Designing Privacy Preserving Data Sharing Middleware for Internet of Things

Sameera Ghayyur, Primal Pappachan, Guoxi Wang, Sharad Mehrotra, Nalini Venkatasubramanian
University of California, Irvine
{sghayyur,primal,guoxiw1}@uci.edu
{sharad,nalini}@ics.uci.edu

ABSTRACT

The rise of low-cost Internet of Things (IoT) sensing and communication capabilities has given rise to a range of new smart services that rely on heterogeneous data from devices embedded in our everyday lives. The provision of such IoT services relies on environmental or user data from other data controllers (e.g. network provider, water agency, building management). Recent privacy regulations such as the European General Data Protection Requirement (GDPR) and California Consumer Privacy Act (CCPA) have made it mandatory for data controllers to perform enhanced processing of the shared data with appropriate privacy-preserving mechanisms before release to service providers. To facilitate this, we propose PE-IoT, a system for orchestrating privacy-enhanced data flows that (a) provides users (data subjects) with capabilities to opt-in/opt-out in the data that is shared with the service providers and (b) enable data controllers to invoke a range of Privacy Enhancing Technologies (PETs) such as anonymization, randomization, and perturbation to transform data streams into their privacy preserving counterparts. PE-IoT is based on a new model for privacy compliant data sharing and we describe the design and architecture of the PE-IoT system based on this model.

CCS CONCEPTS

• **Security and privacy → Information accountability and usage control**; **Data anonymization and sanitization**; **Privacy protections**.

KEYWORDS

Privacy, Internet of Things, User consent, Data Privacy Regulations, Information Flow Control

ACM Reference Format:
Sameera Ghayyur, Primal Pappachan, Guoxi Wang, Sharad Mehrotra, Nalini Venkatasubramanian . 2020. Designing Privacy Preserving Data Sharing Middleware for Internet of Things. In *The 3rd International SenSys+BuildSys Workshop on Data:Acquisition to Analysis (DATA '20), November 16–19, 2020, VirtualEvent, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3419016.3431484

The first three authors have made equal contributions to the paper.

DATA '20, November 16–19,2020, Virtual Event, Japan

ACM ISBN 978-1-4503-8136-9/20/11.
https://doi.org/10.1145/3419016.3431484

1 INTRODUCTION

The phenomenal growth of the Internet of Things (IoT) ecosystem with a multitude of embedded devices and high speed network connectivities has resulted in the ability to pervasively monitor physical spaces and environments. Today, *service providers* leverage heterogeneous real-time sensor data collected from IoT deployments to provide a broad range of IoT services to improve operational efficiency, occupant comfort and safety e,g. class attendance in schools[10], occupancy analysis [11] for HVAC control and fire safety. Typically, *data controllers* - who own the space and devices produce and collect sensor data that is shared with multiple service providers who build services utilizing the data. Data streams flow into a *data ingest* platform from where it is distributed to service providers; additionally, the data may be persisted in a database management system (DBMS) for post facto storage/querying. Service Level Agreements (SLAs) between data controllers and service providers dictate specifics of when, how much and at what quality data is shared. An often ignored party in this data flow are *data subject*s, i.e. individuals who are present in the IoT space and whose data are captured by the sensors.

The personal nature of information that can be captured in instrumented physical spaces poses significant privacy risks for data subjects [12, 14]. The introduction of stringent privacy laws (e.g General Data Protection Act (May 2018) and California Consumer Privacy Act (Jan 2020)) has stimulated the a need for a redesign of data processing systems that are compliant with these regulations. Regulatory compliance is challenging since it involves additional processing overheads. Moreover, implementing the required functionality is often in conflict with the design and operation of modern systems where persistence of data is inherent (e.g. storing data forever, reusing data indiscriminately, etc.)[17]. The following articles from the GDPR[18] specification require enhancements to current systems to articulate collection purpose, maintain audit logs and support erasure and form the design requirements to make today's systems run by data controllers regulation compliant:

- RQ.1) User data must be only collected and processed for specific purposes (Article 5)
- RQ.2) Users (Data subjects) should explicitly allow sharing of their data with other service providers (Article 14)
- RQ.3) Best effort should be made for implementing appropriate data security (Article 32)
- RQ.4) Audit logs of all operations must be stored to demonstrate compliance (Article 30)
- RQ.5) Data collected must be not stored indefinitely (Article 5)

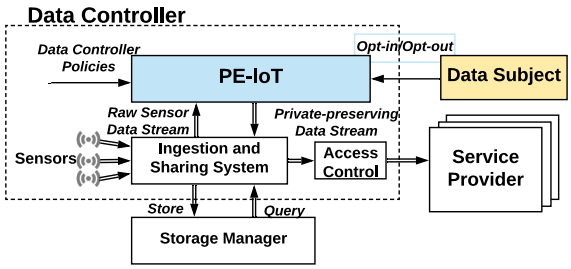

Figure 1: High level view of components and flow of sharing IoT sensor data with the proposed system PE-IoT

Our own experiences with enabling privacy-preserving dataflows for real-world IoT services are derived from the TIPPERS [3] and deployment at UC Irvine. Through a close collaboration with the UCI Office of Information Technology (OIT), WiFi association data on campus was used to develop services such as occupancy monitoring and COVID-19 exposure tracing in campus buildings. In this setting, WiFi association data from individual devices (data subjects) captured by OIT (data controller) are shared with the TIPPERS platform (service provider), subject to appropriate privacy enhancing procedures. This required design of a new data model and a privacy-compliant sensor data sharing system - we refer to this platform as *PE-IoT (Privacy Enhanced-Internet of Things)*.

PE-IoT is designed as a middleware that intercepts information flow in existing IoT data processing systems to add privacy-compliance (See Figure 1 during data ingest and sharing). Specifically, PE-IoT gets raw sensor data streams from an *ingestion and sharing system* and produces corresponding privacy-enhanced data streams by enforcing data subject policies and applying suitable Privacy Enhancing Technologies (PETs). We introduce the concept of a *Data product*, an abstraction shared by various actors involved in the PE-IoT dataflow. Operationally, PE-IoT is deployed by the data controller that takes in incoming raw sensor data-streams and transforms them into (privacy-enhanced) data products by enforcing organizational and individual policies that implement privacy regulations. The data products, thus generated, are shared with service providers based on their needs through *Access Control*. PE-IoT is also capable of storing the sensor stream data from the data controller using a *Storage Manager*. This stored data can be accessed by service providers through PE-IoT which then generates Data Products based on the past data. PE-IoT also provides timely deletion, logging based auditing, and encryption at rest to meet other GDPR design requirements outlined earlier. In the remainder of this paper, we introduce the data model behind PE-IoT, design and architecture of the system based on this model, and discuss research and implementation challenges.

2 PE-IOT DATA MODEL

PE-IoT receives sensor data streams from multiple sensors where each tuple of a sensor stream is associated with a data subject if it captures personal information. For example, for Wi-Fi AP association data, the owner of the device that connects to the infrastructure is the data subject[1]. In this section, we introduce the PE-IoT data model that captures the design principles derived through privacy laws. This model is also developed to minimize the overhead for data subjects and controllers in order to fulfill their rights and responsibilities.

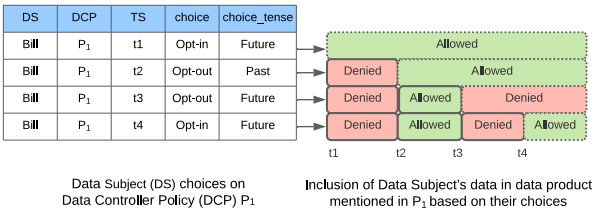

Figure 2: Retroactive Policy Semantics

We propose *Data Product* as an abstraction for privacy enhanced data that data controllers can share with service providers. Data products are created using incoming sensor streams are the unit used by data controllers to share data with service providers. More formally, a data product is defined as: θ=($Filter$, $Policy$, PET) where *Filter* is a set of selection conditions on sensor data that produce a subset of the sensor data that constitutes a data product. For example, for a WiFi association data set, a possible filter might be based on a particular building. *Policy* in a data product consists of two components: a data controller's sharing policies that specify conditions under which a service provider can access the data product, and the data subject's choices that determine the inclusion of a data subject's data in a data product. *PET* defines the information about the Privacy Enhancing Technology (PET) that a data controller uses when sharing the data product with service providers. We further elaborate on these concepts below.

2.1 Data Controller Policy and Data Subject Choices

Policy captures the purposes for which the data product is shared by the data controller and provides a mechanism for data subjects to opt-in/opt-out of the data product. This satisfies the regulation requirement that user data is only collected and processed for specific purposes (RQ.1) and sharing of data is explicitly allowed by the users (RQ.2). The data controller policy for sharing a data product (θ_j) is specified as DCP_i = <θ_j, pa_1...pa_n>. Each policy attribute pa_i consists of a set of tags and represents the metadata associated with a data product. Examples of policy attributes are purpose and the category of service providers who can gain access to the data product. A possible set of policy attributes for a data controller policy on a data product (θ_1) may be pa_1, pa_2 where pa_1 = [*"COVID-19-tracking", "UCI-health"*] indicating that the health officials at UCI can access the data for tracking COVID-19, and pa_2 = [*"Occupancy", "UCI-facilities"*] indicating that θ_1 is accessible to facility entities from UCI campus for determining building occupancy. [2]. Each service provider is associated with the set of attributes that characterize the service provider [3]. A service provider advertises itself using its set of attributes when requesting access to a data product. This set of attributes is matched against the policy attributes associated with data product (through data controller

[1]In out current design, we assume each data is associated with one data subject at most for simplicity. Please refer to Section 4 for further discussion on handling multiple data subjects in a data tuple.

[2]The implementation of this policy model will be based on a simplified subset of the model presented in our own work [15]

[3]Validation of service provider attributes is done by a trusted third party.

policy) to determine if that data product can be shared with the service provider.

A data subject can choose to opt-in or opt-out of the participation in a specific data product. A data subject choice is modelled as $< DS, DCP, choice, TS, choice\_tense >$ where DCP is the data controller policy data subject DS is opting in or opting out ($choice$) at timestamp TS. The $choice\_tense$ is used to denote whether the action (of opting-in/out) applies to data subject's *future* data or *past* data). The choice_tense is used by a data subject to retroactively to opt-in/opt-out from inclusion of their data in a data product. In Figure 2 we illustrate how this retroactive policy semantics works. Bill has opted in for the data product sharing policy P_1 at timestamp $t1$ with choice_tense as $future$ after which his data is allowed to be included in the data product corresponding to P_1. Later on, at $t2$ Bill opt-outs with choice_tense as past and therefore his data is denied from historical queries to that data product. At $t3$, when Bill opts-out with choice_tense as future, his data is denied from the data product but his past data between $t3$ and $t4$ is allowed to be included in that data product. Finally, at $t4$ when he opt-outs with choice_tense as future, his data is denied from being included in the data product again.

If the data subject choice for a data product is opt-in, then the user is opting in to sharing their data with any service provider that satisfies one of the policy attributes in the data controller's policy. Likewise, if it is opt-out, then they are opting out of sharing with any service providers who have access to that data product. Each data product is associated with the default choice (opt-in or opt-out) and choice_tense (future or past) for data subjects which are set by the data controller. The data controller policies and data subject choices are stored in the database.

2.2 Privacy Enhancing Technologies

Due to the recently adopted privacy regulation laws, there has been a significant interest in developing technologies that ensure individual person's privacy. In the literature, a diverse range of Privacy Enhancing Technologies (PETs) have been proposed that allow sharing the data in a privacy preserving manner. These PETs use different techniques e.g. removing personally identifiable information, introducing noise, encryption etc. to prevent the leakage of sensitive information about an individual. The PETs differ in terms of the provided privacy guarantees, underlying assumptions about the adversary and release of aggregate vs individual level data. The adaptation of a specific PET depends on the context of the application. For example, *differential privacy* [6] has emerged as the leading technique for aggregate data analysis with formal privacy guarantees. Differential privacy gives a mathematically rigorous worst-case bound on the maximum amount of information that can be learned about an individual's data from the output of a computation. It assumes a very strong attacker that know about all but one record in the data. The privacy parameter ϵ is used to control the privacy level where lower ϵ means higher privacy. Another example of a PET is randomization of personalized identifiers over time for release of individual level data. Randomization although does not provide a formal guarantee, it is practical and simplistic and has been used to offer sufficient privacy e.g. COVID-19 Alert app for contact tracing by Apple and Google[2]. There are several such PETs specifically designed for streaming data [4, 5, 8, 13, 16].

In the context of PE-IoT, data owner may want to leverage these technologies to share privacy preserving data instead of sharing raw sensitive data with the service providers. Due to the diverse range of existing PETs and it being an emerging area of research, we design PE-IoT to enable integration of various Privacy Enhancing Technologies. We classify the PETs based on their properties that can influence the system design to enable seamless integration of PETs as follows:

• *Stateful vs stateless*: Stateful PETs maintain a "state" that is shared between events and therefore past events can influence the way current events are processed. An example of a stateful PET is Randomization of personal identifiers in a sensor stream. The personal identifiers of users are replaced with randomly generated identifiers after every t time unit. During each window of size t, all the events capturing a single user use the same randomly generated identifier i.e. state. In a stateless PET, past events do not influence the current events thus does not require maintenance of state.

• *Blocking vs non-blocking*: Blocking PETs typically require processing the entire input before an output can be delivered. An example of a blocking PET is Differentially Private Laplace Mechanism for releasing aggregate statistics over time. A non-blocking PET does not need to wait of all the tuple in a window to arrive before applying the PET. The tuples are processed as they come. An example of a non-blocking PET is Randomization of personal identifiers in a sensor stream.

• *Negotiable vs Non-negotiable*: A data product using PET which is non-negotiable is made available to service providers with a fixed privacy model. An example is a data product consisting of a deferentially private sensor stream based on a fixed ϵ. Such data products come with strong privacy guarantees but may not provide any bound on utility of the data product to the end-application/service. A different model is to support data products with negotiable PETs. In such a case, the data product may come with strong privacy properties, but the service provider is capable of negotiating with the data owner about the level of noise/anonymization added to the data product if the data product is unable to meet the utility goals of the applications. Negotiable PETs, specially, in the context of Differential Privacy, is new emerging concept - traditionally, Differential Privacy has explored algorithms to optimize utility with strict privacy constraints. Negotiable privacy offer more flexibility by making utility more central to the way data products are produced for sharing. In particular, it shifts the privacy-utility trade-off problem from optimizing utility given privacy constraint to that of optimizing privacy (i.e., minimizing the privacy loss) given utility constraint [7, 9]. In negotiable PET, the service provider can request for data product with sliding scale privacy based on the demonstration of need for accuracy. In the Section 3, we describe how these classifications influence the system design.

Discussion: The above data model used by PE-IoT offers a trade-off between fine grained control versus overhead of privacy interventions. Each data product is created from a raw sensor stream by executing the filter followed by evaluating the data subject choices, and finally by implementing the PET on top of it. PET execution is expensive and imposes overhead on data controllers and therefore should be limited. By utilizing the data controller policies which

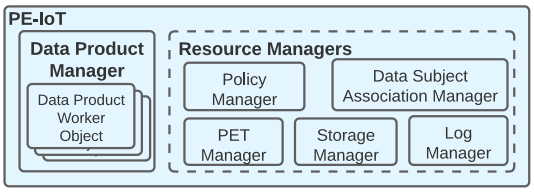

Figure 3: The prototype PE-IoT system's components

group together different service providers with tags, the data controller has to only generate one data product for them (reducing the overhead of PET execution). Additionally, the utilization of PE-IoT model reduces the burden of frequent participation in privacy intervention for data subjects. By opting out of a single data controller policy, a data subject opts out of any of the service providers that has access to this data product. Retroactive policy semantics provides a flexible and powerful method for data subjects to control sharing of their future and historical data. They can opt-in for sharing with their streaming data but have the freedom to opt-out of sharing with their historical data with service providers later on. For example, a student might be okay sharing their live location data to be used with location based services when they are enrolled at the university but they might choose to opt-out of any analysis performed on their data after graduation.

3 SYSTEM DESIGN AND WORKFLOWS

We discuss a prototype system design of PE-IoT (See Figure 3) that realizes the data model design presented. This prototype consists of a *Data Product manager* which conducts the flow of intercepted sensor stream through PE-IoT and coordinates with different resource managers to produce the units of privacy-preserving data stream defined by Data Products. The different aspects of PE-IoT data model are implemented as independent resource managers which can be executed separately. The rationale for such a system design are two-fold: 1) decoupling functions allows each resource manager to perform its tasks independently, 2) some of the resources (e.g., data controller policy, PET) might be stored remotely and independent resource managers allow us to move them closer to the resource. Each resource manager also include interfaces to interact with the Data Product Manager, and other resource managers. The different components in PE-IoT system are

• *Data Product Manager:* provides the most essential function in PE-IoT of instantiating data products to be shared with the Service Providers in compliance with privacy requirements specified by the data controller and data subjects.For each data product,it launches a Data Product Worker Object for each defined data product. The Data Product Workflow Object manages the data product creation workflow with other resource managers (e.g., Policy Manager, PET Manager) to create a data product. The primary function of the Data Product Worker Object is to handle and monitor data flow, and call resource managers to perform the functions. The Data Product manager is also responsible for state management for data products to handle failure and recovery.

• *Data Subject Association Manager* creates the association between data records in the sensor stream and a data subject. The implementation of a Data Subject Association Manager depends on the specific sensor stream it is handling and therefore there will be as many of these managers as there are different sensor data types. For example, in WiFi association data stream, the MAC address captured is mapped to a corresponding data subject by looking up the device registry.

• *Policy Manager* meets the design requirement of ensuring that data sharing with service providers is explicitly allowed by users (RQ.2). It handles creating, modifying, and updating of data controller policies. It also stores the data subject choices associated with data controller policies. After associating data subject with the tuples, the Data Product Manager sends the the sensor stream to the Policy Manager. The Policy Manager evaluates the data subject choices for this stream and decides whether if a data subject's data can be included in a data product. Policy Manager also stores all the policies and provide the interface for both data controller and data subjects to create, read, update, and delete their policies.

• *PET Manager* enables incorporation of diverse set of Privacy Enhancing technologies (PETs) to provide appropriate data security (RQ.3). Each PET is a stand alone module that is incorporated in the PET Manager. The PET Manager manages the data flow between a PET and PE-IoT. This modular approach not only enables seamless integration of new PETs but also allows the integration of remote/third party PETs into PE-IoT. The PET manager calls the PET module with the appropriate privacy parameters defined in the data product and gets privacy enhanced data in return. Another major role of the PET Manager is to exploit the classification of PETs i.e. stateful vs. stateless and blocking vs. non-blocking to efficiently manage the data flow during the privacy preserving data generation. For Negotiable PETs, PET manager facilitates negotiation of the privacy defined by privacy parameters.

• *Log Manager* maintains several types of logs generated by PE-IoT for auditing purposes (RQ.4). First, the Log Manager logs the enforcement process of data subject choice at the Data Product Worker Object and the data product access record generated by the data controller's access control service to demonstrate where the data subjects' data is going and which service providers have access to it. Also, the Log Manager logs all policy changes made at the Policy Manager by the data controller. Furthermore, the PET Manager uses the Log Manager to log the PET negotiation process to show the proof and reasons for the privacy parameter changes. In addition to store the logs, the Log Manager also provides interfaces for the authorized data subject and regulator to query the relevant logs for audition purposes.

• *Storage Manager* is an optional component that receives and stores all sensor data securely (RQ.3), handles historical queries, and supports data erasure based on data controller policies (RQ.5). Besides continuously accessing real-time data, certain service providers may also require sending a *one-shot query* to access historical data. For example, a service provider who studies space usage of a building to improve space management based on WiFi association data needs to query for historical data. A Storage Manager has a cryptographic key management for ensuring data security, query handler for answering historical queries, and a mechanism (e.g., Time-To-Live) for implementing data erasure.

Workflows in PE-IoT: PE-IoT system includes resource managers which should be coordinated using workflows to accomplish different goals for data controllers and service providers with data subject choices enforced.

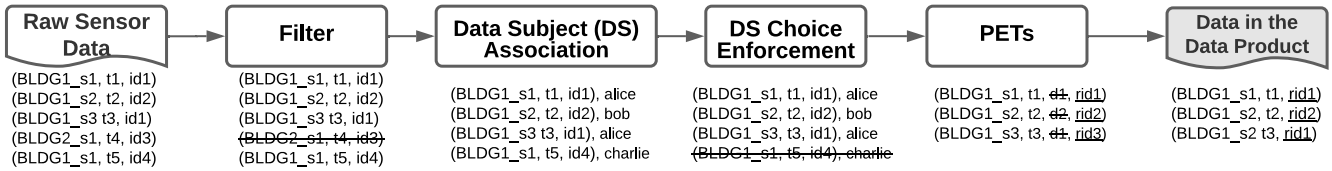

Figure 4: **The processing flow of producing privacy-preserving data flow for the data product. A data product processing example: Data Provider defines Data Product θ_1: Building 1's WiFi association data (Filter) with periodically-changing randomized ID (PET) and shares it for COVID-19 exposure tracking purpose (data controller sharing policy). A data subject, Charlie, who owns device id4 that connects to WiFi network, opt-out from sharing the data product for the given purpose**

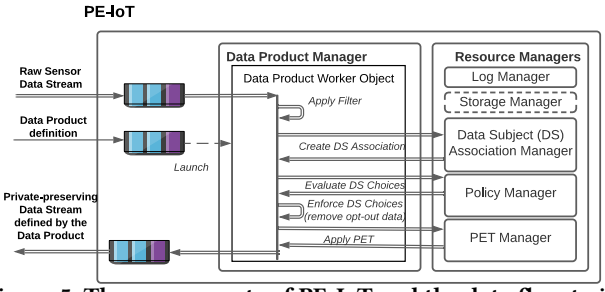

Figure 5: The components of PE-IoT and the data flow to instantiate a data product

• *Data Product Creation Workflow:* In this workflow (see Figure 5), we outline the coordination of different components by *Data Product Manager* to create a data product. A *Data Product Worker Object* is launched after a data product is defined and the raw sensor data is dispatched to the Data Product Worker Object. The first step after receiving raw sensor data is to the data product manager uses the *Filter* to remove the data that is not included in the Data Product to make sure the data is only processed for the given purpose (Data Product). In the second step, the Data Product Worker Object calls the *Data Subject Association Manager* to create the data subject association by adding the metadata about the associated user to each sensor data tuple. Then the Data Product Manager uses Policy Manager to check data subject choices and enforces the choices by removing the data that is associated with data subjects who want to opt-out of the data product. After enforcing subject policies, the *PET manager* facilitates the Data Product Manager to generate privacy preserving data using the Privacy Enhancing Technology defined in the data product. When storing sensor data is allowed by the data controller's retention policy and a *Storage Manager* is in place, the Data Product Manager launches a Worker Object to store the sensor data by coordinating with the Storage Manager.

•*Service Provider Access Management Workflow:* A service provider can either subscribe to a current data product to PE-IoT or query for a historical data product to the *Storage Manager* (data product generated based on past data). In both cases, the access is allowed or denied by the data controller's access control service by coordinating with the policy manager (as explained in Section 2.1). The query for historical data product includes the time range of past data in addition to the data product. If the request is allowed, the Storage Manager sends the data that satisfies the requested time range to PE-IoT. Upon receiving this data, PE-IoT performs the data product creation workflow and sends the generated data product to the Storage Manager which then forwards it to the service provider.

If a service provider has access to a negotiable data product, it can request for negotiation of the privacy level. The negotiation request is forwarded to the PET Manager and it decides whether to accept this request based on PET parameters that are part of the data product definition. The negotiation request contains appropriate negotiation parameters related to utility or privacy depending on the PET. This request is translated by the PET function into the privacy level that fulfils the negotiation request. The data product PET parameters contains minimum privacy level requirement defined by the data controller and the negotiation request is denied if the service provider requests for the data product with a privacy level lower than the minimum privacy level.

4 CHALLENGES AND OPEN QUESTIONS

We outline some of the challenges related to the design and implementation of PE-IoT in this section. Our policy model enables data subjects to opt-in/opt-out of data sharing based on a simple purpose-based access control. In future, we would like to extend this policy model for streaming data by utilizing existing works [15] which present a more complex policy model to allow the data subjects to express their policy based on any of the data tuple attributes and context variables such as time and space. Using a more complex policy model in IoT world, with large number of sensors, data subjects, and service providers, introduces scalability issues as the number of policies can be large.

Additionally, implementing one-to-many mapping between data record and data subjects (by extending Data Subject Association Manager with existing works such as [19]) can introduce more complexity in enforcing policies on data items, as now multiple data subject policies might apply to a single a data item and the resulting policy conflicts will need to be resolved. Scalable implementation of diverse Privacy Enhancing Technologies for stream processing (which requires batching, concurrency and parallelism) is another challenge. PE-IoT should be implemented to exploit the properties associated with a PET e.g. stateful vs. stateless, blocking vs non-blocking to improve the latency at large scale. The negotiable data products, although much needed, can bring further challenges e.g. proof of negotiation need by the service provider and building mechanisms for auditing and attestation of negotiation logs.

In this paper, we presented a design and architecture for deploying PE-IoT at organization level but scaling down it to a single device or scaling it up to city or state level is an open question. When deploying to a large-scale IoT system at the city or state level, scalability issues arise for the current central control model to handle large volumes of data and transmission delay. Different infrastructure to instrument privacy control could be explored, e.g., edge/fog computing[1]. Similarly, adapting PE-IoT to execute in a resource-limited scenario such as home or a single device such as smartphone which has limited computation power and smaller battery makes executing expensive PETs challenging.

REFERENCES

[1] Abduljaleel Al-Hasnawi and Leszek Lilien. 2017. Pushing Data Privacy Control to the Edge in IoT Using Policy Enforcement Fog Module. In *Companion Proceedings of The10th International Conference on Utility and Cloud Computing* (Austin, Texas, USA) *(UCC '17 Companion)*. Association for Computing Machinery, New York, NY, USA, 145–150. https://doi.org/10.1145/3147234.3148124

[2] Apple. 2020. Privacy-preserving contact tracing. https://www.apple.com/covid19/contacttracing

[3] Dave Archer, Michael A August, Georgios Bouloukakis, Christopher Davison, Mamadou H Diallo, Dhrubajyoti Ghosh, Christopher T Graves, Michael Hay, Xi He, Peeter Laud, Steve Lu, Ashwin Machanavajjhala, Sharad Mehrotra, Gerome Miklau, Alisa Pankova, Shantanu Sharma, Nalini Venkatasubramanian, Guoxi Wang, and Roberto Yus. 2020. Transitioning from testbeds to ships: an experience study in deploying the TIPPERS Internet of Things platform to the US Navy. *The Journal of Defense Modeling and Simulation* 0, 0 (2020), 1548512920956383. https://doi.org/10.1177/1548512920956383 arXiv:https://doi.org/10.1177/1548512920956383

[4] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. 2011. Private and Continual Release of Statistics. *ACM Trans. Inf. Syst. Secur.* 14, 3, Article 26 (Nov. 2011), 24 pages. https://doi.org/10.1145/2043621.2043626

[5] Yan Chen, Ashwin Machanavajjhala, Michael Hay, and Gerome Miklau. 2017. PeGaSus: Data-Adaptive Differentially Private Stream Processing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (Dallas, Texas, USA) *(CCS '17)*. Association for Computing Machinery, New York, NY, USA, 1375–1388. https://doi.org/10.1145/3133956.3134102

[6] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3–4 (Aug. 2014), 211–407. https://doi.org/10.1561/0400000042

[7] Chang Ge, Xi He, Ihab F. Ilyas, and Ashwin Machanavajjhala. 2019. APEx: Accuracy-Aware Differentially Private Data Exploration. In *Proceedings of the 2019 International Conference on Management of Data* (Amsterdam, Netherlands) *(SIGMOD '19)*. Association for Computing Machinery, New York, NY, USA, 177–194. https://doi.org/10.1145/3299869.3300092

[8] Sameera Ghayyur, Yan Chen, Roberto Yus, Ashwin Machanavajjhala, Michael Hay, Gerome Miklau, and Sharad Mehrotra. 2018. IoT-Detective: Analyzing IoT Data Under Differential Privacy. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 1725–1728. https://doi.org/10.1145/3183713.3193571

[9] Sameera Ghayyur, Dhrubajyoti Ghosh, Xi He, and Sharad Mehrotra. 2019. Towards Accuracy Aware Minimally Invasive Monitoring (MiM). In *TPDP 2019 - Theory and Practice of Differential Privacy, A workshop in ACM Conference on Computer and Communications Security (CCS) 2019*. TPDP, London, UK, 1–4.

[10] Drew Harwell. 2019. Colleges are turning students' phones into surveillance machines, tracking the locations of hundreds of thousands. https://www.washingtonpost.com/technology/2019/12/24/colleges-are-turning-students-phones-into-surveillance-machines-tracking-locations-hundreds-thousands/

[11] Wilhelm Kleiminger, Christian Beckel, Thorsten Staake, and Silvia Santini. 2013. Occupancy Detection from Electricity Consumption Data. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings* (Roma, Italy) *(BuildSys'13)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/2528282.2528295

[12] Phillip Lee, Eun-Jeong Shin, Valerie Guralnik, Sharad Mehrotra, Nalini Venkatasubramanian, and Kevin T. Smith. 2019. Exploring Privacy Breaches and Mitigation Strategies of Occupancy Sensors in Smart Buildings. In *Proceedings of the 1st ACM International Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities, TESCA 2019*, P. Venkat Rangan, Nalini Venkatasubramanian, Maneesha Vinodini Ramesh, and Serge Miranda (Eds.). ACM, New York, NY, USA, 18–21. https://doi.org/10.1145/3364544.3364827

[13] F. Li, J. Sun, S. Papadimitriou, G. A. Mihaila, and I. Stanoi. 2007. Hiding in the Crowd: Privacy Preservation on Evolving Streams through Correlation Tracking. In *2007 IEEE 23rd International Conference on Data Engineering*. IEEE Computer Society, Istanbul, Turkey, 686–695. https://doi.org/10.1109/ICDE.2007.367914

[14] P. Pappachan, M. Degeling, R. Yus, A. Das, S. Bhagavatula, W. Melicher, P. E. Naeini, S. Zhang, L. Bauer, A. Kobsa, S. Mehrotra, N. Sadeh, and N. Venkatasubramanian. 2017. Towards Privacy-Aware Smart Buildings: Capturing, Communicating, and Enforcing Privacy Policies and Preferences. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*. IEEE Computer Society, Atlanta, GA, USA, 193–198. https://doi.org/10.1109/ICDCSW.2017.52

[15] Primal Pappachan, Roberto Yus, Sharad Mehrotra, and Johann-Christoph Freytag. 2020. Sieve: A Middleware Approach to Scalable Access Control for Database Management Systems. *Proc. VLDB Endow.* 13, 11 (2020), 2424–2437. http://www.vldb.org/pvldb/vol13/p2424-pappachan.pdf

[16] Do Le Quoc, Martin Beck, Pramod Bhatotia, Ruichuan Chen, Christof Fetzer, and Thorsten Strufe. 2017. PrivApprox: Privacy-Preserving Stream Analytics. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*. USENIX Association, Santa Clara, CA, 659–672. https://www.usenix.org/conference/atc17/technical-sessions/presentation/quoc

[17] Supreeth Shastri, Melissa Wasserman, and Vijay Chidambaram. 2019. The Seven Sins of Personal-Data Processing Systems under GDPR. In *Proceedings of the 11th USENIX Conference on Hot Topics in Cloud Computing* (Renton, WA, USA) *(HotCloud'19)*. USENIX Association, USA, 1.

[18] European Union. 2020. General Data Protection Regulation GPDR. https://gdpr.eu/.

[19] Liyan Zhang, Dmitri V Kalashnikov, Sharad Mehrotra, and Ronen Vaisenberg. 2014. Context-based person identification framework for smart video surveillance. *Machine Vision and Applications* 25, 7 (2014), 1711–1725.