9	25.8	17.7	30.8	–
228	Temple Meads St. (TMS)	6,446	1,174.3	122.0	680.3	–	–

Table II
TIMINGS OF THE DATA QUALITY QUERIES

key size (bits)	label size	Operations	Times
128	1024	$\mathsf{Enc}(\mathsf{sk},\ell)$	$150.7ms$
128	1024	$\mathsf{Enc}(\mathsf{c}_l,\mathbf{x})$	$0.06ms$
128	1024	$dec(\mathsf{sk},\mathcal{P})$	$2.204ms$
128	1024	$\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}},\mathsf{c})$	$0.723ms$
128	1024	add	$0.001ms$
128	1024	mul	$0.042ms$
128	2048	$\mathsf{Enc}(\mathsf{sk},\ell)$	$557.725ms$
128	2048	$\mathsf{Enc}(\mathsf{c}_l,\mathbf{x})$	$0.055ms$
128	2048	$dec(\mathsf{sk},\mathcal{P})$	$6.083ms$
128	2048	$\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}},\mathsf{c})$	$2.685ms$
128	2048	add	$0.001ms$
128	2048	mul	$0.151ms$

the data user learns nothing from other data users. Table III addresses the air quality data at *AURN St Pauls*.

The homomorphic addition operation takes $0.001ms$, and homomorphic mul operation takes $0.151(ms)$. In the proposed solution, the $\mathsf{Enc}(\mathsf{sk},\ell)$ takes $557.725ms$, $\mathsf{Enc}(\mathsf{c}_l,\mathbf{x})$ takes $0.055(ms)$, at the data user app, the $\mathsf{Dec}(\mathsf{sk},\mathcal{P})$ takes $6.083ms$, the $\mathsf{Dec}(\mathsf{sk}_{\mathcal{P}},\mathsf{c})$ takes $2.685ms$, it is clear that the proposed scheme can significantly reduce the computation costs at both data owner and data user. Table II compares the time consuming of each computing component in proposed scheme when the key size label size are different.

Each site first generates keys using $\mathsf{KeyGen}() \rightarrow (\mathsf{ek},\mathsf{sk},\mathsf{vk})$, and then share sk and pk with server. Each node then encrypts the data using $\mathsf{Enc}(\mathsf{sk},\ell,\mathbf{x})$ to generate labeled ciphers and transmitted to cloud server. In this work, each data owner submitted its data to a untrusted cloud server and the server allocate it a label, ℓ_{BLD}, the ID of BLD is 203. The cloud server with Eval the data before doing further processing. The data user decrypted results are exactly the same with the original data at the data owner, as shown in Table III. The result shows as $NOx = 59.9$, $NO2 = 10.3$, $NO = 32.3$, $PM10 = 15.5$, $PM2.5 = 7.0$.

Furthermore, when an application needs to access the data, *e.g.*, someone wants to know the average temperature in Fishponds Rd in the past five years (from 2014 - 2019), he may send a function f_{avg} to server together with the label ℓ_{FPR}, the server can perform a homomorphical and return the results.

In practical, IoT devices and applications create or collect huge amount of data, which is a valuable business asset. Typically, the data owner is the device or app that generated the data itself. In some cases, the data is aggregated before being encrypted and sent to cloud server. New classes of IoT devices that can collect information that was not previously available are emerging, such as *wearable, smart sensors, etc.*, individual data in industry networks are facing increasing sophisticated cyberattacks. As a very modern development, the IIoT has to facing new challenges using new approaches include lightweight cryptography, computational intelligence, and distributed ledger technology (DLT) techniques.

VI. CONCLUSION

In this paper, we presented the privacy between data owners, third part cloud server, and data users. A lightweight privacy preserving protocol for the data owner-server-data user model is proposed based on the labeled HE. The evaluation computation costs are shifted from resourced constrained IoT devices to third part powerful server without losing security and privacy strength. It is an efficient and practical scheme in IIoT environment and allows a remote, non-confident, cloud computing to perform complex computational over encrypted data, and permits the data owner, data users to verify the exactitude of decrypton. We further propose effective protocols

Table III
AIR QUALITY DATA AT ASP ON 5TH AUG 2019 (ug/m^3)

ID	Data Onwer	NOx	NO_2	NO	PM10	PM2.5
452	AURN St Pauls (ASP)	59.9	10.3	32.3	15.5	7.0

that can be easily incorporated with existing IIoT and so that can significantly reduce the computational costs.

REFERENCES

[1] K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, Aug 2018.

[2] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2497–2505, April 2019.

[3] J. Zhang, Z. Wang, L. Shang, D. Lu, and J. Ma, "Btnc: A blockchain based trusted network connection protocol in iot," *Journal of Parallel and Distributed Computing*, vol. 143, pp. 1 – 16, 2020.

[4] L. Zhou, K. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial internet of things: An overview of approaches to safeguarding endpoints," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, Sep. 2018.

[5] K. R. Choo, S. Gritzalis, and J. H. Park, "Cryptographic solutions for industrial internet-of-things: Research challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, Aug 2018.

[6] S. Jana, D. Molnar, A. Moshchuk, A. Dunn, B. Livshits, H. J. Wang, and E. Ofek, "Enabling fine-grained permissions for augmented reality applications with recognizers," in *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C.: USENIX, 2013, pp. 415–430.

[7] K. Fan, W. Jiang, H. Li, and Y. Yang, "Lightweight rfid protocol for medical privacy protection in iot," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1656–1665, April 2018.

[8] R. Yugha and S. Chithra, "A survey on technologies and security protocols: Reference for future generation iot," *Journal of Network and Computer Applications*, vol. 169, p. 102763, 2020.

[9] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in iot-enabled smart metering systems," *Future Generation Computer Systems*, vol. 78, pp. 547 – 557, 2018.

[10] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3590–3598, 2018.

[11] S. N. Foley, D. Gollmann, and E. Snekkenes, "Computer security–esorics 2017."

[12] J. H. Cheon and T. Takagi, *Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*. Springer, 2016, vol. 10032.

[13] M. Barbosa, D. Catalano, and D. Fiore, "Labeled homomorphic encryption - scalable and privacy-preserving processing of outsourced data," *IACR Cryptology ePrint Archive*, vol. 2017, p. 326, 2017.

[14] J. H. Cheon and J. Kim, "A hybrid scheme of public-key encryption and somewhat homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, May 2015.

[15] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced iot," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2497–2505, April 2019.

[16] Z. Ma, A. Hudic, A. Shaaban, and S. Plosz, "Security viewpoint in a reference architecture model for cyber-physical production systems," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, April 2017, pp. 153–159.

[17] H. Mahdikhani, R. Lu, Y. Zheng, J. Shao, and A. A. Ghorbani, "Achieving $o(log^3 n)$ communication-efficient privacy-preserving range query in fog-based iot," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5220–5232, 2020.

[18] M. Alloghani, M. M. Alani, D. Al-Jumeily, T. Baker, J. Mustafina, A. Hussain, and A. J. Aljaaf, "A systematic review on the status and progress of homomorphic encryption technologies," *Journal of Information Security and Applications*, vol. 48, p. 102362, 2019.

[19] M. M. H. ONIK, C. KIM, and J. YANG, "Personal data privacy challenges of the fourth industrial revolution," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, Feb 2019, pp. 635–638.

[20] D. Fiore, A. Mitrokotsa, L. Nizzardo, and E. Pagnin, "Multi-key homomorphic authenticators," in *Proceedings, Part II, of the 22Nd International Conference on Advances in Cryptology — ASIACRYPT 2016 - Volume 10032*. New York, NY, USA: Springer-Verlag New York, Inc., 2016, pp. 499–530.

[21] M. Beunardeau, A. Connolly, R. Geraud, and D. Naccache, "Fully homomorphic encryption: Computations with a blindfold," *IEEE Security Privacy*, vol. 14, no. 1, pp. 63–67, Jan 2016.

[22] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "Cpal: A conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 46–57, Feb 2014.

[23] C. Dsouza, G. Ahn, and M. Taguinod, "Policy-driven security management for fog computing: Preliminary framework and a case study," in *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, Aug 2014, pp. 16–23.

[24] H. Xiong, Q. Mei, and Y. Zhao, "Efficient and provably secure certificateless parallel key-insulated signature without pairing for iiot environments," *IEEE Systems Journal*, vol. 14, no. 1, pp. 310–320, 2020.

[25] Y. Lu, J. Li, and Y. Zhang, "Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted iiot," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2553–2562, 2020.

[26] A. Al-Hasnawi, I. Mohammed, and A. Al-Gburi, "Performance evaluation of the policy enforcement fog module for protecting privacy of iot data," in *2018 IEEE International Conference on Electro/Information Technology (EIT)*, May 2018, pp. 0951–0957.

[27] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology – CRYPTO 2013*, R. Canetti and J. A. Garay, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 75–92.

[28] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography Conference*. Springer, 2005, pp. 303–324.

[29] H. Chen, Z. Huang, K. Laine, and P. Rindal, "Labeled psi from fully homomorphic encryption with malicious security," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 1223–1237.

[30] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, May 2018.

[31] S. Li, S. Zhao, P. Yang, P. Andriotis, L. Xu, and Q. Sun, "Distributed consensus algorithm for events detection in cyber-physical systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2299–2308, April 2019.