2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)

On the Case of Privacy in the IoT Ecosystem: A Survey

Sana Imtiaz*†, Ramin Sadre*, and Vladimir Vlassov†

*Université catholique de Louvain, ICTEAM/INGI, Louvain-la-Neuve, Belgium
Email: {sana.imtiaz,ramin.sadre}@uclouvain.be
† KTH Royal Institute of Technology, SCS/EECS, Stockholm, Sweden
Email: {sanaim,vladv}@kth.se

***Abstract*—IoT has enabled the creation of a multitude of personal applications and services for a better understanding of urban environments and our personal lives. These services are driven by the continuous collection and analysis of user data in order to provide personalized experiences. However, there is a strong need to address user privacy concerns as most of the collected data is of sensitive nature. This paper provides an overview of privacy preservation techniques and solutions proposed so far in literature along with the IoT levels at which privacy is addressed by each solution as well as their robustness to privacy breaching attacks. An analysis of functional and non-functional limitations of each solution is done, followed by a short survey of machine learning applications designed with these solutions. We identify open issues in the privacy preserving solutions when used in IoT environments. Moreover, we note that most of the privacy preservation solutions need to be adapted in the light of GDPR to accommodate the right to privacy of the users.**

***Keywords*-IoT; wearables; privacy; privacy-aware machine learning; recommender systems;**

I. Introduction

With the increasing popularity of the Internet of Things (IoT), the past decade has seen the appearance of a plethora of smart devices. It has been predicted that there will be more than 4 devices for every human on Earth by 2020 [1]. Broadly speaking, any sensor that is capable of collecting data, processing it using built-in circuitry and transmitting it qualifies as a smart device. Typically, these devices upload the data to the cloud where it is further processed and stored in order to offer personalized services to the end users. An advanced variant of this approach includes offloading further processing and analytic capabilities to the devices, commonly referred to as *edge computing*.

Several models for the architecture of the IoT have been proposed in literature. Figure 1 shows a widely accepted general model with three layers [2]:

- L1: *perception layer* – consisting of the sensory devices collecting data.

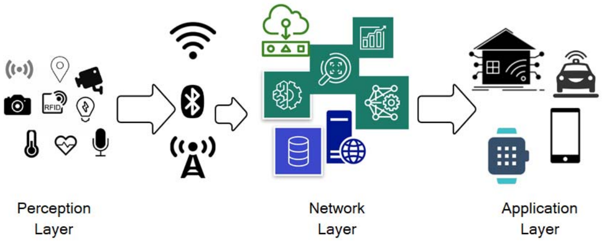

Figure 1. IoT Architecture Layers

- L2: *network layer* – responsible for collecting, aggregating, processing, and transmitting the data from the perception layer.
- L3: *application layer* – consisting of all the applications and solutions driven by the sensory data that are available to the users.

Based on this three-layer model, Chen [3] proposed that the IoT ecosystem is composed of four major components: *sensors* (in L1), *communication* (in L2), *computation* (in L2) and *service* (in L3). We will use both models interchangeably in this work.

For example, a *wearable sensor*, such as a fitness tracker, is part of L1. The network infrastructure as well as the supporting technologies that store, aggregate and process this data (commonly in the cloud) are part of L2. Finally, users interact with fitness applications using their smart phones in L3.

Obviously, privacy is a major concern in IoT. In our above example, the fitness tracker collects information about the user's location with respective timestamp, heart rate, daily activities, etc. This data is then collectively analyzed by recommender systems to give users personalized health advices. In many cases, such recommender systems are driven by models created by machine learning (ML) algorithms. Unfortunately, these models are often sensitive towards specific training samples: Due to the nature of the datasets and the uniqueness of the data points, some of the training samples are implicitly memorized [4], [5]. Research has shown that it is possible for attackers to replicate or

This work is funded by the Erasmus Mundus Joint Doctorate program in Distributed Computing (EACEA of the European Commission under FPA 2012-0030).

978-1-7281-2980-8/19/$31.00 ©2019 IEEE
DOI 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00177

recover the details of the recommender's underlying model, referred to as *model stealing* [6]–[8]. Moreover, private and sensitive training data can be recovered from the models by performing *model inversion* attacks [5], [9], [10].

When it comes to IoT devices and solutions available commercially, privacy is often confused with security, and secure solutions are often marketed as privacy preserving. Moreover, existing solutions and techniques mainly focus on securing the communication channel as well as authentication and authorization mechanisms. Much less consideration is given to the preservation of privacy in the data collection, aggregation, storage and retrieval processes [11]. There is an imminent need to introduce privacy in all components, which requires better understanding of privacy threats in the IoT ecosystem. Furthermore, it is important to analyze the impact of privacy preservation techniques on data analysis and the quality of service in terms of trade-offs between accuracy, privacy and efficiency.

This paper presents an overview of privacy preserving techniques for IoT along with the privacy threats addressed by each solution, their limitations, and known resistance to attacks on user privacy. For this work, we focus on IoT devices and services used for personal applications such as health care and smart home solutions. Moreover, we focus on the three components *sensors*, *computation* and *service* since, since they have received much less attention so far than the *communication* component, as mentioned above. Consequently, we consider privacy in communication protocols as outside the scope of this paper.

Contributions: Our main contributions can be summarized as follows:

- We propose and present a taxonomy of privacy preserving techniques and solutions for the IoT ecosystem;
- We provide a comparison of privacy preserving techniques and solutions that we have observed in this work;
- We analyze the techniques in the light of the EU's General Data Protection Regulation (GDPR);
- We identify some open issues in privacy preserving techniques that should be addressed.

Organization: This paper is structured as follows. Section II presents common privacy threats and attacks in the IoT ecosystem. A taxonomy of privacy preserving techniques is presented in Section III along with the limitations of each solution designed with these techniques, and their merits and demerits. Afterwards, we briefly comment on privacy-aware ML and data mining solutions in Section IV. An analysis of privacy preserving techniques in the light of GDPR is presented in Section V. Finally we list some open issues and suggestions for future work, and conclude in Sections VI and VII respectively.

II. Privacy threats and attacks in IoT ecosystem

According to [12], privacy is "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others". A definition of privacy concerns is proposed by Smith et al. [13]: concerns for collection of personal information, concerns for unauthorized secondary use (internally in organizations and externally), concerns for improper (unauthorized) access to personal data and concerns for errors in collected personal information.

In order to categorize privacy preservation solutions, we first identify privacy threats in the IoT ecosystem and the architecture layers associated with them. Afterwards, we provide an overview of attacks on privacy and the respective threats associated with them.

A. Privacy threats

Ziegeldorf et al. [14] categorize the most common privacy threats in IoT. In the following, we give a short overview of the threats and we attribute them to the different layers of the IoT architecture. Note that these threats often occur in combination in IoT solutions, depending on the type of service offered.

1) Identification: Denotes the threat of associating a persistent identifier with an individual or their data. For example, a name, pseudonym, an image or voice, or an address can be associated with an individual from a database or collection. It is classified as the most common threat.
Affects: information processing in the network layer (L2).

2) Localization and tracking: With a notion of identification, this denotes the threat of determining an individual's physical location and recording it over time without authorization or consent. Location based services (LBS) commonly suffer from this threat as they can enable GPS stalking [15], though internet traffic can also be exploited for this purpose. Moreover, IoT-based LBS in indoor environments require additional constraints on data sharing and authorization, e.g., they can enable stalking and unintended bias in work environments.
Affects: all layers of IoT architecture, though it is more visible on the network (L2) and application (L3) layers.

3) Profiling: Users are profiled for the sake of personalization but this often results in unwanted advertisements, price discrimination or biased automatic decisions. In an IoT-based environment, this threat is more imminent due to the availability of multiple information sources which potentially allow compiling complete information dossiers about individuals and inferring user preferences by correlation with other profiles.
Affects: information processing in the network layer (L2), especially in scenarios that require data dissemination or sharing with third parties.

4) Interaction and presentation: Similar to shoulder surfing, this refers to the threat of violating user privacy by conveying some private information intended for a specific user over a public medium. For example, one may get recommendations through the speaker or screen of a smart thing and people in the vicinity can also observe that information.
Affects: application layer (L3). Also occurs on the perception layer if the solutions offered are presented using peripherals of the sensory devices (e.g., speaker or screen).

5) Lifecycle transitions: This threat occurs upon change of ownership of the IoT devices. Most IoT devices are sold with the assumption of buy-once-use-forever, and log huge amounts of personal data throughout their lifetime history. This data (and its impact on personalization offered by the IoT device) may not be completely removed upon a memory wipe before transfer of device ownership.
Affects: information collection in the network layer (L2).

6) Inventory attacks: This threat mainly occurs due to communication capability of the sensor devices, which enables unauthorized access or collection of data. Unauthorized parties can also observe the communication pattern (and other distinguishable properties) and deduce the presence of devices as well as their type and model. Moreover, inventories can give information on user preferences which may be exploited by law enforcement agencies to conduct unwarranted searches or by burglars for targeted break-ins.
Affects: information collection in the network layer (L2). May be enabled using application layer (L3) by exploiting security flaws in the application (L3) or through perception layer (L1).

7) Linkage: Users consent to sharing different attributes of personal data with each IoT service they use. However, the combination of data collected from independent sources can reveal information about individuals that they originally did not consent to reveal [16]. Moreover, data maybe be incorrectly inferred due to loss of context as a result of the combination of different permissions (non-uniform data access restrictions).
Affects: Information dissemination in the network layer (L2).

B. Attacks on user privacy

Here, we briefly describe some of the common attacks on user privacy. Note that it is not an exhaustive list of possible privacy attacks. Descriptions of more attacks targeting IoT ecosystem components (e.g. databases and ML models) that in turn compromise user privacy, can be found in [17], [18] and other works.

1) Membership inference attack: With this attack, the adversary can reveal whether or not a specific data record was used to train the ML model, given that the adversary has knowledge of the ML model and the individual data record [19], [20]. Privacy is violated in this attack if inclusion of an individual in a training set is itself sensitive. For example, inclusion as a data record in a health-related ML model leaks information about the health of that individual. In terms of privacy threats, this attack directly threatens the identification of a person, can aid in profiling, and can make use of linkage and inventory attacks.

2) Data inference attack: As observed by [21], this attack is commonly associated with encryption-based privacy preserving solutions. It tries to recover some information about a given data record by using Linkage (with publicly known information) and making tailored queries to the system and observing the responses to see if any information about underlying records is leaked. A classical example of this attack is using frequency analysis to break ciphers.

3) Attribute disclosure: Attribute disclosure occurs when some released data records make it possible to infer characteristics of an individual more accurately than is generally known about that individual [22]. In other words, new information about some individuals is revealed by the data release. This attack commonly uses linkage from multiple data sources to infer user information.

4) Fingerprinting and Impersonation attacks: Using Inventory attacks, an adversary might observe the communication pattern of a device and try to mimic it [23], [24]. If the privacy is compromised, the adversary might be able to access credentials of the device to alter privacy preferences of the user and inject fake data into the system.

5) Re-identification attacks: In this attack, an adversary can use linkage to combine data from multiple collections to re-identify a record from outsourced, published or open data records [25]. Re-identification is a very commonly observed attack, with the classic example of a voter list used for re-identification of a government official's health record from the records released by a health insurance company in 1997 [26].

6) Database reconstruction attacks: As observed in [27], confidential data may be vulnerable to database reconstruction attacks when statistical data is published by agencies for research or information purposes. This enables partial or full reconstruction of the original database records, which may lead to identification or unintended profiling of some users based on their association with some attributes in the targeted database.

7) Model stealing: Similar to database reconstruction, it is also possible to reconstruct or reveal the internal training parameters and other sensitive details of ML or recommender models using model stealing techniques [6]–[8], [28]. This reveals sensitive information about the training data used for these models and can result in unintended profiling of users.

8) Model inversion: By observing ML model predictions, model inversion attacks enable adversaries to extract underlying training data of the individuals, as observed in [5], [9], [10]. A specific training record may not always be extracted as a result of this attack. Instead, the adversary will extract an average representation of inputs that are similarly classified. However, this can be a huge privacy threat if the exposed classes are sparsely populated, i.e., a class may correspond to a single individual in the records [9].

III. Taxonomy of Privacy preserving techniques

We now present a taxonomy of privacy preserving techniques that eliminate the risk of privacy threats (presented in Section II-A) and prevent the attacks on user privacy (described in Section II-B).

Terminologies: Techniques represent the general principle(s) and methodology employed for privacy preservation, e.g., anonymization. *Solutions* represent algorithms designed using these principles. *Functional limitations* refer to design limitations on where the solutions can be applied depending on the data or the nature of the algorithm. *Non-functional limitations* include issues such as performance, scalability and accuracy.

A. Anonymization techniques

The industry and health care sectors have been employing data de-identification for years as a privacy preserving measure [29]–[31]. Common practice includes removal of some sensitive attributes like names, gender, state codes, or identification numbers – commonly referred to as *personally identifiable information (PII)*. Moreover, more sophisticated methods such as *k*-anonymization [32], [33] and *l*-diversity [34] and *t*-closeness [22], are employed for better privacy preservation guarantees.

1) k-anonymity: *k*-anonymity provides privacy protection by guaranteeing anonymity between *k* entries – each released data record will relate to at least *k* individuals in the collection even if the records are directly linked to external information [32], [33]. It uses generalization (replacing or recoding a value with less specific but semantically consistent value) and suppression of records (not releasing a value at all) to achieve privacy goals. However, these might skew the characteristics of the original dataset. Functional limitations include data diversification to ensure there are at least *k* similar records in the database. *k*-anonymity has been shown to perform well for location based services (LBS) to prevent fake data injection attacks [35] and for privacy-preserving publishing of Electronic Health Records (EHR) [36]. However, it has been shown that *k*-anonymity is susceptible to data inference attacks [37], as well as attribute disclosure [22], re-identification attacks and database reconstruction attacks [38]. Improved versions such as (α, *k*)-anonymity model [38], have been proposed in literature to mitigate re-identification and database reconstruction attacks.

2) l-diversity: Improves upon *k*-anonymity and provides protection against attribute inference attacks [34]. Each anonymized group of (generally *k*) users has at least *l* "well represented" sensitive attribute (SA) values. Another improved version requires to have at least *l* distinct SA values in each group, called *distinct l-diversity* [22]. Similar to *k*-anonymity, functional limitations include diversification in the dataset, as we need to ensure presence of well distributed SA values. However, in some cases, attackers are still able to associate an individual's record to have a certain SA when that value appears more frequently than other values in the group [36].

3) t-closeness: This solution improves upon its precedents and aims at limiting the distance between the probability distributions of SA values within an anonymized group and SA values in the entire dataset [22]. This method provides better privacy guarantees against attribute disclosure as the attacker can not learn any information about an individual's SA value other than what is already available from the entire dataset. Some practical implementations have found *t*-closeness to be resistant to attribute disclosure attacks, however, its resistance to membership inference attacks still needs to be investigated [39].
Researchers have also combined these solutions for better privacy guarantees. For example, Yin et al. [40] propose using *k*-anonymity and *l*-diversity in combination to prevent imbalanced sensitive attribute distribution in datasets to prevent attribute disclosure attacks. Moreover, there are many versions of all the aforementioned techniques proposed in literature, each focusing on protecting against a specific type of attack depending on the use case.

B. Model or output obfuscation techniques

User re-identification by model inversion attacks can be prevented by obfuscating the output of ML models within a provided range. Differential privacy is a solution that aims to maximize the accuracy of queries from statistical databases while minimizing the chances of identifying its records [41]. A function ensuring ε-differential privacy adds appropriately chosen random noise (with Laplacian distribution) to the true answer of an ML model to produce the response. This implies a fixed uncertainty in all measurements, implicating less probability of exposing a specific record. However, differential privacy alone cannot provide privacy guarantees for all scenarios due to certain functional limitations: a) it is designed for low sensitivity data queries, and b) using statistical inference and adaptive querying, one can infer the form of the underlying data distribution.

Differential privacy can be regarded as the most widely researched and adopted solution for privacy preservation in the current era. It is highly effective against model inversion and inference attacks, and is being used heavily in combination with other techniques to develop privacy preserving applications and services [4], [19], [42], [43].

C. Multi-tier Machine Learning as a privacy preservation mechanism

Training openly available ML models on sensitive user data directly allows for data memorization. This technique proposes introduction of multiple training levels, which can reduce the footprint of distinct and sensitive training data on output models. Semi-supervised knowledge aggregation and transfer [4] is a multi-tier ML solution that proposes a 'teacher' and 'student' models hierarchy. Teacher models train directly on partitions of sensitive data, and afterwards apply an aggregation mechanism as privacy preserving layer to train a student (openly available) model on non-sensitive data using the teacher models. This technique uses differential privacy to define privacy-preserving properties of student models during the training phase. As only the student models are published, using model inversion attacks cannot compromise the original training samples given the fact that noisy voting is used in the training procedure instead of considering the absolute majority of classification decisions of teacher models. This is a relatively new distributed solution with strong privacy guarantees and applicable to a wide range of ML models. However, its utility with respect to quality of recommendations needs to be researched.

D. Decentralized machine learning

Decentralized machine learning solutions offer a new computing paradigm for better privacy preservation. Instead of transmitting (potentially sensitive) user data to computation, a part of the computation is offloaded to end-user devices and each device contributes partial updates to the system model. Doing so eliminates the risk of exposing sensitive and private raw data to the service provider as well as other *honest-but-curious* adversaries present in the environment. Federated machine learning [44], [45] has become an increasingly popular solution based on this technique in the past few years and is increasingly being researched and used in ML models and recommender systems [46]–[49]. It proposes creation of a global model as a result of learning attributes from updates pushed by users. Since it is a relatively new technique, it caters well to the nature of distributed computing systems used in the IoT ecosystem: It is highly scalable and efficient. However, there is a need to investigate how different applications and use cases can be ported to this solution. Federated machine learning can, however, be susceptible to inference attacks [50] as it exposes intermediate results which may actually leak important information about user data [47].

E. Cryptography-based solutions

It is believed that if data is encrypted during analysis, user privacy can not be compromised. Homomorphic Encryption [54] is a cryptographic solution that allows computation to be executed directly on encrypted data. It supports addition, multiplication, and quadratic functions. Moreover, homomorphic encryption offers privacy-preserving capabilities in both training and classification phases of ML models, unlike most of the existing works that only focus on the training phase. Homomorphic schemes are further classified as fully or partially homomorphic.

1) Partially Homomorphic schemes: These support limited operations like addition and multiplication as well as other operations on ciphertexts, but do not support arbitrary computation on ciphertexts. These schemes perform relatively well in practice and have better performance due to lower computational complexity as compared to fully homomorphic schemes. However, fewer algorithms can be implemented using the restricted set of operations [55].

2) Fully Homomorphic Encryption: Fully Homomorphic Encryption (FHE) schemes not only support multiplication and addition, but also support quadratic function and arbitrary computation on ciphertexts. Classifiers designed using this schema are privacy preserving by nature and are better suited for real world applications in terms of privacy guarantees, because they support arbitrary computation. However, few fully homomorphic encryption schemes exist, and they are often computationally expensive, i.e., around 2-5 seconds per operation [55]. Some efficient FHE schemes have been proposed [56], but they have been found susceptible to data inference attacks like encryption key recovery and data decryption in both known message (broadcast) and unknown message (secret) scenarios [51].

Other popular solutions include garbled circuits and Secure Multi-Party (SMP) computation protocols. Originally proposed by Yao in the 1980s [57] as a secure way of computation, garbled circuits are now used extensively for providing privacy guarantees. Similarly, SMP solutions are also being investigated for providing privacy guarantees in information processing.

F. Data summarization techniques

Exposing raw user data not only poses communication overhead but also puts user privacy at risk. This technique proposes creation of aggregated and summarized versions of datasets for efficient creation of ML models as well as providing user privacy. This poses the trade-off between accuracy of data and privacy preservation. The data summarization solution proposed by [58] uses this technique for improving performance and potentially enhancing privacy preservation in the recommender systems. It marks portions of the data as private and summarizes the rest of the data from all users to create a representative training dataset.

When it comes to non-functional limitations, similar to decentralized ML, it is a relatively new technique and is scalable and efficient to cater to the nature of distributed platforms and systems used by IoT services. However, privacy guarantees using this solution need to be investigated.

G. Ensuring privacy with dataflow models

This technique proposes creation of data flow models with respective permissions at each level to ensure user privacy and transparent accountability.

Table I
ANALYSIS OF PRIVACY PRESERVING SOLUTIONS

Privacy preserving technique	Solution	Merits	Affected IoT layers	Relevant privacy threat(s)	Limitations	Trade-offs	Relevant attacks	Known resistance
Anonymization	*k*-anonymity	Easy to implement, low complexity	L2 (information aggregation)	Identification, localization and tracking, profiling, linkage	Requires diverse data	Accuracy/ privacy	Re-identification, Database reconstruction, Data inference, Attribute disclosure	Weak resistance (in some implementations) [22], [37]
	l-diversity	Low complexity	L2 (information processing)	Same as above	Requires diverse data	Accuracy/ privacy	Attribute disclosure	Mediocre resistance [22], [36]
	t-closeness	Protects sensitive attributes	L2 (information processing)	Same as above	Requires strong dataset diversification	Accuracy/ privacy	Attribute disclosure	Strong resistance [39]
Model or output obfuscation	Differential privacy	Easy to integrate with solutions	L2, L3	Identification, profiling, linkage	Works for low sensitivity data queries	Accuracy/ privacy	Model Inversion, Inference attacks	Strong resistance [19], [42], [43]
Multi-tier ML	Semi-supervised knowledge transfer	Distributed, applicable to any ML model	L2, L3	Profiling, linkage	Affect on accuracy of ML models is unknown	Accuracy/ privacy	Model stealing and inversion, Inference attacks	Strong resistance [4]
Decentralized ML	Federated ML	Highly scalable and efficient	L2, L3	Inventory attacks, linkage, profiling	Potential information leakage	Efficiency/ privacy	Inference, Fingerprinting and impersonation attacks	Mediocre resistance [50]
Cryptography	Fully Homomorphic encryption	Private ML models training and classification	L2	Inventory attacks	Large computational overhead	Efficiency/ privacy	Data inference (data/key recovery)	Strong/ mediocre resistance [51]
	Partially Homomorphic encryption	Relatively lower computational overhead	L2	Inventory attacks	Not applicable to all ML models	Accuracy/ privacy	Inference attacks	Mediocre resistance
Data summarization	Public-private data summarization	Highly efficient solution with minimal loss of accuracy	L2	Identification	Unquantified Privacy guarantees	Accuracy/ privacy	Inference attacks	Unknown
Data flow models	blockchain for privacy	Verifiable privacy	L2	Inventory attacks	Computational overhead, low scalability	Efficiency/ privacy	Fingerprinting and impersonation attacks	Strong resistance [52]
	privacy-preserving programming languages and platforms	Low overhead with verifiable privacy	L2, L3	Inventory attacks	Information flows to be declared beforehand	Efficiency/ privacy (in some cases)	Fingerprinting and impersonation attacks (some cases)	Strong resistance [23]
Personalized data stores	HAT	User controls and monetizes her data	L1, L2	Linkage (under consent)	Requires users to pay for storage	Cost/ privacy	Attribute disclosure	Unknown
Private Compute Units	Intel® SGX processors	Protected execution environment; no data or computation is exposed	L2 (information processing, computation)	Inventory attacks	Requires specific application design for SGX programming model	Efficiency/ privacy	Data inference (using side-channel information and cache-timing [53])	Strong/ mediocre resistance [53]

1) Blockchain to ensure privacy and verifiability: Researchers have proposed the use of blockchain for verifiability and accountability of data collection, storage and access in IoT environments [52], [59]–[61]. For example, blockchain-based data provenance can provide tamper-proof records and enable data accountability in the cloud [52]. Moreover, blockchains are being extended for use in the context of IoT for healthcare, as surveyed in [62]. However, there is room for research for introducing scalability in blockchains so they can adapt well to IoT environments.

2) Privacy-centric programming languages and platforms: These solutions require information flows and privileges to be declared beforehand, so all the data elements are attached to respective policies [23], [63]–[65]. For example, Jeeves [65] is a privacy centric programming language, used as an add-on library with Java. HomePad [63] applications are implemented as directed graphs of elements (instances of functions that process data in isolation). It allows for automatic verification of the application's flow graph against user-defined privacy policies with low computational overhead by modeling these elements and the information flow graph. In addition to that, [66] outlines some guidelines for privacy preservation while designing IoT applications.

H. Personalized data stores

Personalized data stores offer a flipped environment for privacy preservation, where the users collect and maintain their data from multiple sources in one place, e.g. an encrypted data store, and authorize its informed use. The Hub-of-All-Things project (HAT) is a solution that proposes total control of data by the user and monetizing this data [67], [68]. Instead of storing data on different platforms, it is aggregated in the data store and users can offer their data to interested parties in exchange for personalized services.

I. Privacy preservation at processing level

This technique proposes secure and private compute/processing units to ensure that no data or computation is exposed in the entire information flow. Intel® introduced Software Guard Extensions (SGX) [69] as a solution that proposes the use of "enclaves", protected areas of execution, to protect selected code and data from disclosure or modification. A huge merit of this solution is that it is a hardware-assisted execution environment with the smallest possible attack surface: the CPU boundary. It also provides specific architecture instructions to mark portions of data and code as private, which makes it similar to sandbox concepts in the security domain. In principle, it is a privacy-preservation solution for both users and corporations – users may execute analytic codes locally without moving their data anywhere, and corporations can analyze data on user-end without exposure of their algorithms. However, recent research has shown that it is susceptible to some data inference attacks using side-channel information like cache-timing [53] when working with weaker versions of encryption algorithms. It also requires designing application complying with a specific programming model, which may be inefficient for adapting private implementations of algorithms currently in use by large organizations.

Table I summarizes the results of our analysis and classification of privacy preservation techniques and solutions, affected IoT layers and their known resistance towards attacks. For each of the privacy preservation solutions in the table, we indicate a level of privacy, namely strong/mediocre/weak resistance, based on the assessments provided in the literature (studied papers).

IV. Privacy-aware ML and Data Mining

A number of privacy preserving implementations of machine learning and data mining algorithms can be found in literature. Papernot et al. [70] survey the state of the art of privacy preserving ML algorithms. Moreover, differential privacy is used extensively in ML models for protection against model inversion attacks [4], [71]–[73].

Chiron [74] is an interesting implementation of privacy-preserving ML-as-a-service, designed particularly for cloud environments which form a major part of the IoT ecosystem. It uses private compute units (with SGX) to enhance privacy guarantees. Moreover, implementations of *k-anonymity* in combination with ML algorithms and cryptography techniques with ML [75] also exist in literature.

When it comes to data mining, as mentioned in Section III-D, various implementations of recommender systems use federated learning as a privacy preservation measure [46]–[49]. Collaborative filtering is used extensively in recommender systems [76]. Some privacy-preserving implementations include [77], which combines *k*-anonymity with collaborative filtering; [78], which applies obfuscation; and [79], which uses differential privacy in combination with homomorphic encryption to ensure private recommendations. Also, [48] proposes a federated ML version of collaborative filtering for personalized recommendations.

V. GDPR and its implications

The GDPR enforces all organizations that collect and process data from users to include Privacy by Design and Privacy by Default (originally explained in [80]). Privacy by Design dictates that organizations should design all their services involving processing of personal data while considering data protection and privacy measures at every step. Privacy by Default dictates that all public services should apply the strictest privacy settings by default, without any manual input from the end-user. The GDPR also grants some basic rights to end-users: right to (give and withdraw) consent, right to be forgotten and right to access (personal) information [81]. Veale et al. [82] analyze the impact of incorporating the GPDR law in ML models for protection against model inversion and membership inference attacks. They conclude that some ML models may need to be legally classified as personal data as a result of this law.

Relatively new privacy preserving techniques proposed in literature are GDPR-compliant by design. For example, personalized data stores are directly based on the principles of user consent and the right to access. The right to be forgotten can also be exercised by removing the data access from organizations that fail to comply with the user's privacy preferences. Also, for private compute units, since user data can potentially always stay on the device, the right to access data is respected. However, organizations need informed consent of the users for analyzing their data. Similarly, data flow models (solutions using blockchain and pre-defined information flows) are also GDPR-compliant by design. Moreover, for these solutions, the user defines privacy preferences and is able to verify if they are respected by the service. For solutions based on data summarization, users may not be able to exercise their right to access information, as the information is used in a modified (summarized) form. Moreover, cryptography-based techniques may also hinder the right to access collected information although they may ensure privacy by design and by default. Other techniques based on multi-tier and decentralized ML might also not be able to comply with the right to access information as it might give out sensitive details about how the organizations are training their ML and recommender models. We believe that, in principle, it is hard to enforce the right to forget in ML algorithms once user data has already been used to train an ML model (though the effects of data point on the trained model might disappear eventually), which in turn implies that they should be classified as personal data as proposed by [82].

VI. Open issues and future work

In the light of our analysis of privacy preserving techniques and the discussion on GDPR presented above, we identify some open issues and suggestions for future work. First, it is advised to use synthetic or representative datasets for where exact computations are not needed [83]. Moreover, there is a need to find an optimal trade-off between data utility and privacy preservation when generating the representative datasets. Solutions for data summarization should be combined with other privacy preserving techniques for better privacy guarantees. However, the effect of combining different techniques on accuracy and efficiency of solutions needs to be investigated. Also, there is a strong need to formulate guidelines for publishing privacy preserving open datasets, ML and recommender models. Additionally, blockchains-based solutions might be good candidates for verifiable privacy preservation on the user-end. In general, there is no clear winner among the privacy preservation techniques – depending on the use case, some techniques will outperform others in terms of robustness towards attacks. Another interesting observation is that industry and healthcare organizations have often found the relatively weaker solutions to be strong candidates for privacy preservation.

VII. Conclusion

In this paper, we have identified privacy threats on different layers of the IoT ecosystem as well as associated attacks on user privacy. We presented a taxonomy of state of the art privacy preservation techniques along with their limitations, susceptibility to privacy threats and their proved robustness towards attacks on privacy. Depending on the use case, model obfuscation techniques, multi-tier and decentralized ML, private compute units, and data flow models (using blockchains and pre-defined information flows) emerge as relatively stronger techniques for privacy preservation. However, not all of the solutions based on these techniques can guarantee the rights granted by the GDPR to users. We also highlight some open issues and directions for future work. In summary, the solutions proposed in the recent years are trying to incorporate the GDPR, which will ensure better privacy guarantees for users.

References

[1] M. Hung, "Leading the IoT," *Gartner Insights on How to Lead in a Connected World, 2017. [Online]: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf*, Accessed March 2019.

[2] K. Zhao and L. Ge, "A survey on the internet of things security," in *9th International Conference on Computational Intelligence and Security*, Dec 2013, pp. 663–667.

[3] Y.-K. Chen, "Challenges and opportunities of internet of things," in *17th Asia and South Pacific design automation conference*. IEEE, 2012, pp. 383–388.

[4] N. Papernot, M. Abadi, Ú. Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint arXiv:1610.05755*, 2016.

[5] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *arXiv preprint arXiv:1802.08232*, 2018.

[6] F. Tramèr *et al.*, "Stealing machine learning models via prediction apis," in *25th $USENIX$ Security Symposium*, 2016, pp. 601–618.

[7] B. Wang and N. Z. Gong, "Stealing hyperparameters in machine learning," in *Symposium on Security and Privacy (IEEE S&P)*. IEEE, 2018, pp. 36–52.

[8] M. Juuti, S. Szyller, A. Dmitrenko, S. Marchal, and N. Asokan, "Prada: protecting against DNN model stealing attacks," *arXiv preprint arXiv:1805.02628*, 2018.

[9] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the CCS*. ACM, 2015, pp. 1322–1333.

[10] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "Towards the science of security and privacy in machine learning," *arXiv preprint arXiv:1611.03814*, 2016.

[11] P. P. Jayaraman *et al.*, "Privacy preserving internet of things: From privacy techniques to a blueprint architecture and efficient implementation," *Future Generation Computer Systems*, vol. 76, pp. 540–549, 2017.

[12] A. Westin, *Privacy and freedom*. Atheneum New York, 1967, vol. 1.

[13] H. J. Smith, S. J. Milberg, and S. J. Burke, "Information privacy: measuring individuals' concerns about organizational practices," *MIS quarterly*, pp. 167–196, 1996.

[14] J. H. Ziegeldorf *et al.*, "Privacy in the internet of things: threats and challenges," *Security and Communication Networks*, vol. 7, no. 12, pp. 2728–2742, 2014.

[15] J. Voelcker, "Stalked by satellite-an alarming rise in gps-enabled harassment," *IEEE Spectrum*, vol. 43, no. 7, pp. 15–16, 2006.

[16] N. Madaan, M. A. Ahad, and S. M. Sastry, "Data integration in IoT ecosystem: Information linkage as a privacy threat," *Computer law & security review*, vol. 34, no. 1, pp. 125–133, 2018.

[17] N. Papernot *et al.*, "Practical black-box attacks against machine learning," in *Proceedings of the Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519.

[18] G. Kellaris, G. Kollios, K. Nissim, and A. O'neill, "Generic attacks on secure outsourced databases," in *Proceedings of the SIGSAC Conference*. ACM, 2016, pp. 1329–1340.

[19] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "LOGAN: Membership inference attacks against generative models," *Proceedings of Privacy Enhancing Technologies (PoPETs/PETS)*, pp. 133–152, 2019.

[20] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *IEEE S&P*. IEEE, 2017, pp. 3–18.

[21] M. Naveed, S. Kamara, and C. V. Wright, "Inference attacks on property-preserving encrypted databases," in *Proceedings of the 22nd ACM SIGSAC Conference*. ACM, 2015, pp. 644–655.

[22] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *23rd International Conference on Data Engineering*. IEEE, 2007, pp. 106–115.

[23] G. Sagirlar, B. Carminati, and E. Ferrari, "Decentralizing privacy enforcement for internet of things smart objects," *Computer Networks*, vol. 143, pp. 112–125, 2018.

[24] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, "Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic," *arXiv preprint arXiv:1708.05044*, 2017.

[25] A. Narayanan, J. Huey, and E. W. Felten, "A precautionary approach to big data privacy," in *Data protection on the move*. Springer, 2016, pp. 357–385.

[26] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA l. Rev.*, vol. 57, p. 1701, 2009.

[27] J. Abowd *et al.*, "Privacy-preserving data analysis for the federal statistical agencies," *arXiv preprint:1701.00752*, 2017.

[28] S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt, "Model reconstruction from model explanations," *arXiv preprint arXiv:1807.05185*, 2018.

[29] A. C. Fernandes *et al.*, "Development and evaluation of a de-identification procedure for a case register sourced from mental health electronic records," *BMC medical informatics and decision making*, vol. 13, no. 1, p. 71, 2013.

[30] C. Kushida *et al.*, "Strategies for de-identification and anonymization of electronic health record data for use in multicenter research studies," *Medical care*, 2012.

[31] K. Moselle, S. Robertson, and A. Koval, "'Real-World" de-identification of high-dimensional transactional health datasets." *Studies in health technology and informatics*, vol. 257, pp. 319–324, 2019.

[32] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, pp. 557–570, 2002.

[33] L.Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[34] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no. 1, p. 3, 2007.

[35] P. Zhao, J. Li, F. Zeng, F. Xiao, C. Wang, and H. Jiang, "ILLIA: Enabling k-Anonymity-Based Privacy Preserving Against Location Injection Attacks in Continuous LBS Queries," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1033–1042, 2018.

[36] A. Gkoulalas-Divanis, G. Loukides, and J. Sun, "Publishing data from electronic health records while preserving privacy: A survey of algorithms," *Journal of biomedical informatics*, vol. 50, pp. 4–19, 2014.

[37] P. Zhao *et al.*, "On the performance of k-anonymity against inference attacks with background information," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 808–819, 2019.

[38] R. C.-W. Wong *et al.*, "(α, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD*. ACM, 2006, pp. 754–759.

[39] R. Wang, Y. Zhu, T.-S. Chen, and C.-C. Chang, "Privacy-preserving algorithms for multiple sensitive attributes satisfying t-closeness," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1231–1242, 2018.

[40] C. Yin, S. Zhang, J. Xi, and J. Wang, "An improved anonymity model for big data security based on clustering algorithm," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 7, p. e3902, 2017.

[41] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[42] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[43] M. Lecuyer *et al.*, "On the connection between differential privacy and adversarial robustness in machine learning," *stat*, vol. 1050, p. 9, 2018.

[44] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1175–1191.

[45] J. Konečnỳ *et al.*, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[46] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434.

[47] Q. Yang *et al.*, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, 2019.

[48] M. Ammad-ud din *et al.*, "Federated collaborative filtering for privacy-preserving personalized recommendation system," *arXiv preprint arXiv:1901.09888*, 2019.

[49] F. Chen, Z. Dong, Z. Li, and X. He, "Federated meta-learning for recommendation," *arXiv preprint arXiv:1802.07876*, 2018.

[50] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks," *arXiv preprint arXiv:1812.00910*, 2018.

[51] S. Bogos, J. Gaspoz, and S. Vaudenay, "Cryptanalysis of a homomorphic encryption scheme," *Cryptography and Communications*, vol. 10, no. 1, pp. 27–39, 2018.

[52] X. Liang *et al.*, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proceedings of the 17th IEEE/ACM CCGrid*. IEEE Press, 2017, pp. 468–477.

[53] J. Götzfried, M. Eckert, S. Schinzel, and T. Müller, "Cache attacks on intel sgx," in *Proceedings of the 10th European Workshop on Systems Security*. ACM, 2017, p. 2.

[54] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data." in *NDSS*, vol. 4324, 2015, p. 4325.

[55] A. Padron and G. Vargas. Multiparty homomorphic encryption. Online: https://courses.csail.mit.edu/6.857/2016/files/17.pdf.

[56] H. Zhou and G. Wornell, "Efficient homomorphic encryption on integer vectors and its applications," in *Information Theory and Applications Workshop (ITA)*. IEEE, 2014, pp. 1–9.

[57] A. C.-C. Yao, "Protocols for secure computations," in *FOCS*, vol. 82, 1982, pp. 160–164.

[58] B. Mirzasoleiman, M. Zadimoghaddam, and A. Karbasi, "Fast distributed submodular cover: Public-private data summarization," in *NIPS*, 2016, pp. 3594–3602.

[59] G. Ayoade *et al.*, "Decentralized IoT data management using blockchain and trusted execution environment," in *International Conference on Information Reuse and Integration (IRI)*. IEEE, 2018, pp. 15–22.

[60] G. Zyskind, O. Nathan *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*. IEEE, 2015, pp. 180–184.

[61] X. Liang *et al.*, "Towards data assurance and resilience in IoT using blockchain," in *MILCOM*. IEEE, 2017, pp. 261–266.

[62] T. McGhin, K.-K. R. Choo, C. Z. Liu, and D. He, "Blockchain in healthcare applications: Research challenges and opportunities," *Journal of Network and Computer Applications*, 2019.

[63] I. Zavalyshyn *et al.*, "Homepad: A privacy-aware smart hub for home environments," in *IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 58–73.

[64] E. Fernandes *et al.*, "Flowfence: Practical data protection for emerging IoT application frameworks," in *25th USENIX Security Symposium*, 2016, pp. 531–548.

[65] J. Yang, K. Yessenov, and A. Solar-Lezama, "A language for automatically enforcing privacy policies," in *ACM SIGPLAN Notices*, vol. 47, no. 1. ACM, 2012, pp. 85–96.

[66] Z. B. Celik *et al.*, "Program Analysis of Commodity IoT Applications for Security and Privacy: Challenges and Opportunities," *arXiv preprint arXiv:1809.06962*, 2018.

[67] I. Ng *et al.*, "Making value creating context visible for new economic and business models: Home Hub-of-all-Things (HAT) as platform for multisided market powered by IoT," in *Panel Session at The Future of Value Creation in Complex Service Systems Minitrack of Hawaii International Conference on Systems Science (HICSS)*, 2013, pp. 7–10.

[68] Hub-of-All-Things. Last Accessed: Apr 01, 2019. [Online]. Available: https://www.hubofallthings.com/

[69] V. Costan and S. Devadas, "Intel SGX explained." *IACR Cryptology ePrint Archive*, vol. 2016, no. 86, 2016.

[70] N. Papernot *et al.*, "SoK: Security and privacy in machine learning," in *IEEE EuroS&P*. IEEE, 2018, pp. 399–414.

[71] Z. Ji, Z. C. Lipton, and C. Elkan, "Differential privacy and machine learning: a survey and review," *arXiv preprint arXiv:1412.7584*, 2014.

[72] M. Abadi *et al.*, "Deep learning with differential privacy," in *Proceedings of SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[73] C. Dwork and V. Feldman, "Privacy-preserving prediction," *arXiv preprint arXiv:1803.10266*, 2018.

[74] T. Hunt, C. Song, R. Shokri, V. Shmatikov, and E. Witchel, "Chiron: Privacy-preserving machine learning as a service," *arXiv preprint arXiv:1803.05961*, 2018.

[75] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *IEEE S&P*. IEEE, 2017, pp. 19–38.

[76] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, 2009.

[77] F. Zhang *et al.*, "Privacy-aware smart city: A case study in collaborative filtering recommender systems," *Journal of Parallel and Distributed Computing*, 2018.

[78] S. Berkovsky *et al.*, "Enhancing privacy and preserving accuracy of a distributed collaborative filtering," in *Proceedings of the ACM conference on Recommender systems*. ACM, 2007, pp. 9–16.

[79] R. Guerraoui *et al.*, "I know nothing about you but here is what you might like," in *47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2017, pp. 439–450.

[80] S. Spiekermann and L. F. Cranor, "Engineering privacy," *IEEE Transactions on Software Engineering*, vol. 35, no. 1, pp. 67–82, Jan 2009.

[81] P. Voigt and A. Von dem Bussche, "The EU General Data Protection Regulation (GDPR)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[82] M. Veale, R. Binns, and L. Edwards, "Algorithms that remember: model inversion attacks and data protection law," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 376, 2018.

[83] M. Young *et al.*, "Beyond open vs. closed: Balancing individual privacy and public accountability in data sharing," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 2019, pp. 191–200.