IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 August 2024, accepted 18 August 2024, date of publication 21 August 2024, date of current version 30 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3447017

SURVEY

Securing Personally Identifiable Information: A Survey of SOTA Techniques, and a Way Forward

IMRAN MAKHDOOM, (Senior Member, IEEE), MEHRAN ABOLHASAN, (Senior Member, IEEE), JUSTIN LIPMAN, (Senior Member, IEEE), NEGIN SHARIATI, (Senior Member, IEEE), DANIEL FRANKLIN, (Member, IEEE), AND MASSIMO PICCARDI, (Senior Member, IEEE)

Faculty of Engineering and IT, University of Technology Sydney, Ultimo, NSW 2007, Australia

Corresponding author: Imran Makhdoom (imran.makhdoom@uts.edu.au)

This work was supported in part by Food Agility Corporative Research Center (CRC) Ltd., funded by the Commonwealth Government CRC Program, which supports industry-led collaborations between industry, researchers, and the community; and in part by Robert Bosch (Australia) Pty Ltd., and Robert Bosch GmbH.

ABSTRACT The current age is witnessing an unprecedented dependence on data originating from humans through the devices that comprise the Internet of Things. The data collected by these devices are used for many purposes, including predictive maintenance, smart analytics, preventive healthcare, disaster protection, and increased operational efficiency and performance. However, most applications and systems that rely on user data to achieve their business objectives fail to comply with privacy regulations and expose users to numerous privacy threats. Such privacy breaches raise concerns about the legitimacy of the data being processed. Hence, this paper reviews some notable techniques for transparently, securely, and privately separating and sharing personally identifiable and non-personally identifiable information in various domains. One of the key findings of this study is that, despite various advantages, none of the existing techniques or data sharing applications preserve data/user privacy throughout the data life cycle. Another significant issue is the lack of transparency for data subjects during the collection, storage, and processing of private data. In addition, as privacy is unique to every user, there cannot be a single autonomous solution to identify and secure personally identifiable information for users of a particular application, system, or people living in different states/countries. Therefore, this research suggests a way forward to prevent the leakage of personally identifiable information at various stages of the data life cycle in compliance with some of the common privacy regulations around the world. The proposed approach aims to empower data owners to select, share, monitor, and control access to their data. In addition, the data owner is a stakeholder and a party to all data sharing contracts related to his personal data. The proposed solution has broad security and privacy controls that can be tailored to the privacy needs of specific applications.

INDEX TERMS Internet of Things, security and privacy, data sharing, regulations and policies, privacy-preserving computation, personally identifiable information.

I. INTRODUCTION

With the ongoing exponential expansion of the Internet of Things (IoT) and the increasing ubiquity of big data technologies, data (facts) and information (knowledge obtained through interpretation and contextualization of facts) have

The associate editor coordinating the review of this manuscript and approving it for publication was Lo'ai A. Tawalbeh.

become key strategic resources of the 21st century [1]. Many organizations compete to access such resources to generate economic value or strategic advantage. Due to the openness of cyberspace and Internet technologies, access to data has become easy, enhancing the efficiency of data-driven services. However, due to limitations in the security of the systems, e.g. the lack of privacy controls, there are increasing opportunities for data leakage, leading to identity theft and user privacy attacks.

It is estimated that by 2025, humans will generate approximately 463 exabytes of data every day [2]. Such a vast volume of data presents a growing challenge in guaranteeing its security and maintaining the privacy of its owners. Most of these data are private and derivative information associated with individuals, such as social media activity, purchasing habits, or health information, which is highly valued by both legitimate users and malicious parties [3, 4], and [5]. As more information of this type is collected and stored online, its collective value and the size of the potential attack surface continue to increase.

In this context, the significant issue is the misuse of data and the threats to user privacy. For example, in 2018, Facebook illegitimately shared the personal data of approximately 87 million users with Cambridge Analytica (a political data analytics firm) [6]. Cambridge Analytica profiled the users based on the acquired data, e.g., their political affiliations and the type of content they post and like. Later, users were psychologically exploited with tailored political content. In another incident, a ransomware attack on the Australian health insurance company Medibank in 2022, resulted in a data breach that affected 9.7 million customers [7].

IoT devices, in particular, represent a new class of threat to security and privacy. Researchers in [8] highlight numerous security and privacy issues associated with this domain, particularly discussing the evolution of cyber attacks that exploit the near-exponential growth of IoT devices and associated applications. For example, physicians collect extensive longitudinal records of patients' health data, including those collected from temporarily or permanently attached or implanted IoT devices. Similarly, vehicle insurance providers may track drivers' private information, such as location, speed, and other driving habits, through embedded IoT devices to more accurately estimate risk [9]. Such personal information is of enormous value to hostile actors (for example, for identity theft or fraud), creating a strong motivation for its theft or leakage. Existing solutions may protect sensitive customer data during transmission to some extent. However, they cannot guarantee data protection at the central point of collection, where a rogue administrator may disclose sensitive personal data [10].

In some cases, misconfiguration of analytics services can also compromise privacy [11]. For example, to facilitate debugging, target advertising, and improve service quality, developers of mobile applications may collect Personally Identifiable Information (PII) and use various analytic services to evaluate user behavior. PII includes any information that can be used independently or in combination with any other information to distinguish or trace an individual's identity, such as name, social security number, date and

TABLE 1. List of abbreviations.

Abbreviation	Definition
AMP	Approximate Minima Perturbation
AMQP	Advanced Message Queuing Protocol
APA	Australian Privacy Act
API	Application Programming Interface
APP	Australian Privacy Principles
App	Application
ASM	Attribute Setting Method
CAS	Central Authentication Service
CCPA	California Consumer Protection Act
CoAP	Constrained Application Protocol
CSL	Cybersecurity Law
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DDL	Data Definition Language
DFD	Data Flow Diagram
DL	Deep Learning
DP	Differential Privacy
DR	Dimensionality Reduction
DSL	Data Security Law
DTLS	Datagram Transport Layer Security
EHR	Electronic Health Record
FL	Federated Learning
GDPR	General Data Protection Regulation
HCI	Human- Computer Interface
HIPPA	Health Insurance Portability and Accountability Act
HMS	Hospital Management System
I-AM	Inform Alert & Mitigate
ID	Identifier
INTERPOL	International Criminal Police Organization
IoT	Internet of Things
IPEN	Internet Privacy Engineering Network
KNN	k-Nearest Neighbors
LDP	Local Differential Privacy
LocalDB	Local Database
MITM	Man-in-the-Middle
ML	Machine Learning
MOOC	Massive Open Online Course
MPC	Multi-Party Computation
NCB	National Central Bureau
NN	Neural Network
OTP	One-Time Password
PCA	Principal Component Analysis
PE	Privacy Engineering
PIA	Privacy Impact and Risk Assessment
PII	Personally Identifiable Information
PIPL	Personal Information Protection Law
POC	Proof of Concept
PPML	Privacy-Preserving Machine Learning
PSP	Privacy Service Provider
RAPPOR	Randomized Aggregatable Privacy- Preserving Ordinal Response
RBAC	Role-Based Access Control
SDLC	Software Development Life Cycle
SGD	Stochastic Gradient Descent
SGX	Software Guard Extension
SMPC	Secure Multi-Party Computation
SOTA	State-of-the-Art
SP	Service Provider
SQUARE	Security Quality Requirement Engineering
SVM	Support Vector Machine
TLS	Transport Layer Security
TPC-H	Transaction Processing Performance Council (TPC) Benchmark-H
UML	Unified Modeling Language
VCP	Virtual Cloud Provider

place of birth. Developers mostly use techniques such as anonymization and aggregation to reduce the disclosure of PII. However, these protective measures are usually not implemented as a core part of the application design. There is always a possibility that developers misconfigure the analytic services, thus leaking sensitive information. For example, an application developer may set attributes for their customers using certain Attribute Setting Methods (ASMs) provided by analytic services. In ASM, a user's PII, such as an email, a username, or date of birth, is replaced with a unique pseudonymous ID. The pseudorandom ID is then used to label the data associated with a particular user. In contrast, using PII as an ID may unnecessarily expose sensitive data and threaten users' privacy. If PII is wrongly used with ASMs, a user's private data may not be anonymized, violating the government's privacy regulations and the service provider's privacy policy. Hence, data analytics service providers threaten users' privacy once they gain control of their data. Consequently, users and app developers lose control of the data. Data owners must learn how their data are used and how many parties can access them. Even if data owners trust analytics service providers, long-term storage of user behavior data is always vulnerable [12].

According to [13], the General Data Protection Regulation (GDPR) relates only to PII and non-PII is outside of its scope of application. Therefore, it is important to carefully identify PII from non-PII as it determines whether an entity processing data is subject to the privacy regulations or not. However, in reality, the definition of privacy and sensitivity of data may differ from person to person. The classification of personal data is dynamic and, depending on the context, the same data point can be personal or non-personal [13]. Hence, a particular application or system cannot employ a single autonomous solution to differentiate between PII and non-PII. Similarly, if an entity, for example, a health service provider keeps patient PII but shares anonymized data for research with third parties, it is still termed risky and intolerable [14]. Consequently, data owners should select data assets, draft rules/conditions for sharing, and control access based on the sensitivity of their data.

In addition to ethical and mishandling issues related to private data, a small percentage of organizations implement a secure end-to-end digital transformation solution [15]. In reality, companies integrate various technologies from different vendors to provide services to their customers without considering data security and privacy controls [3]. On the other hand, a strict line has been drawn through various regulations such as the European Union's GDPR [16], China's Personal Information Protection Law (PIPL) [17], Indian Digital Data Protection Bill [18], California Consumer Protection Act (CCPA) [19], and Australian Privacy Act (APA) [20]. These laws require organizations to collect and process user data in a transparent way. Similarly, the increased interest of users in data privacy has forced organizations to reevaluate their data practices to avoid data breaches and privacy threats.

Despite the promulgation of strict regulations, security breaches that threaten user privacy indicate a void in existing digital transformation strategies. Hence, there is a need to review the existing IoT/personal data processing approaches to ascertain their effectiveness in preserving users'/data owners' privacy and compliance with data protection regulations. Table. 2 summarizes some of the significant threats to data security and privacy.

TABLE 2. Data security and privacy issues.

Category	Threats/Attacks
Data Confidentiality	Eavesdropping
	Man-in-the-Middle (MITM)
Data Integrity	Data forging
	Data injection
Data Availability	Denial of Service (DoS)
	Ransomware
	Unauthorized deletion
Data Accountability	Repudiation
	Ownership
Privacy (Transparency)	Unauthorized data sharing
	Unauthorized data collection
	Over-data collection
	Unanonymized data processing
	Disclosure of user identity
	Non-compliance with privacy regulations

A. RELATED WORK

According to the best of our information, none of the existing works presents a survey of PII and non-PII segregation techniques and provides an in-depth review of technical solutions to preserving users' privacy. For example, [21] presents the general cybersecurity risks and threats related to online education. Similarly, researchers in [22] conduct a comprehensive review of privacy and security threats to the IoT. However, the primary contribution of the research remains the taxonomy of IoT threats and countermeasures with a brief discussion on privacy issues.

Sen et al. comprehensively cover the security and privacy issues and related solutions in cloud-supported IoT, where users' data are collected, stored, and processed in the cloud [23]. Concerning countermeasures, the authors discuss conventional security measures such as open ports, security protocols, intrusion detection, and single-sign-on. Likewise, researchers in [24] discuss security and privacy issues with respect to big healthcare data. The researchers review approaches to counter security and privacy issues, but only briefly discuss anonymization and encryption techniques. Similarly, [25] studies the techniques

to preserve users' privacy in eHealthcare. However, the scope of the study revolves only around pseudonymization and privacy-preserving access control.

The primary issue with existing works is their concentration on security more than privacy. In addition, they do not purely talk about techniques to protect PII throughout the data lifecycle. However, existing solutions that address user privacy have limited scope or the work comprises individual solutions to classify and securely use PII. These techniques are discussed in Section III.

B. CONTRIBUTIONS OF THIS STUDY

The key highlights of this study are:

1) Construes compelling requirements from the world's leading data protection regulations.
2) Provides a comprehensive review of privacy-preserving techniques under different application scenarios.
3) Highlights the strengths, limitations, and status of compliance of each scheme.
4) Identifies challenges associated with user/data privacy.
5) Proposes a way forward for developing a secure and transparent framework to prevent PII leakage throughout the data life cycle.

C. ORGANIZATION

This article is organized into various sections: Some significant data protection regulations are highlighted in Section II. Section III illustrates current techniques for classifying and segregating PII and non-PII and protecting against privacy threats. Section IV introduces the new engineering dimension, i.e., Privacy Engineering, and Section V analyzes existing privacy-preserving techniques and highlights their pros and cons. The existing challenges are discussed in Section VI, and Section VII proposes a way forward. The study is finally concluded in Section VIII.

II. DATA PROTECTION REGULATIONS

Before we discuss state-of-the-art PII protection techniques, it is imperative to highlight some of the notable data protection regulations in the world. We have selected GDPR, PIPL, Indian Digital Data Protection Bill, CCPA, and APA because of the respective region/country's market share in the IoT industry and subsequent requirements for data privacy. For example, Europe is expected to dominate the IoT market by 2030, surpassing North America [26]. Similarly, China and India are among the top market shareholders in the Asia-Pacific region. APA is reviewed given the Australian Government's increasing emphasis on data privacy. However, first, we need to define some important terms:

- Personal Data: Information that can be related to and help identify an individual. For example, name, gender, browser cookies, biometrics, or location data.
- Data Subject: Data owner whose data are collected, processed, or analyzed. For example, patients, customers, users of the web site / mobile app.
- Data Controller: An individual in an organization with access to user data and authority to decide why and how data will be used.
- Data Processor: The party with delegated power (by data controller) to process users' data. This may include individuals, cloud servers that run analytics applications, or email service providers.
- Data Processing: Any manual or automated operation performed on user data, e.g., data collection, data storage, data analysis, erasing, sharing, or revoking access.

A. GENERAL DATA PROTECTION REGULATION (GDPR)

The GDPR [27] is considered the strictest data privacy regulation. It can be broadly divided into the following categories:

1) DATA PROCESSING

Regulations concerning data processing revolve around seven key data protection and accounting principles.

(a) Transparent and Legal Processing: Data processing must be legal and transparent. Data subjects must have visibility of the complete process.
(b) Minimal Data Collection: Data controllers and processors should collect the minimum possible data to serve their purposes.
(c) Period of Data Storage: Data must not be stored beyond their desired utilization.
(d) Data Confidentiality and Integrity: Integrity and confidentiality must be ensured during storage and processing.
(e) Data Accountability: It is the responsibility of the data controller to comply with all the above-mentioned GDPR principles.

2) DATA PROTECTION BY DESIGN

The developers should consider the following points when designing and developing their applications:

(a) What type of data is to be collected?
(b) How to avoid over-data collection?
(c) How can data be secured?
(d) How to anonymize data?

3) CONSENT

EU-GDPR has tough requirements concerning data subjects' consent. The prominent regulations in this regard include:

(a) Explicit and descriptive request for consent.
(b) Free consent of the data subject.
(c) The data subject must be a party to the data processing contract.
(d) Data subjects should be able to withdraw consent, restrict processing, control access, and delete data whenever desired.
(e) Data subjects must be informed in the event of data breaches.

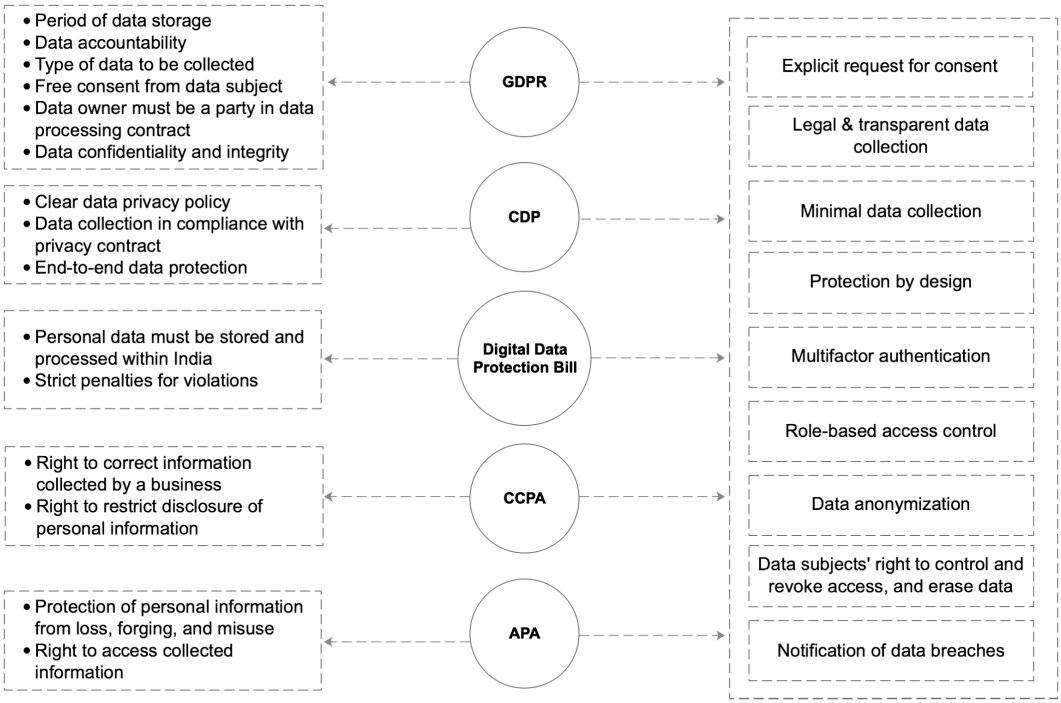

FIGURE 1. Data protection requirements.

4) TECHNICAL CONTROLS

(a) Multifactor authentication be implemented to access stored data.
(b) End-to-end data encryption should be in place to ensure confidentiality.
(c) Organizations should enforce role-based access control (RBAC) to ensure the security and privacy of collected data.

B. CHINA'S DATA PROTECTION (CDP) REGULATIONS

Data protection and privacy in China are based on three legal frameworks [28], i.e., PIPL, Cybersecurity Law (CSL) and Data Security Law (DSL). The main takeaways from these regulations are the following:

1) Consent: Requests for data subjects' consent to collect and process data should be clear and well described. Separate consent must be obtained for data collection, processing, sharing with third parties, overseas transfer, and public disclosure.
2) Privacy Policy: Data controllers are required to provide a data privacy policy to data subjects covering aspects including; the data controller's identity, list and purpose of collecting various types of data, location and period of data storage, scenarios when data will be shared with third parties or transferred across the borders.
3) Rights of Data Subjects: Data subjects should have the power to control access to their data. Data owners must be able to rectify or erase data whenever they want.
4) Data Processing: Data collection and processing should comply with the data privacy contract signed between all stakeholders. Excessive data collection must be avoided for all the reasons.
5) Data Confidentiality: Data must have end-to-end protection.
6) Data Breach: Data subjects must be notified of a security incident resulting in data leakage and unauthorized disclosure.

C. DIGITAL DATA PROTECTION BILL 2022 - INDIA

This regulation ensures safeguards for the collection and processing of personal digital data on the Internet within Indian territory. Data processing may involve offering data-oriented services or user profiling [18]. The bill enforces regulations related to data breach notifications, data subject rights, cross-border data transfer, and penalties.

1) Breach Notifications: The bill requires the data controller or data processor to issue notifications to the data protection board and to each affected data subject.
2) Data Subject Rights: The data controller must obtain explicit or deemed consent from the data subject. Data subjects must have the right to information about their data usage. Data owners should be able to rectify or erase data and revoke sharing whenever they want. In addition, data subjects must have the right to redress grievances in the event of any data breach.
3) Cross Border Data Transfer: Personal data must be stored and processed within India. However, if necessary, it can be transferred to other countries after approval from the central government.

4) Penalties: The digital data protection bill enforces strict penalties for the parties. For example, in the case of any violation, data subjects can be fined up to 10,000 Indian rupees (equivalent to $121). Similarly, data controllers or processors can be penalized up to 5000 million Indian Rupees (approximately 60.9 million USD).

D. CALIFORNIA CONSUMER PRIVACY ACT (CCPA)

This law allows consumers to exercise more control over their data [29]. It also guides organizations/businesses on effective implementation. The significant privacy rights for the consumers include:

1) Right to correct the information an organization or a business collects.
2) Right to restrict the use and disclosure of personal information.
3) Right to know what information businesses have about a customer, how businesses use the collected data and with whom the data are shared.
4) Consumers also have the option to opt out or revoke the sharing of their data.
5) Data controllers and processors are bound to issue notices to customers about privacy practices.

E. AUSTRALIAN PRIVACY ACT (APA)

This act revolves around thirteen privacy principles:

1) Personal information should be managed openly and transparently.
2) Data subjects must be able to hide their identity through anonymity and pseudonymity.
3) Non-sensitive personal information can be collected by an APP entity (any organization to which Australian Privacy Principles (APP) are applied) for a specific purpose through lawful means. In contrast, sensitive information can only be collected with the data subject's consent or when required by an Australian law or court order.
4) The information should be destroyed and deidentified when no longer required.
5) Any APP entity collecting personal information must inform the respective individuals about such requirements.
6) The information an APP entity collects for a specific requirement must not be used for any other purpose.
7) Any organization collecting personal information must not use it for direct marketing without the consent of the respective data subject.
8) Individuals' personal information can be shared by an APP entity with an overseas party only if that party complies with the APPs.
9) Organizations collecting personal information are prohibited from using government-related identifiers of an individual as their own without proper government permission.
10) APP entities collecting personal information must ensure that it is complete, correct, and up-to-date.
11) Organizations that store personal information should protect the data from unauthorized access, loss, forging, and misuse.
12) All individuals have the right to information.
13) The information collected can be corrected at the request of the respective data subjects.

In summary, Fig. 1 shows some of the significant requirements mandated by each privacy regulation discussed above. The figure also highlights data protection requirements common to the privacy regulations, which include:

- Clear and explicit request for data subject's consent.
- Legal and transparent data collection and processing.
- Minimum data collection for a minimal time.
- Data protection by design through various security controls, including multifactor authentication, end-to-end encryption, and RBAC.
- Personal data should be anonymized.
- The data owner should be able to withdraw consent, control access, restrict processing, revoke data sharing, and erase data whenever desired.
- Data subjects should be immediately notified of any data breach.

III. THE STATE-OF-THE-ART (SOTA) TECHNIQUES

For better understanding, as shown in Table. 3, we have categorized SOTA techniques to protect PII in various domains/applications, including healthcare, smart analytics applications, mobile and web applications, broad applications, smart city, and online education applications.

A. HEALTHCARE APPLICATIONS

1) PROACTIVE HEALTHCARE MANAGEMENT

Authors in [30] proposed an intelligent healthcare management solution enabled by IoT. The suggested model gathers data from sensors embedded in various health devices, including wearable gadgets (smart watches, pacemakers, etc.), smart modules in ambulances, doctors' workstations and hospital management systems (HMS software). The raw data are then forwarded to a central repository through secure communication protocols for further processing and analytics. The researchers claim to comply with the Health Insurance Portability and Accountability Act (HIPPA) by implementing security and privacy controls and the required monitoring and audit measures. The authors also mention the need for strong authentication (RBAC), classification of PII and non-PII, and controlled access to PII with the respective user's consent. However, this work lacks proof for the implementation of such security and privacy controls.

Concerning users' privacy, all devices associated with a particular user are assigned a unique ID during registration. The user and device mapping based on the unique ID is stored on a central server for later use. Consequently, sensor data collected and transmitted are identified based on the unique ID that maps them to the respective user/owner. It ensures that even if malicious users intercept or steal data without mapping to the user ID, they cannot gain anything.

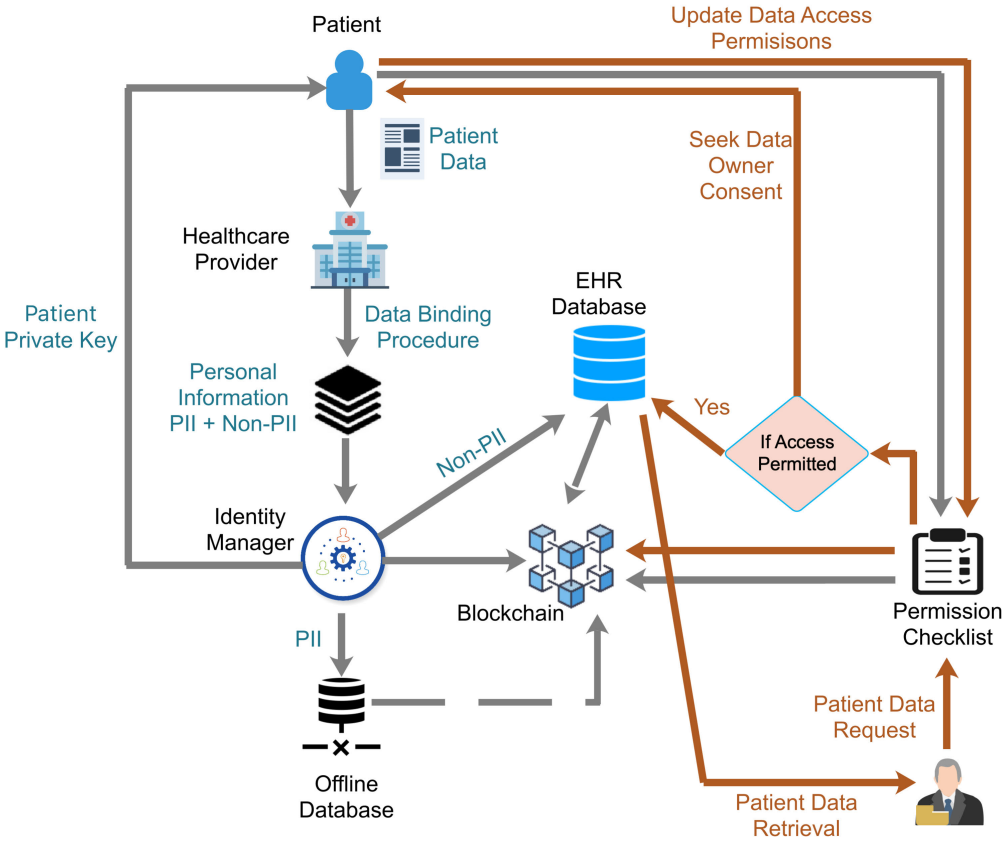

FIGURE 2. Patient onboarding and patient data request process.

2) BLOCKCHAIN FOR HEALTHCARE

The digitization of patient data in healthcare in the form of Electronic Health Records (EHRs) has undoubtedly improved the overall treatment experience. However, in the digitized world, there are always threats to data security and privacy. Hence, security researchers in [31] introduced a blockchain-based framework for secure storage and patient-driven sharing of health records according to GDPR. This work also facilitates building trusted AI models while ensuring data sources' traceability (provenance) and running analytics without disclosing PII.

As shown in Fig. 2, during the patient onboarding process (represented with dark gray arrows), the healthcare provider first collects patient data. It then segregates PII from non-PII and bundles the data (PII+non-PII). However, it is unclear how the researchers separate PII from non-PII, i.e., whether the classification is done while collecting information from the patient or at a later stage. The bundled data are then sent to the identity manager (integral to the blockchain). The identity manager generates a unique identifier (ID) for the patient and stores it on the blockchain. It also generates a private key for the patient, which is communicated to the patient via email or SMS. Moreover, the identity manager sends the non-PII data to the EHR database, and the PII data is sent to the offline database with its hash stored in the blockchain. In addition, during onboarding, patients provide the healthcare provider with a list of permissions for their data. The same is linked to the patients' private key and stored in the blockchain by the healthcare provider. Hence, patients with a unique ID, a private key, and access permissions to their data enable a patient-driven healthcare ecosystem. Consequently, data subjects have control over their data and no one can have access to data without an owner's consent.

On the other hand, to access a patient's data (shown in rust arrows in Fig. 2), the desired party sends a request for access to particular data items. The request is verified against the patient-created data access checklist. The third party can access the desired data items with the required permissions. Otherwise, the data owner is alerted concerning data access requests. If the owner updates the access permissions, a blockchain-based smart contract updates the data access checklist. Consequently, transactions concerning data access are logged in the blockchain for accountability and audit purposes. The immutable audit trail of data collection and sharing transactions preserves the provenance of the data and ensures its ethical use with the consent of the data owner. The blockchain-based, trustless, distributed, and credible data collection and sharing framework helps create a trusted decentralized AI model. However, the proposed framework

TABLE 3. SOTA personal data processing techniques.

Technique	Basic Idea	Advantages	Limitations/ Performance Overheads
Healthcare Applications			
Intelligent healthcare [30]	Propose an IoT-enabled intelligent healthcare management system	Highlights users' privacy concerns and the need for the classification of PII and non-PII. Recommends a strong authentication mechanism for controlled access	Does not specify the technique used for the classification of PII and non-PII
Blockchain for healthcare [31]	Leverages blockchain technology for secure storage and sharing of patient health records while keeping a transparent log of all the transactions	Empowers data owners/patients to control access to their health data, preserve provenance of data sources, facilitates creating a trusted and decentralized AI model	The mechanism of segregating PII from non-PII is not clear, more so it is a framework to ensure an audit trail of data sharing transactions and manage patient-driven access to data, scalability issues
I-AM: Privacy awareness in eHealth [33]	Introduces an inform, alert, and mitigation strategy to empower users to preserve their privacy	Presents a detailed analysis of various eHealth apps' usage of PII, highlights issues related to privacy information, risk alerts, and mitigation options in existing apps, suggests ways to inform and alert users, proposes mitigation techniques	Does not present a POC of the proposed strategy, especially the user interface for I-AM cycle
Smart Analytics Applications			
Privacy-preserving Machine Learning (PPML) [34]	Proffers a review of the ML threat environment and PPML approaches	Highlights threats including reconstruction, model inversion, membership inference and deanonymization	Current PPML techniques face the issues of flexibility, scalability, and policy enforcement
Carbyne Stack [35]	Uses a unique open source multi-party computation technology to offload computations to multi-party virtual cloud instances	Does not rely on a trusted third party, ensures confidentiality of data, has good scalability	Focuses on privacy-preserving analytics, currently in commercial testing phase
Mobile & Web Applications			
Code Analysis [36, 37]	Performs static analysis of an app's source code or binary to detect the use of PII	Detects access to sensitive information, e.g. location and contact details by apps and third-party libraries through sensitive Application Programming Interface (API) calls	Ineffective against dynamic code loading and reflection approaches, does not cater to the arbitrary identifiers generated by the apps
Manipulating OS and APIs [38, 39]	Modifies the OS or intercept APIs that access PII and then alert the users	Prevents leakage of sensitive data by replacing PII with mock data	Only works if the mobile device is rooted or jailbroken
Network Flow Analysis [40, 41]	Mostly, employs differential black-box fuzz testing schemes to monitor changes to network traffic	Detects and removes PII while being exfiltrating from a device	Scalability and privacy issues
PrivacyProxy [42]	Send cryptographic hashes of sensitive data (key-value pairs in HTTP requests) to the servers for analysis. Probable PII is identified based on the uniqueness of the hashes, each hash enacts the content of a particular HTTP request	Monitors real-time network traffic of app usage (without fabricated inputs), works on unrooted devices, scalable, employs an effective alert mechanism for users to control access to their PII, minimal performance overheads	Ineffective against custom encryption (at app layer), non-standard encodings, unconventional ways to track users, and certificate-pinned apps

TABLE 3. *(Continued.)* **SOTA personal data processing techniques.**

Privacy-Preserving Techniques for Broad Applications			
SYPSE - Privacy-first data management [43]	Utilizes pseudonymization, synthetic data, and separation between PII and non-PII to reduce the impact of data breaches	Provides a mechanism to delete user data by overwriting a portion or all of the data	Significant performance limitations, fewer privacy guarantees than other techniques like Differential Privacy (DP)
GDPR-compliant PII management [44]	A blockchain-based privacy-preserving data sharing scheme, stores PII in the off-chain local database while publishing non-PII and the hash of PII on the blockchain	Utilizes smart contracts to enforce terms and conditions of data usages, track changes to users' data, detect unauthorized modifications to PII, offers the right to forgetting users sensitive data, helps in identifying the privacy violators	The work is still in development stages, the controller and the processor has access to users' PII
Smart City Application			
PrivySharing [45]	Divides blockchain into numerous channels to share different types of data on a particular channel	Preserves users' privacy, provides user-defined fine-grained access control, rewards users for sharing their data, ensures compliance with GDPR	No key management mechanism for symmetric encryption, does not differentiate between PII and non-PII
Online Education			
moocRP [46]	Addresses the problems concerning modularity, transparency, and privacy in the collection, management, distribution, and analytics of Massive Open Online Course data	Authenticates users through existing institutional CAS to avoid overheads, encrypt data sets using users' PGP public key, data set access requests are approved or rejected by an administrator, researchers do not upload data for analytics rather run a local instance of analytics module, PII has separate authorizations than non-PII and is not viewable via analytics	Does not have an automated analytic module security screening, lacks alternative authentication protocols
Encryption-based Security			
CP-ABE [10]	A user's private key is composed of arbitrary number of attributes, and the sender encrypts the message by specifying the access structure based on desired attributes	Ensures privacy of data, attributes and access policy, suitable for resource-constraint IoT devices as it outsources intensive computations to multiple clouds	Only addresses one aspect of IoT data life cycle, i.e. secure sharing of data between cloud and end devices

may have scalability issues with increased users, as the Proof of Concept (POC) encompassed only a few nodes in a controlled environment.

3) I-AM

The general public is using several mobile eHealth applications (apps). These apps tend to access sensitive data related to users' health. Hence, it is natural for people to be concerned about their privacy. Most eHealth apps inform users about using sensitive information, such as location data, heart rate, step count, exercise history, or fall alert. However, users find privacy statements/notifications very complex and are often unaware after reading the privacy notice [32].

Researchers in [33] determine that, generally, the eHealth apps do not address the users' privacy concerns and usually collect PII that does not conform to the apps' functionality. For example, a fitness tracking app collects contacts, accounts, and phone identity information. Hence, the researchers introduced an inform, alert and mitigate strategy to preserve user privacy [33].

The inform, alert and mitigate (I-AM) approach focuses on designing and developing privacy-aware interfaces for mobile eHealth apps based on the analysis of some fitness trackers, weight-loss and medical apps. As shown in Fig. 3, the inform stage of the I-AM cycle informs a user about the privacy policy and an app's intended use of PII before it is installed. Especially the information about data that the app will collect, how and where it will be stored, when and with whom it will be shared, and how it may be used. Similarly, the alert module warns users of possible privacy risks. The authors suggest that the alert module should be developed as a separate layer

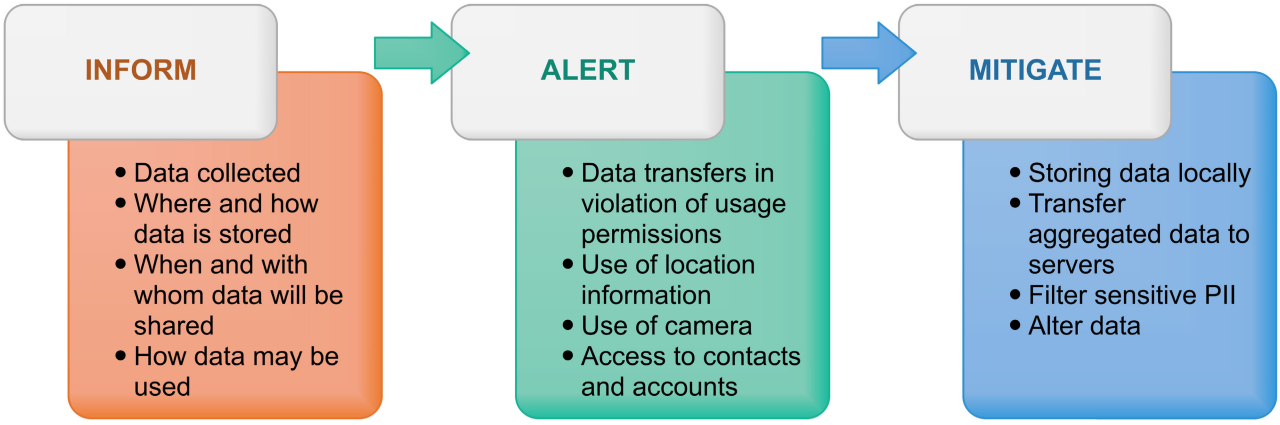

FIGURE 3. I-AM - Inform, alert, mitigate cycle.

between the OS and the app for simplicity and early adoption. Consequently, alert notifications can be displayed as scalable pop-up windows or small icons on the device status bar, depending on the alert's severity. Based on the alerts, the mitigation component provides various options for the users to reduce the incurred risks. The mitigation strategies may include:

- Storing data locally instead of forwarding it to the app's server.
- Transfer aggregated data to the server instead of individual sensor readings, e.g. sending an aggregated heart rate calculated over some time rather than an individual heart rate value at a specific time.
- Filter sensitive PII form data being transferred to the server, e.g. fall alert.
- Alter data such as the GPS coordinates to hide the actual location of the activity.

The concept of I-AM methodology seems to be effective in educating users about their use of data. However, there may be practical challenges in integrating the I-AM framework with the apps' user interface such that the users are not annoyed by receiving too many notifications. Moreover, the design of the user interface should not hamper an app's primary functionality.

B. SMART ANALYTICS APPLICATIONS

1) PRIVACY-PRESERVING MACHINE LEARNING

Researchers in [34] provide a detailed threat landscape with respect to various systems that employ machine learning (ML) models. The authors identify three major roles: data owners acting as the input party, the computation party performing ML tasks, and the results party. If all of these roles are played by two or more parties, privacy issues arise. Data owners should be aware of how their data are being used, who has access to them, and for how long. Depending on the threat environment (shown in Fig. 4), data owners share raw data with the computing party over a secure channel. The computation party creates feature vectors using raw data. Suppose that the computation party keeps the raw data unencrypted and does not delete the feature vectors after building the ML models. In that case, the system is vulnerable to insider and external attacks, including reconstruction, model inversion, membership inference, and deanonymization attacks.

Reconstruction attacks occur if feature vectors are not removed after building ML models, as with the Support Vector Machine (SVM) and k-Nearest Neighbors (kNN). In such an attack, security researchers exploited gesture features, including direction and velocity, to reconstruct touch events on mobile devices. Hence, failure to protect private data (raw data or feature vectors) can result in authentication failures causing security breaches and loss of privacy. Therefore, ML models such as neural networks (NNs) or ridge regression that do not store feature vectors should be preferred to avoid reconstruction attacks. Moreover, the ML models should not be shared with the results party.

An adversary can also build a feature vector similar to the one utilized in training an ML model by exploiting responses to the test queries sent by the results party. Such an attack is termed a model inversion attack. The attacker primarily generates an average representation of a particular class. If that class, e.g. represents a face image in the case of face recognition, then a particular person would be singled out. Hence, inversion attacks can pose a threat to user privacy. However, model inversion attacks do not infer any information about the actual items in the data set or help determine if a particular sample was in a data set given an ML model. Nevertheless, these attacks can be mitigated if the results party gets responses to the testing samples with limited output, such as rounded confidence values or the predicted class tables. In some cases, the results party may only be provided with aggregated responses calculated over multiple testing samples.

Consequently, given an ML model, if attackers want to confirm the presence of a sample in the data set, they may launch a membership inference attack. The adversaries may

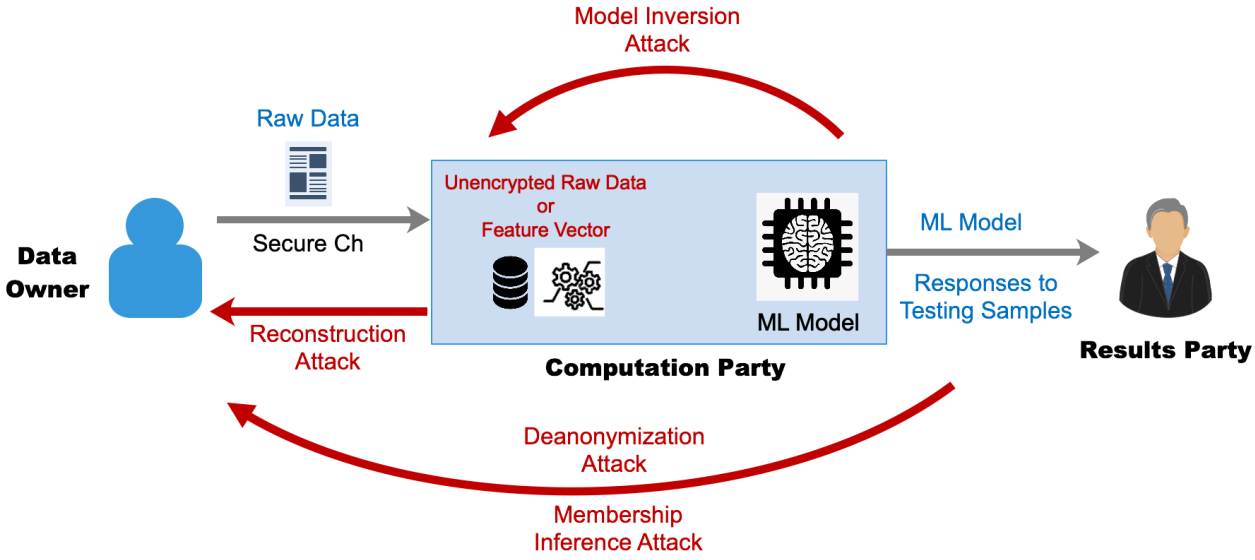

FIGURE 4. Threats to ML systems.

use a labeled sample and the probable output of the target ML model as input to an attack model [47]. The attackers' objective is to determine the presence of the sample in the training data set. As a defense, the authors in [47] recommend that the output of an ML model should be limited to the class label. However, the proposed approach may mitigate membership inference attacks in some cases, but may not completely prevent the attack. In addition, the authors suggest that DP might be a potent defense against membership inference attacks.

Moreover, releasing anonymized data in the public domain does not guarantee user privacy. Netflix released an anonymized data set containing 100 million users' movie ratings in 2006. The security researchers deanonymized the known Netflix users by combining the information extracted from IMDb and the Netflix data set. Hence, anonymization alone cannot preserve users' privacy in a hostile threat environment.

Researchers in [47] present various privacy-enhancing approaches focused on secure multi-party computations. Multiple data owners can collectively train ML models without disclosing sensitive data. Privacy-preserving ML techniques can be broadly divided into cryptographic schemes and perturbation approaches. In cryptographic techniques, the computation parties (servers) train the ML models on encrypted private data generated by the respective data owners. Such approaches do not require the input parties (data owners) to stay online. To ensure data security and privacy, cryptographic approaches may utilize homomorphic encryption, garbled circuits, secret sharing, or secure processors.

(a) **Homomorphic Encryption:** Fully homomorphic encryption [48] supports computations on encrypted data in the form of addition and multiplication operations. Similarly, complex functions may be built on simple addition and multiplication operations. However, most privacy-preserving ML techniques use additive homomorphic encryption to prevent computation overheads. For example, the Paillier cryptosystem [49] performs only addition operations on encrypted data and multiplication on plaintext. Besides, numerous data-packing schemes have been developed and used by different privacy-preserving ML approaches, such as the collaborative filtering system [50], to augment the effectiveness of additive homomorphic encryption. There is a Privacy Service Provider (PSP) and a Service Provider (SP) in such schemes. The PSP provides privacy-preserving computation services. On the other hand, the SP offers computation and storage services. SP gives suitable privacy suggestions to the data owners (its customers). Data owners then encrypt their private data with PSP's public key. The critical security requirement is that the computation parties, i.e. PSP and SP, must not collude. The data owners serve as the input and results parties both.

(b) **Garbled Circuit:** It is a secure computation mechanism for two parties. It facilitates the parties holding inputs x and y to compute the output of a function $f(x, y)$ such that no information about the inputs x and y is leaked other than what is inferred by the output of the function f [51]. For instance, if Alice and Bob want to see whether they agree to work mutually on a project without explicitly disclosing their personal preferences to each other. Accordingly, Alice and Bob's choices can be represented with the bits 0 or 1. Bit 1 means each of them agrees to work on the project as a group, and Bit 0 means otherwise. Hence, both of them need to know the output of an AND gate, i.e., if Alice's bit 'a' is 1,

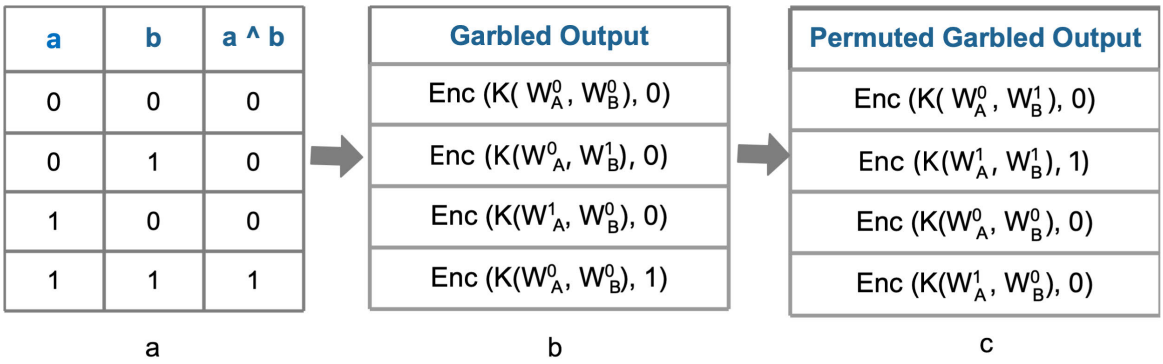

FIGURE 5. Garbled AND Gate: a) Garbled circuit generator, b) Garbled output, c) Permuted garbled output.

and Bob's bit 'b' is 0, then the output of a∧b equal to 0 means both Alice and Bob cannot work on the project as a group.

As shown in Fig. 5a, Alice is the garbled circuit generator. She selects four random labels, i.e., W_A^0, W_A^1, W_B^0, and W_B^1. W_A^0 is an event that indicates that bit $a = 0$ and W_A^1 means bit $a = 1$. Correspondingly, W_B^0 is an event in which bit $b = 0$, and W_B^1 means bit $b = 1$. Alice then computes the truth table for the four possible scenarios; $(a = 0, b = 0)$, $(a = 0, b = 1)$, $(a = 1, b = 0)$, and $(a = 1, b = 1)$. In the next step (shown in Fig. 5b), Alice encrypts the output corresponding to the respective scenarios using a key K derived through a key derivation function computed over respective labels (W_A^i, W_B^i). Finally, the garbled gate comprises four permuted ciphertexts (shown in Fig. 5c). Once Bob receives the garbled gate, he decrypts just a particular ciphertext to determine the decision on the group project. To do this, Bob receives the values W_A^a, W_B^b from Alice; since Alice knows a, she can send Bob W_A^a. Here, all the labels (as shown in Fig. 5c), are random and identically distributed, so Bob cannot infer anything about a from W_A^a. However, it is not easy to send W_B^b to Bob. Alice cannot send both W_B^0 and W_B^1 to Bob because Bob will then decrypt two ciphertexts in the garbled gate. Bob cannot ask for the exact ciphertext he needs, unless he wants Alice to learn about that. Therefore, to avoid this, Alice and Bob utilize *Oblivious Transfer* [52, 53] so that Bob learns only W_B^b without revealing b to Alice. However, to realize this, Bob must know when the decryption is correct. Otherwise, Bob cannot determine which ciphertext gives the right answer. Hence, just *XORing*, the encrypted value with the key, will not work here.

In the above example, Alice garbled a single gate. In actual settings, Alice and Bob may have multiple input bits with a more complicated function. Then Alice will have to garble the entire function circuit, that is, serving the gate output to other gates as input. Moreover, rather than encrypting the output of a gate, Alice only encrypts the label associated with an output bit, for example, W_G^0 or W_G^1. The label is later used to generate the decryption key for ciphertexts in other gates.

(c) **Secret Sharing:** This scheme divides a secret into multiple shares and distributes these shares among more than one party. An individual share is only useful if all the shares are combined to reconstruct the share [54]. In a threshold secrete sharing scheme, the secret can be reconstructed by utilizing only a specific number (t) of these shares. Similarly, to facilitate privacy-preserving ML, data owners (input parties) can create multiple shares of sensitive data and distribute them to the non-colluding computation servers (computation party) [55]. Accordingly, each server performs computations over the shares it receives and outputs draft results. Later, a proxy or a results party collects all the draft results and computes the final result. However, to avoid collusion of the computation servers and boost the confidence of data owners, it is to be ensured that the servers are located at different locations and controlled by different parties.

An exemplary instance of the secret sharing scheme is ShareMind [56], developed by Cybernetica. ShareMind is a privacy-preserving system for conducting a Principal Component Analysis (PCA) computation. It employs a parallelized approach to the secret sharing scheme. PCA, primarily used for dimensionality reduction in large data sets, involves transforming a set of variables into a smaller one that retains most of the original information.

The researchers in [57] presented an alternative variation of the secret sharing scheme. The shares are not sent to the computation party (servers), but are transmitted to other input parties (users). The proposed scheme aggregates user model updates to compute vector sums and train the NN model. Additionally, users mask their respective private update vector using their secret value and the secret shares (distributed among other users).

Moreover, as a precaution against premature protocol termination due to input parties leaving, users send

specific secret share and Diffie-Hellman private key to other users. According to the threshold secret sharing protocol, if at least t users can contribute to the protocol with the respective secret share in the last round, the server, responsible for message routing between users, performs the final computation. Such a communication-efficient protocol exhibits a maximum overhead twice that of its plaintext counterpart and is well suited for high-dimensional vectors.

(d) **Secure Processors:** Intel's Software Guard Extension (SGX) was initially developed to address trust issues in remote computation by creating a secure execution environment within an untrusted remote system. It guarantees data confidentiality and integrity throughout the computation process [58]. Presently, SGX is being utilized for performing privacy-preserving computations. Similarly, researchers in [59] have devised an SGX processor-based algorithm for data-oblivious ML, which encompasses various ML techniques such as SVM, NN, k-means clustering, matrix factorization, and decision trees. The proposed scheme enables multiple data owners to execute a specific ML task collaboratively with a computation party operating within an SGX-enabled data center. Despite the possibility of an adversary compromising all hardware and software components, the SGX processors remain an exception.
In this approach, a secure channel is created between end users (data owners) and the enclave. The data owners then authenticate themselves and verify the legitimacy of the ML code in the cloud. Subsequently, the data owners securely upload their sensitive data to the enclave. Once the data are uploaded, the secure processor executes the ML tasks. Later, the output is shared with the results parties through secure and authenticated channels.

(e) **Differential Privacy (DP):** DP protects against membership inference attacks by adding noise to the input data at every step of an ML algorithm or to the algorithm's output [65]. Most DP schemes rely on a trusted aggregator. However, in Local Differential Privacy (LDP), the input party (data owner) adds the noise locally without any involvement of a trusted party. Moreover, in the case of multiple input parties, original data privacy can be preserved by employing DP in distributed learning techniques. In addition, DP is resistant to post-processing, which means an attacker cannot compromise data privacy, even if they have access to auxiliary information. Hence, DP avoids deanonymization by protecting against linkage attacks. Based on how the randomness is applied, DP techniques can be categorized as follows:
Input Perturbation: This type of DP adds noise to the original data before performing any computation. The computation process generates a differentially private output. For example, in PCA, we usually perform the Eigen-decomposition on the covariance matrix. However, as a variation, symmetric Gaussian noise is added to the covariance matrix before performing Eigen-decomposition [60]. Hence, in the end, a differentially private matrix is produced instead of projected data.
Algorithm Perturbation: In this approach, iterative algorithms introduce perturbations to the intermediate values. For instance, the power method (an interactive algorithm) can execute Eigen-decomposition for PCA. Likewise, [61] incorporated Gaussian noise at each algorithm step, thereby working with the original covariance matrix to produce DP-PCA. In the same way, authors in [62] introduced Deep Learning (DL) system based on DP. They modified the Stochastic Gradient Descent (SGD) algorithm to introduce Gaussian noise in each iteration.
Output Perturbation: This methodology focuses on adding noise to the model generated by running a non-private learning algorithm. However, in scenarios where the modified output may lose its value, the exponential mechanism offers a viable alternative. To illustrate, if we consider a function $f(D, s)$ that reveals the proficiency of s on database D, the exponential mechanism selects an output s with a probability proportional to the function. Consequently, the result tends to exhibit a bias toward higher-quality instances. Likewise, DP-PCA can be achieved by approximating the top k PCA subspace while sampling a random k-dimensional subspace [63].
Objective Perturbation: This type of perturbation is typically conducted in two stages in its standard form. In the initial stage, a random linear term is added to perturb the objective function. Subsequently, the perturbed objective function's minima is determined in the second stage [64]. A study by researchers in [65] reveals that the resulting minima satisfy the DP requirements.
However, it should be noted that the objective perturbation only ensures privacy if the mechanism's output corresponds precisely to the minima of the perturbed objective. Due to scalability concerns, convex optimization algorithms often employ first-order iterative methods such as gradient descent or SGD. However, these methods cannot guarantee exact minima within a finite time frame because of the dependence of their convergence rates on the number of iterations. Hence, it remains doubtful whether objective perturbation, in its current state, can be effectively employed to reach minima in resource-constrained settings such as limited computing power. Consequently, the authors in [64] propose the Approximate Minima Perturbation (AMP) approach, which guarantees both privacy and utility by releasing a noisy *approximate* minima for the perturbed objective.

(f) **Local Differential Privacy (LDP):** To further explore the DP approach, an LDP methodology is recommended when the input parties cannot train an ML model due to

the lack of information. In LDP, each party first perturbs its data locally and then forwards the obscured data to the computing party. For instance, *Randomized Response* [66] is one of the oldest techniques of local privacy that prevents the visibility of sensitive queries to the users. For example, when the respondent flips a fair coin, they will answer truthfully if it is a tail. However, if it is a head, they will flip a second coin, and if head again, they would reply with "Yes," while if it is a tail, they would respond with "No."

Another technique, Randomized Aggregatable Privacy-Preserving Ordinal Response (RAPPOR) [67], gathers crowd-sourcing statistics from end-user client software. RAPPOR applies a random response to bloom filters with strong ϵ-DP guarantees. This approach is utilized in the Google Chrome web browser to collect client-side statistics, such as categories, frequencies, and histograms of different values and strings. In cases where multiple responses are periodically collected from a user, privacy protection is maintained by performing a randomized response twice.

For ML-oriented tasks, AnonML [68] is a method that uses randomized responses to generate histograms from multiple inputs. Then it uses histograms to create synthetic data for the training of ML models. This scheme is recommended when a trusted aggregator is unavailable and the input parties need more data to build an ML model independently.

(g) **Dimensionality Reduction (DR):** To achieve obscurity, this approach projects data onto a lower dimensional hyperplane [69]. It should be noted that this transformation is inherently lossy, rendering the accurate retrieval of the original data from the reduced dimension state impossible. The reason behind this limitation is more unknowns than available equations, resulting in infinite potential solutions. To address this issue, researchers in [69] propose using a random matrix to perform dimensionality reduction. However, a random matrix may offer limited utility. Consequently, alternative approaches combine unsupervised and supervised dimensionality reduction techniques, such as PCA, discriminant component analysis, and multidimensional scaling. These methods use reduced dimensionality to balance privacy-preservation and utility by identifying the optimal projection matrix.

However, researchers in [70] have observed that it is still possible to approximate the original data from the reduced dimensions. To address this concern, the authors suggest integrating DR with DP to achieve secure data publishing. Furthermore, for data sets containing samples with utility and privacy labels, [71] proposes a DR method that enables users to project their data in a way that makes it difficult to infer privacy labels while simultaneously maximizing the accuracy of utility label inference. Although this approach does not eliminate all privacy risks associated with data, it effectively controls data misuse when the privacy objective is known.

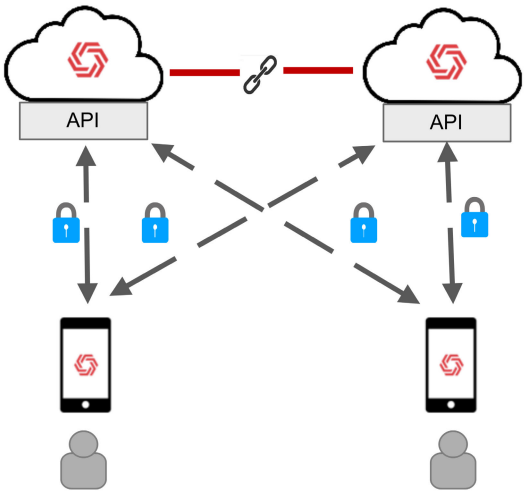

FIGURE 6. Federated cloud instances for secure MPC services.

2) CARBYNE STACK

Carbyne Stack is a novel open source technology that helps build scalable, secure multi-party computation (SMPC) applications [35]. MPC is a cryptographic technique that distributes computation between multiple parties so that no party can see the other parties' data. Carbyne Stack implements SPDZ-like MPC [72] in the client/server model, first described by [73]. In this variant of MPC, clients offload computations to a set of servers that act as the MPC parties. It does not require a trusted third party. Instead, the Carbyne stack depends on MPC to ensure the confidentiality of data throughout its life cycle, that is, at rest, in transit, and while in use. Moreover, due to the client-server MPC model, the proposed platform scales well while allowing any number of clients to collaborate securely on private data.

As shown in Fig. 6, Carbyne Stack federates cloud instances into multi-party virtual clouds (called Virtual Cloud Providers (VCPs)) to ensure the storage and processing of secret-shared data in a distributed, decentralized way. Each VCP hosts some fundamental services, including Castor, Amphora, and Ephemeral. Together, these services implement a fully functional cloud-native MPC party. Castor stores data-independent tuples generated in the MPC offline phase and serves them on request to Amphora and Ephemeral. Whereas, Amphora is a storage service that stores and retrieves client secrets. Secrets are stored as secret shares on distributed Amphora instances in a virtual cloud. They can be used as inputs to an Ephemeral computation or are created as results of such a computation. Ephemeral executes programs using the MP-SPDZ MPC framework. The secrets are fetched from Amphora at the beginning of an MPC program execution, and later they are written to Amphora at the end of the execution. Ephemeral fetches tuples from Castor consumed throughout the execution of the MPC program.

Clients using these services can securely offload computations on sensitive data. Carbyne Stack encompasses cloud-based MPC by leveraging containerization and the orchestration features of Kubernetes. In addition, to materialize flexible routing and deploy MPC as serverless workloads, Carbyne Stack relies on Istio [74] and Knative [75], respectively.

C. MOBILE AND WEB APPLICATIONS

1) DETECTION OF PII LEAKAGE IN MOBILE APPLICATIONS

Today, smartphone users are vulnerable to privacy breaches because most smartphone apps collect and utilize PII [76]. Although PII enables the provision of useful services to the users, many apps track users' behavior without their consent [77]. For example, on Android-based smartphones, the unique Android ID per device can be accessed by any app without permission. Researchers in [38] observe that third-party libraries make it difficult to protect PII. In addition, [78] also highlights that multiple libraries in an app may utilize the same PII differently. For example, one library may utilize a user's location for maps and another for targeted advertisements. Therefore, access to PII cannot be enforced per app, as a particular library may operate across multiple apps. Much work has been done to detect and mitigate leakage of PII in mobile apps. Based on the functionality, these techniques can be divided into three categories; code analysis, manipulating the OS and APIs, and network flow analysis.

(a) **Code Analysis:** This methodology employs static analysis of an app's source code or binary to identify how sensitive information is utilized. For instance, FlowDroid [36] and Droid-Safe [37] conduct flow analysis on specified sources and sinks to uncover potential leaks of PII. Similarly, AndroidLeaks [79] utilizes taint-aware slicing to detect cases of private information leakage by users. Another code analysis technique, Privacygrade [80], decompiles the code of mobile apps to identify anomalies arising from sensitive API calls made by third-party libraries. Furthermore, researchers in [81] apply text mining to the app code to determine the purpose of accessing user location and contact lists. The code analysis techniques neither modify the app functionality at runtime nor perform PII filtering. Instead, these schemes notify users of probable data leaks. Such methods are ineffective against dynamic code loading [82] and reflection approaches. In addition, many code analysis schemes can detect sensitive data leakage by relying solely on well-defined permissions or APIs rather than arbitrary identifiers generated by an app.

(b) **Manipulating OS and APIs:** An alternative approach to code analysis is the modification of the OS, identification of APIs that access PII, and subsequent generation of alert notifications to the respective users [38] and [39]. Some of the examples include MockDroid [83], which replaces PII with mock data by modifying the OS of an Android user. Similarly, TaintDroid [84] provides a custom Android OS version that allows information flow tracking from sources (requesting for sensitive information or PII) to sinks. In another work [85], researchers resort to binary rewriting to find out if user interactions in apps cause the leakage of sensitive information.

Although the above-discussed approaches enhance the privacy of iOS and Android users, the mobile device must be rooted or jailbroken to realize these methodologies.

(c) **Network Flow Analysis:** This approach detects and sometimes removes sensitive private information while it exfiltrates from a device. For instance, [40] employs a differential black-box fuzz testing scheme to detect information leakage. The proposed solution injects various types of inputs into the apps and observes related changes in the network traffic. Similarly, [41] uses differential black-box analysis to determine if an app's network traffic changes when some specific PII (e.g. ID or a location) is altered or kept the same. However, the differential black-box-based approaches do not scale well, as it is practically impossible to generate different sets of inputs per app for millions of apps out there.

The researchers developed AntMonitor [86] as an Android VPN service to facilitate sharing network traces with a central server. It allows users to selectively turn off traffic at the app level and make choices regarding forwarding all data or only the headers to the server. Similarly, PrivacyGuard [87] and Haystack [88] utilize an Android-based VPN service to intercept PII. However, these proposed techniques rely on regular expressions for PII detection, limiting their effectiveness to specific PII types. For instance, they may need help identifying PII when an app generates its identifier or employs non-standard encodings. Another approach named *Recon* [76], uses a VPN to forward all user traffic to a cloud-based proxy and employs heuristics for PII detection. However, since PII data reside in the cloud, this methodology raises specific privacy concerns [42].

2) PrivacyProxy

The authors in [42] introduced a proxy-based [88] privacy-aware method called PrivacyProxy, which utilizes network analysis and crowd-sourcing to automatically infer PII. PrivacyProxy deploys a VPN to intercept and analyze network data. The proposed scheme mitigates privacy and security threats by sending cryptographic hashes of sensitive data (key-value pairs in HTTP requests) to the servers instead of forwarding all of the user's data to the remote server for analysis. In this manner, PrivacyProxy identifies probable PII, such as email and MAC addresses, phone numbers, and social security numbers, based on the uniqueness of the hashes, with each hash representing the content of a specific HTTP request.

To observe PII usage by mob apps, PrivacyProxy monitors the data sent over the network. It segregates network traffic into different information flows based on an app's functionality or hosts to whom data is forwarded. To fulfill this requirement, PrivacyProxy leverages the internal VPN service of the host smartphone OS. Traditional VPN services utilizing SSL/TLS or IPSec protocols direct all device traffic through an encrypted tunnel to remote servers for data processing and forwarding to the final destination. In contrast, PrivacyProxy is built upon PrivacyGuard [87], which uses a custom VPN service that establishes a local server on the smartphone. Acting as a MITM proxy, the local server pretends to be the destination for all app-server communication. Subsequently, the local server forwards each request to the remote server and facilitates the necessary TCP handshakes. Moreover, to decrypt SSL/TLS, PrivacyProxy uses MITM SSL injection.

PrivacyProxy demonstrates adaptability and robustness against various data obfuscation techniques, such as PII hashing. Unlike HayStack [88] or PrivacyGuard [87], it does not require regular expressions or hardcoded rules. Moreover, this technique monitors more network requests and can detect PII more accurately. Since PII is detected and cryptographic hashes are generated on mobile devices, PrivacyProxy enables users to control access to PII. PrivacyProxy scales effectively with an increasing number of users and exhibits greater resilience to changes in data formats compared to other methods. It relies on valid network requests based on users' regular app usage patterns rather than fabricated inputs or modified PII, which improves the likelihood of detecting a wider range of PII due to its comprehensive coverage.

Regarding implementing crowdsourcing for PII usage detection in apps, PrivacyProxy monitors the outgoing data from a mobile device belonging to multiple users (at least five users) of the same app. The underlying idea is that the key-value pairs that frequently appear in a specific device's data streams are likely to be PII, as they are unique to a particular user. In contrast, key-value pairs that appear across multiple devices are less likely to be PII. However, the proposed scheme may not detect PII leakage if mobile app developers employ custom encryption, non-standard encodings, or other indirect methods to track users. Additionally, PrivacyProxy is ineffective against certificate-pinned apps. While certificate pinning enhances security by mitigating MITM attacks, it also hampers the ability of intermediaries to inspect network flows for PII.

D. PRIVACY-PRESERVING TECHNIQUES FOR BROAD APPLICATIONS

1) SYPSE

Researchers in [43] propose a mechanism named *Sypse* for the relational databases to manage users' sensitive data in a trusted environment. The suggested scheme uses pseudonymization and separation between PII and non-PII data to reduce the impact of data breaches. Sypse also allows users to delete their data by overwriting a portion or all of the data. The question may arise as to why the authors use pseudonymization instead of completely anonymizing users' data, as pseudonymized data can leak the identity of users through auxiliary data [89] and [90]. Nonetheless, pseudonymization is still considered a potent strategy against data breaches while facilitating internal data sharing [91, 92]. Pseudonymization makes it easier to remove information following a user's request *to be forgotten*. However, due to some practical challenges, you often cannot delete all the personal information of an individual. Hence, to make it difficult to reassociate a piece of information with a user, the most endorsed approach is to overwrite some pieces or complete data for the respective individual.

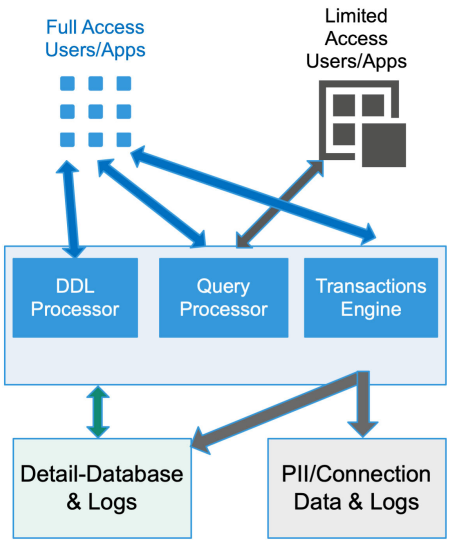

FIGURE 7. Sypse architecture (with two partitions).

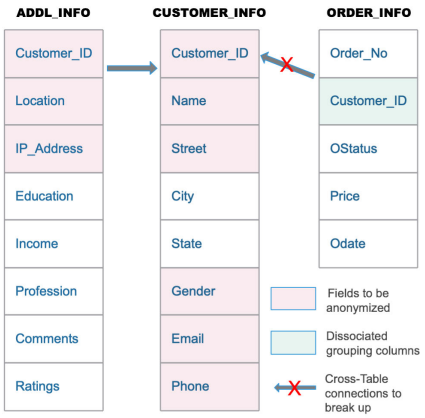

FIGURE 8. Annotated schema.

Accordingly, as shown in Fig. 7, Sypse divides the data into two partitions, i.e., the Detail-database and PII. Any personal information that may help identify an individual and connection information is stored in the PII-database. Similarly, information other than PII is kept in the Detail-database. It is imperative to highlight that the information in the Detail-database is pseudonymized and not completely

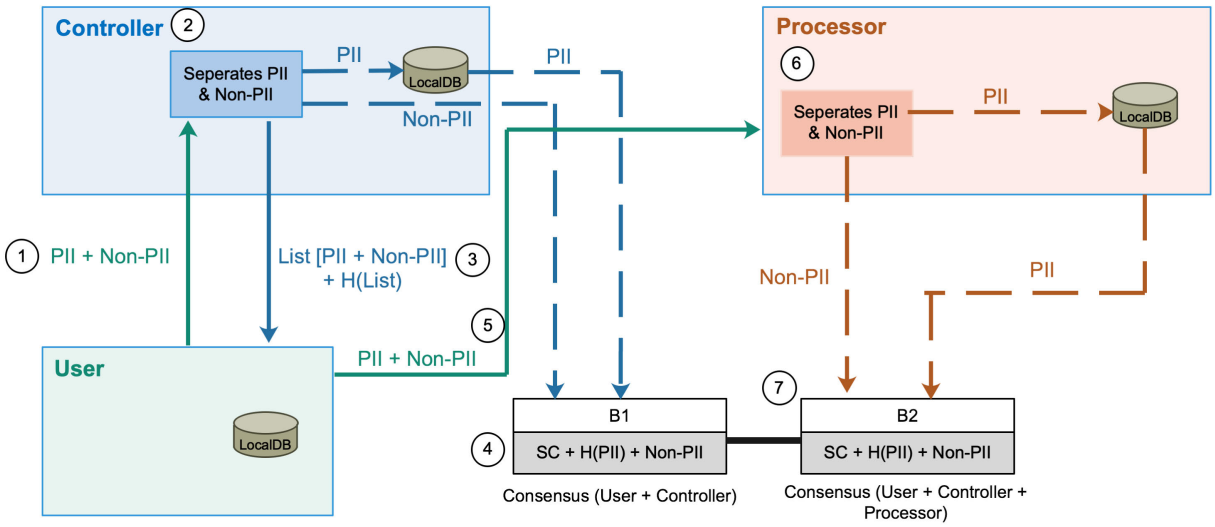

FIGURE 9. GDPR-compliant blockchain-based PII management system architecture.

anonymized or secured with DP or homomorphic encryption. Hence, it also requires careful handling along with strong access controls. The Data Definition Language (DDL) processor is responsible for intercepting schema updates and maintaining the tables in the respective databases. Similarly, the Query Processing Engine processes the user queries according to their corresponding access levels. Finally, the Transaction Engine routes the updates to the requisite tables and generates pseudonymized and synthetic data as required. The databases are administratively independent (separated) with different access controls for security reasons. Separation may be achieved through different geographical locations or virtual private clouds.

The highlight of Sypse is the way it segregates PII from non-PII. As shown in Fig. 8, the data fields, including Customer_ID, Name, Street, Gender, Email, Phone number, Location, and IP address, can be directly associated with a person. To separate such fields from other data fields, these are replaced with pseudonymous IDs and stored in the PII-database. Similarly, any data field that stores explicit information about a specific individual is treated similarly. However, even if PII is pseudonymized, analyzing multiple orders from the same customer ID may help reidentify a person. Hence, these groups also need to be broken up to prevent reidentification. Correspondingly, depending upon the sensitivity of data and the level of anonymization required, different tables can be dissociated from each other. For example, the connection between two customer tables can be broken up to prevent the possibility of reidentification by combining the fields of these tables. In this context, the identification of PII and non-PII in data fields is easy; however, practically, due to the lack of schema discipline, identifying the group/connection information requires detailed analysis.

Concerning privacy guarantees of the proposed model, it offers weak privacy compared to DP techniques. Moreover, anyone accessing both databases (i.e., Detail and PII) can extract private information. Nevertheless, someone with access to only the Detail-database cannot infer any PII about a specific person or correlate information across multiple tables. In contrast, access to only a PII-database may result in loss of privacy. Hence, strong access control mechanisms need to be deployed.

Sypse used and evaluated different strategies for segregating PII and non-PII. The first involves duplicated tables and columns. A subset (data fields) of the columns containing sensitive information, or PII (marked for anonymization), is copied into the PII-database. Synthetic data replace the values in the Detail-database. One of the limitations of this strategy is that most update-transactions require both tables to be refreshed. Additionally, due to the mapping tables, the PII-database size increases in proportion to the Detail-database. Similarly, to delete a customer's information, requisite data and grouping information are removed from the PII-database tables.

The second strategy is encrypted columns in which all the PII and connected columns are encrypted. The encrypted data and encryption keys are maintained in the same table. Moreover, a different key is stored in the PII-database for each customer. Nevertheless, a drawback of this technique is that joins involving real data in encrypted form are very difficult compared to just synthetic data. This is because all the encryption keys are checked individually to find the correct one for a given encrypted order or line item. Finally, the third strategy is pseudorandom sequencing, which balances the previous two strategies by maintaining a small-sized PII-database and efficient joins. Pseudorandom sequencing refers to taking a hash of the per-customer encryption keys that make it easy to find an encryption key for a given encrypted data item.

Summarily, to provide stricter privacy guarantees, techniques like data partitioning, pseudonymization, and using synthetic data may be integrated with other approaches such as homomorphic encryption [93], DP or secure hardware such as Intel SGX [94].

2) GDPR-COMPLIANT BLOCKCHAIN ARCHITECTURE

GDPR focuses on protecting individuals' information. All stakeholders handling data must attain the data owners' consent and ensure secure use without leaking any PII. Moreover, GDPR also provides the *right to forget* in which user data must be erased on request. Hence, researchers in [44] propose a blockchain-based GDPR-compliant PII management system. The proposed model involves three nodes; a user, a controller, and a processor. All nodes maintain a copy of the blockchain and verify the transactions before adding a new block. The user is a data owner whose information is used by other parties. The controller is the legal or public entity that processes users' PII. In comparison, the processor is the legal or public party that processes user information on the controller's behalf.

As shown in Fig. 9, (1) First, the users share their personal information with the controller. (2) The controller segregates the PII from non-PII and stores PII in the local database (LocalDB). (3) Then, the controller computes the hash of PII and shares a list of PII and non-PII and the hash value with the user. (4) Later, a smart contract is created based on the terms and conditions of data sharing between the user and the controller. A new block, say B1 containing the smart contract, a hash of PII, and non-PII, is proposed and added to the blockchain after achieving consensus between the user and the controller.

Subsequently, if the controller wants to share a user's PII with the processor for further analytics, (5) it forwards the list of the user's PII and non-PII to the processor. (6) The processor then separates the PII from the non-PII and stores the PII in its LocalDB. (7) The three parties, i.e., the user, the controller and the processor, then agree on the terms and conditions of data usage, and the processor creates a smart contract. Later, the processor proposes a new block B2 containing the new smart contract, the hash of PII, and non-PII. The new block is then added to the chain after an agreement (consensus) is reached between all three parties, i.e., the user, controller, and processor.

Concerning the right to forget, once the users need to erase some of their personal information, they inform all the nodes to delete or modify particular data. The nodes then verify from the smart contract whether the desired operation is permitted. If verified, the nodes delete or modify the data. Later, a list of erased/updated data and its hash is shared among all the nodes. Finally, the hash of the modified data is updated on the blockchain. The proposed methodology has certain advantages, such as; tracking changes to user data and detecting unauthorized modifications, the right to forget users' sensitive data and identification of privacy violators. However, the proposed scheme is still in the development stages. Moreover, the controller and processor can access the users' PII stored in their LocalDB, threatening user privacy.

E. SMART CITY APPLICATION

Security researchers in [45] propose "PrivySharing," a solution to securely share private data in a smart city environment based on fine-grained access controls defined by the user. The proposed framework complies with many GDPR requirements and utilizes blockchain technology for fast transaction settlement. As shown in Fig. 10, the blockchain network is divided into numerous channels depending on the type of data to be shared. Hence, data privacy is partially preserved by publishing specific data on a particular channel. For example, channel 1 shares smart energy and channel 2 shares smart transport data only with channel members (stakeholders) concerned.

PrivySharing also ensures the users' right to forget information. The data can be deleted while its transaction history remains intact for audit at later stages. Access control rules are embedded into blockchain smart contracts to provide user-defined controlled access to data. Hence, users can control the permissions and duration of data sharing. In addition, data owners are given an incentive in the form of digital currency equivalent to the period for which their data are shared.

The proposed framework ensures data privacy and integrity according to the wishes of the data owner in various network settings. It also effectively prevents Sybil and false data injection attacks. For additional security, data owners can encrypt their data using symmetric encryption and then share the decryption keys with concerned stakeholders over other communication media. The authors claim that it is scalable (in terms of the number of users) and energy and computationally efficient framework for secure data sharing. However, PrivySharing does not explicitly segregate PII from non-PII.

F. ONLINE EDUCATION

Security researchers in [46] present moocRP to address problems concerning modularity, transparency, and privacy in the collection, management, distribution, and analysis of Massive Open Online Course (MOOC) data. Currently, most universities associated with online higher education platforms such as edX [95] and Coursera [96] are often disoriented in managing large data received from providers. Moreover, another problem is how to quickly and accurately determine what data is actionable and to which instructor/professor it is related. Accordingly, moocRP tries to solve the issues of preparing data before it is securely shared with the users and later examining it for actionable information.

The researchers developed an open source tool to ensure transparency in data collection and background operation of different data processing and analytics modules of moocRP on the collected data. Resultantly, the uploaded analytics and visualizations have the source accessible to all the users. Moreover, a data model is used only if it discloses

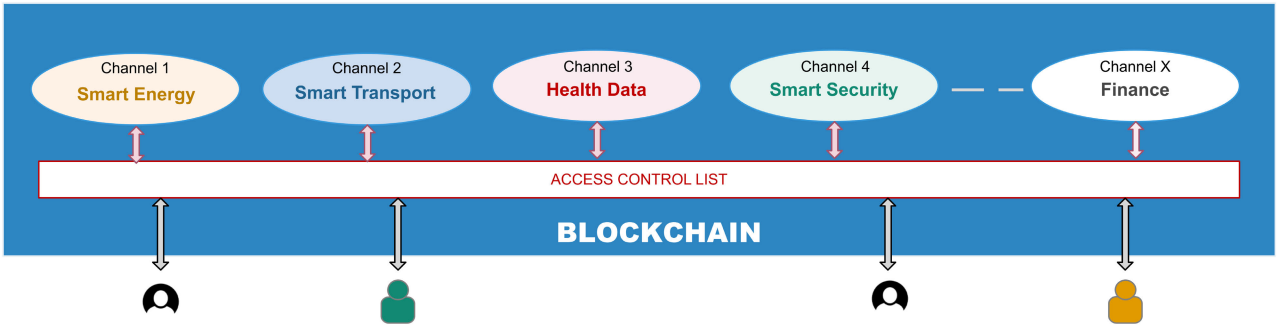

FIGURE 10. PrivySharing - Network architecture.

information about exposed data elements to the authors and users of the analytics services. Similarly, in accordance with the modularity principle, moocRP offers a feature-rich, user-friendly, and time-saving analytics module to the users.

In addition, to preserve users' privacy, moocRP brings analytics to the data source, in contrast to performing analytics centrally on previously collected data. Researchers/instructors of a particular online course do not need to upload data to a centralized server; rather, they can run an instance of the moocRP tool and execute the desired analytics module locally. This allows researchers to directly access the transformed aggregate data or other analytical outcomes necessary to evaluate the correlation between student activity and the success of a specific course. In summary, edX retrieves the tracking and database logs from Amazon S3 servers, and grants users access to raw data that comprise compressed and encrypted event files. Since each online course is served by multiple servers, aggregating data from various sources is required to obtain the complete event data for a specific course. In contrast, moocRP offers data cleansing and processing scripts that facilitate multiple procedures, including data ingestion, decryption, and visualization of analytics results.

Numerous measures adopted by moocRP are designed to ensure the security and privacy of data sets. Firstly, registered users (researchers/instructors) log into moocRP API through the respective institution's Central Authentication Service (CAS). In this way, users are authenticated based on existing CAS instead of implementing another authentication service with additional overheads. An authenticated user is then logged into the moocRP services. Once logged in, the user submits a request for the desired data set using a request data form. Optionally, moocRP implements separate authorizations for PII and non-PII versions of a data set. After a data request is submitted, an administrator either grants or denies the request, which is then reflected on the instructor's dashboard. To ensure data confidentiality, the data set is encrypted using the users' PGP public keys, which they provide during registration. Subsequently, the encrypted data set is made available for download using a one-time download link provided by the server.

Moreover, as shown in Fig. 11, to ensure the security of the connection between a client and the server against MITM attacks, moocRP uses the HTTPS protocol. Administrators of moocRP benefit from additional security features, such as user management functionalities that include user removal and editing, deletion of uploaded malicious modules, and comprehensive log of user activities within the system. These logs can be parsed to detect any abnormal or suspicious actions. It is important to note that moocRP operates as a relatively closed ecosystem, with authorized access limited to users affiliated with specific institutions.

Additionally, moocRP incorporates a meticulous approval process for data requests and analytic modules overseen by the administrators. This process ensures that PII data remains inaccessible through the analytics feature. Institutions employing moocRP also enforce security policies concerning non-PII data. Moreover, data processed by the browser-based analytic modules are not stored on disk, mitigating the risk of data interception by a MITM attacker. However, it should be noted that an attacker with full privileges to download the data can only access encrypted analytics. It is worth mentioning that moocRP currently lacks an automated security screening mechanism for analytic modules and would benefit from integrating alternative authentication protocols.

G. ENCRYPTION-BASED SECURITY

Security researchers in [10] propose a broadcast encryption technique based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to address the weak link in IoT, i.e. downward communication from the cloud to the end devices. The concept of ABE is that users should be able to access encrypted data only when they possess specific attributes, e.g. the users are located at a specific location and have security clearance above a certain level. In conventional ABE approaches, a central server that stores the data and manages access control is a single point of failure. In addition, traditional ABE schemes use user attributes for the description of encrypted data and embed policies in users' keys. In contrast, in the CP-ABE technique [97], the attributes represent users' credentials, and the sender that encrypts the data establishes a policy on how users can decrypt the data. CP-ABE functions similarly to RBAC.

In CP-ABE, a user's private key is made up of an arbitrary number of attributes. The sender encrypts the message and

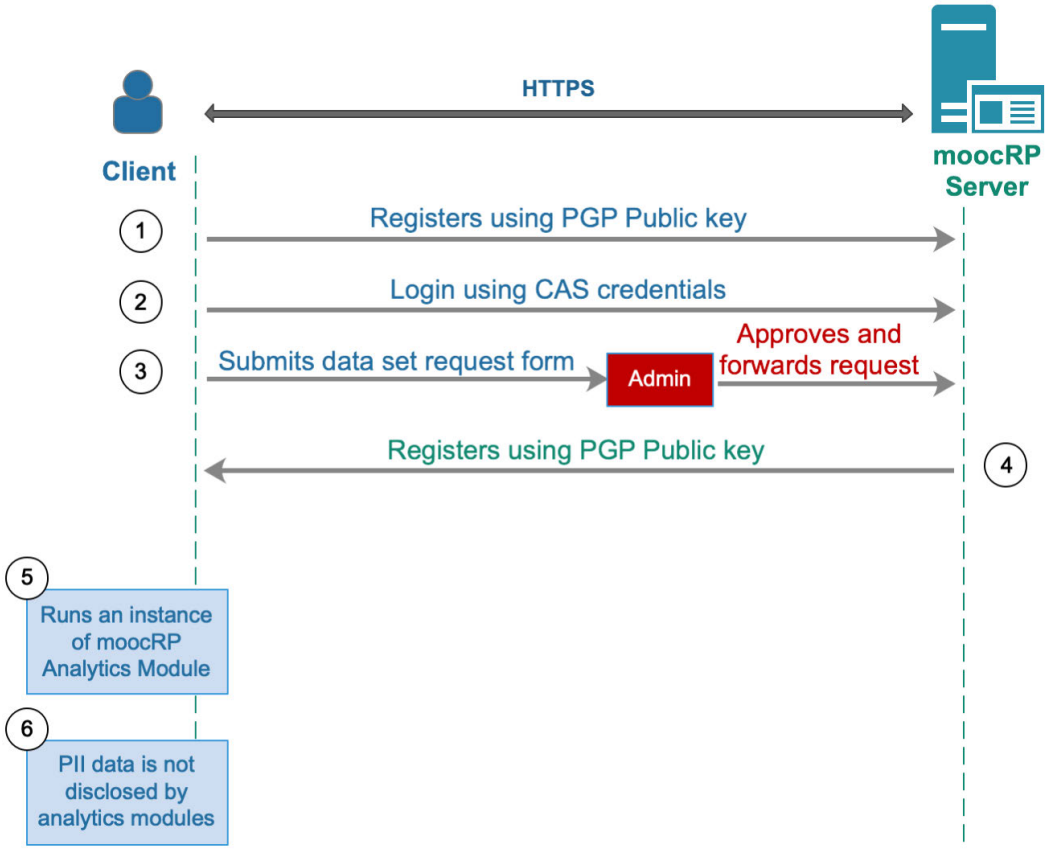

FIGURE 11. moocRP - data security features.

specifies the access structure based on certain attributes. For example, the General Secretariat of the International Criminal Police Organization (INTERPOL) wants to send a memo secretly to two of its representatives at the National Central Bureau (NCB) in Estonia and Latvia. The General Secretariat wants only specific individuals with desired attributes to be able to decrypt the memo. Hence, the sender at the General Secretariat can establish the access structure for the secret memo in the following manner: (("NCB") **AND** ("ESTONIA" OR "LATVIA") **AND** ("CLEARANCE-LEVEL > 4") **OR** ("NAME: XYZ")).

The scheme proposed in [10] improves existing CP-ABE techniques by outsourcing intensive computations required for decryption to multiple cloud platforms. Outsourcing computations was deemed necessary because the computational complexity increases linearly with an increase in the attributes (characteristics of access policy) and also due to the costly operations performed by the receivers to match their attributes with the access policy. Another reason is that conventional ABE approaches have high computational costs for resource-constraint IoT devices.

The given scheme ensures privacy-preserving information sharing between the cloud and IoT devices. The researchers present two schemes, i.e., parallel-cloud and chain-cloud ABE, to provide privacy to data, attributes, and access policy. The primary advantage of using ABE is reduced communication complexity compared to unicast approaches, in which encrypted messages are sent individually to each device/user. In addition, ABE also avoids complex key management requirements compared to unicast schemes. However, the CP-ABE approach provides data security and privacy in a single segment of the IoT data life cycle, that is, while disseminating information from cloud platforms to end users/devices.

- **Parallel-Cloud Scheme.** In this approach, as shown in Fig. 12, the users' attributes are divided into n parts, and each part is outsourced to one cloud server. An end user with an IoT device has to connect to all n servers simultaneously. Each cloud server works on the received access policy and the encrypted message in parallel and sends interim results to the end user separately. In this way, the users' attributes are protected compared to a trusted cloud server. The parallel-cloud approach supports only **AND** gate for access structure, offering increased data privacy with lower overheads. However, it is less flexible in terms of specifying access policies.
- **Chain-Cloud Scheme.** In contrast to parallel-cloud, the chain-cloud scheme allows message receivers to decide how many and which cloud servers to use and what attributes are delegated to each cloud. As shown in

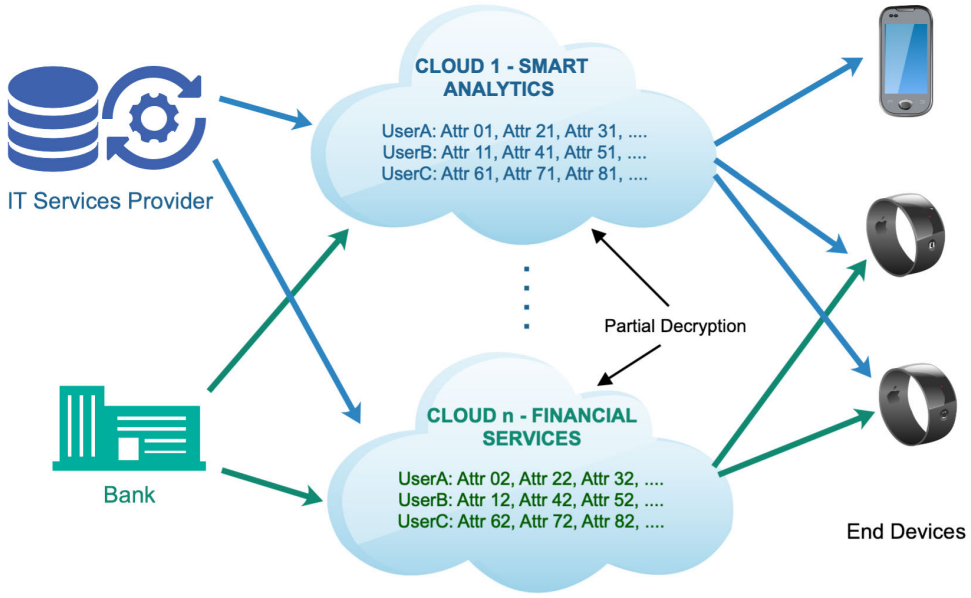

FIGURE 12. Privacy-preserving parallel-cloud targeted broadcast.

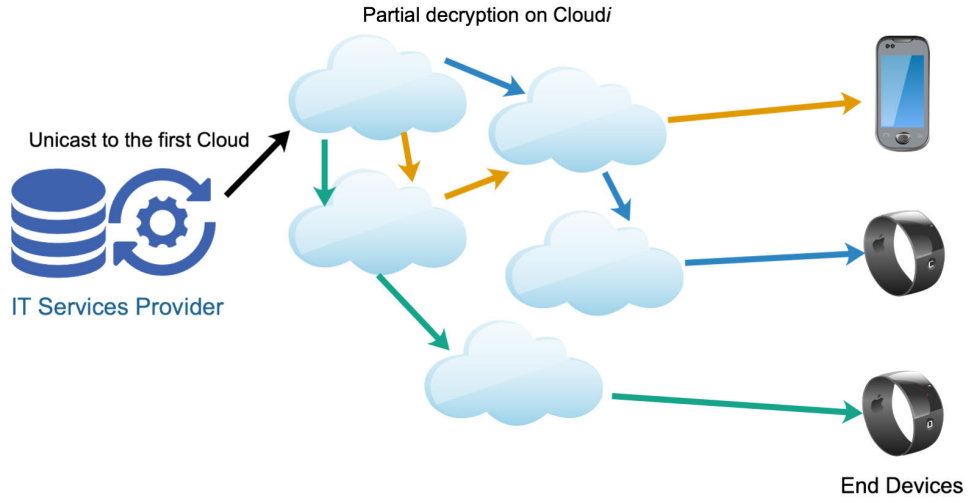

FIGURE 13. Privacy-preserving chain-cloud targeted broadcast.

Fig. 13, end users/devices can decide which three sets of cloud servers they want to connect, thus forming different communication paths. Moreover, unlike the parallel-cloud scheme, the sender can encrypt messages with a more detailed access policy consisting of k-threshold gates. Each cloud employs a bloom filter to store the attributes delegated by the receiver. The cloud servers partially decrypt the encrypted messages using these attributes by satisfying the respective access structures.

IV. PRIVACY ENGINEERING

The privacy controls/techniques described in the previous section are considered insufficient. As a result, a new field of engineering called *Privacy Engineering (PE)* has emerged. It is the practice of applying privacy protections to personal data while building systems and apps. According to [98], in the realm of the Software Development Life Cycle (SDLC), PE can be defined as research and practice aimed at creating privacy-oriented solutions. Similarly, [99] defines PE as a technique that enables organizations to provide sufficient protection to all stakeholders' data. PE may involve different activities, but it aims to embed privacy into systems and apps. For example, developing a code that minimizes the risk of PII leakage. Similarly, for a user, PE may involve creating effective privacy controls based on the user's preferences [100].

Software developers face a significant challenge in implementing GDPR guidelines, particularly privacy-by-design requirements, to protect user privacy, as highlighted by

the Internet Privacy Engineering Network (IPEN) [101]. According to Gutwirth et al. [102], the data holder or controller must incorporate all necessary controls to protect user privacy before and during data processing. Furthermore, [103] introduced seven foundational principles of privacy-by-design, including consideration of user privacy, proactive and preventive approaches, privacy as the default setting, privacy-by-design, full functionality, end-to-end security, and transparency.

Fig. 14 illustrates the PE approaches defined by researchers in [100], encompassing three key areas: privacy impact and risk assessment (PIA), model-based design and privacy requirement engineering. PIA focuses on identifying and mitigating privacy risks to provide privacy protection [104]. For example, CNIL [105] presents a risk model that includes privacy threats, inherent vulnerabilities, and potential risk sources. Similarly, PRIAM is a privacy risk assessment technique [106] that evaluates risks based on information collected about seven components: data, risk sources, privacy vulnerabilities, probable events and harms, and the system itself. Risks are derived for each category using harm trees. The harm trees draw the relationships between privacy weaknesses, potential events, and associated harms.

LINNDUN, another model-based approach, offers software engineers a systematic threat assessment using data flow diagrams (DFD) to depict the flow of personal data through various processing events [107]. LINNDUN incorporates seven threat categories: linking attacks, identification, non-repudiation, disclosure, detection, non-compliance, and unawareness. Similarly, various strategies are proposed in [108] to develop privacy-by-design IT systems and assess the impact of privacy. These strategies include minimizing, hiding, separating and aggregating data, enforcing and maintaining controls, informing users, and demonstrating privacy compliance.

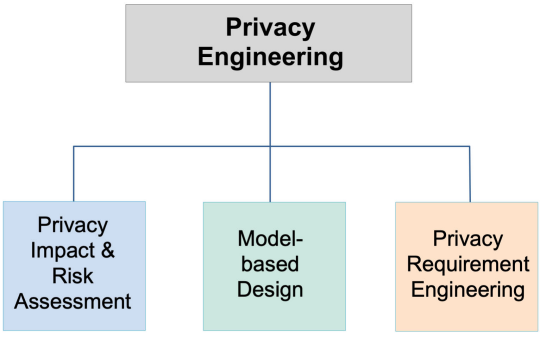

FIGURE 14. Approaches to privacy engineering.

ProPAn, the Problem-based Privacy Analysis approach [109], models the problems to identify threats to user privacy. Using Unified Modeling Language (UML) profiles that contain privacy requirements, you can determine the personal information that requires protection [110]. Adapting the Security Quality Requirement Engineering (SQUARE) process [111] is one way to engineer privacy requirements. There are nine steps in the SQUARE process, including techniques for deriving security requirements, categorizing and prioritizing them, and conducting inspections. However, privacy remains a challenging and elusive concept to fully incorporate within the SDLC.

V. GAP ANALYSIS

Although the SOTA privacy-preserving techniques discussed in Section III exhibit some advantages, all the schemes also have weaknesses. It is therefore difficult to declare that a particular technique offers end-to-end privacy protection to the users throughout the data life cycle at worthy costs. For example, the IoT-enabled intelligent healthcare management system [30] advocates separating PII from non-PII and strong access control measures to protect sensitive data. However, the proposed scheme does not specify the technique for classifying PII and non-PII data. Similarly, SYPSE [43] provides fewer privacy guarantees than DP or homomorphic encryption-based techniques. However, fully homomorphic encryption [112] with high-performance overheads is suitable for a limited family of computations [113]. SYPSE also has significant performance limitations. Similarly, blockchain for healthcare [31] focuses on secure storage and sharing of patients' health records while keeping a transparent log of all transactions. However, it does not mention the mechanism of segregating PII from non-PII. In addition, the proposed solution does not appear to be scalable.

Concerning PPML techniques [34], most of these schemes protect sensitive user data during model training and testing. However, PII is usually stored on cloud platforms. In addition, non-privacy-aware ML algorithms are also being widely used. Existing regulations on the subject may require organizations to notify users that they are collecting private data. Consequently, data owners can opt out, but it is a binary decision. Hence, there is a need for alternative PPML techniques that must be scalable to meet future computation and communication costs and emerging privacy requirements. Similarly, to justify the assumption of PPML techniques that the computation parties may not collude, different parties should perform different roles, such as SP or the PSP. The question remains: How do we trust someone with the PSP role? What is an ideal business model for a PSP?

The code analysis approaches [36] and [37] analyze the apps' binary or source code to detect the use of PII. However, these schemes are ineffective against dynamic code loading and reflection approaches. They also do not cater to the arbitrary identifiers generated by the apps. Consequently, the techniques involving the manipulation of OS and APIs [38] and [39] are only useful for rooted or jailbroken mobile devices. Similarly, PrivacyProxy [42] is inept against custom encryption algorithms, non-standard encodings, and indirectly tracking users and certificate-pinned apps. PrivySharing [45] is also a promising work that shares a particular data type on a specific blockchain channel. Consequently, a blockchain network may be divided into the desired number of channels. PrivySharing preserves users'

privacy, offers user-defined data access controls, rewards data owners for sharing their data, and ensures compliance with GDPR. However, it has no intrinsic key management mechanism for symmetric encryption and does not explicitly differentiate between PII and non-PII.

Today, PE is a buzzword in the information security community. It enforces adopting privacy measures to protect personal data during all stages of SDLC, including the design and development stages. However, a significant challenge in fully harnessing the potential of privacy and security engineering lies in the fact that software engineers typically need more time and autonomy to construct fundamental ethical systems successfully [114]. It should be noted that while privacy-by-design effectively outlines the necessary steps for protecting privacy, it needs to improve the efficiency of translating privacy requirements into actionable engineering activities [115].

As shown in Table. 4, we also evaluated the above-mentioned privacy-preserving techniques in the purview of some of the most common privacy requirements mandated by data protection regulations discussed in Section II. The ✓ indicates compliance with the privacy regulation, ✗ implies non-compliance and – shows that a particular requirement was not discussed in the respective solution/literature, and its compliance could not be determined. It can be observed that almost none of the existing techniques is developed to comply implicitly with all common data protection regulations. One reason could be the territorial jurisdiction of the privacy laws. Since implementing security based on the data protection-by-design principle always incurs costs overhead, the tech giants or software houses provide customized solutions per the customer or regional needs rather than implementing security as a standard. Therefore, data protection is always a weak link. Most existing solutions may apply some security and privacy controls but fail to ensure data protection throughout their life cycle. Consequently, they only meet some of the data security and privacy requirements.

VI. CURRENT CHALLENGES

The challenges perceived in designing and building privacy-preserving solutions include the following:-

a. **Building Privacy-Aware Database System**: A crucial aspect in developing a database solution that prioritizes privacy is automating various steps, in order to achieve a privacy-first approach through streamlined processes. One key component is the automation of schema analysis, where the selection of data fields for pseudonymization and determining the connections to be severed should be performed automatically. This automation relies on persistent analysis of the database schema and its data. Furthermore, an in-depth examination of the actual data and reconstruction of the underlying entity-relationship structure may be necessary to make decisions concerning PII, foreign keys, and approximate functional dependencies [43]. Furthermore, databases often consist of multiple distinct personal entities. For example, within a Transaction Processing Performance Council (TPC) Benchmark-H database, a separate employee relation necessitates the independent pseudonymization of associated personal data. Consequently, there is a need for comprehensive research to identify different types of entities present in a database and to understand their interrelationships.

b. **Enforcement/Implementation of Privacy Policies**: The enforcement and implementation of privacy policies pose significant challenges in the present scenario. Privacy policies typically define the data being shared, the associated rules or privacy guarantees, the intended data users, and the purpose of data usage. However, the effectiveness of policy implementation on the client side versus sending the policy along with the data to computation servers, where trust is required for adherence to policy rules, remains unclear. In addition, there may be scenarios when the same organization or entity controls both the party responsible for performing computations on data and the party receiving the results. The following trust issues may emerge in this context due to the inherent conflict of interest and potential data misuse.
   - **Potential for Bias**: When the same entity controls both the computation and the results parties, there is a risk of bias or favoritism in how the data is processed or interpreted. The entity may manipulate the computations or results to serve its interests, leading to skewed outcomes.
   - **Lack of Accountability**: With the same entity performing both roles, there might be a need for more independent supervision and accountability. If privacy breaches or policy violations occur during data processing, it could be challenging to hold the entity responsible, as there are no separate checks and balances in place.
   - **Conflicts of Interest**: The entity controlling both parties may have conflicting interests in data handling. For example, they may use the data to maximize their personal gains instead of using them for the original purpose. This can result in users' privacy rights violations.
   - **Misuse of Data**: When an organization controls both the computation and results functions, there is a risk of misusing the data beyond the scope defined by privacy policies. This can lead to unauthorized access, data breaches, or other privacy violations.
   - **Lack of Transparency**: Transparency is crucial in data processing, especially concerning sensitive or personal information. When both parties are under the same entity's control, they need to be more transparent about their data handling practices, which could lead to mistrust of data subjects.

c. **Practical Manifestation of Privacy Engineering**: No doubt privacy-by-design is a potent mechanism to preserve user privacy. However, there is a need to

TABLE 4. Compliance with privacy regulations.

Technique	Specific Focus	Consent Request	Transparent Process	Minimal Data Collection	Protection by Design	Multifactor Authentication	Role-based Access Control	Anonymized Data	Data Subjects' Rights	Notify Users
Intelligent healthcare [30]	HIPAA	✓	✗	✗	✗	✓	✓	✓	✗	✓
Blockchain for healthcare [31]	GDPR & HIPPA	✓	✓	✗	✓	✗	✓	✓	✓	✗
I-AM [33]	–	✓	–	–	✓	–	–	✓	Partial	✓
Privacy-preserving ML [34]	–	✓	✗	✗	✓	✗	✓	✓	✗	✗
Carbyne Stack [35]	GDPR & CCPA	✓	–	✗	✓	–	–	✓	✓	–
Code Analysis [36, 37]	✗	–	–	–	–	–	–	–	✗	✓
Manipulating OS and APIs [38, 39]	✗	✗	✓	✗	✗	–	–	✓	✗	✓
Network Flow Analysis [40, 41]	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓
PrivacyProxy [42]	✗	–	✗	✗	✓	✗	✗	✓	✓	✓
SYPSE [43]	GDPR & CCPA	✗	✗	✗	Partial	✗	✗	✓	✓	✗
GDPR-compliant PII management [44]	GDPR	✓	✓	✗	✗	✗	✗	✗	✓	✗
PrivySharing [45]	GDPR	✓	✓	✓	Partial	✗	✗	✗	✓	–
moocRP [46]	✗	✗	✓	–	✓	✗	✓	✓	✓	–

develop an approach to convert privacy-by-design and GDPR guidelines into engineering activities within SDLC or other system development life cycles [100].

d. **Development of an Integrated Privacy Protection**: After conducting a thorough gap analysis in Section V, it has been determined that none of the existing solutions ensures data privacy across the entire data life cycle. This encompasses all stages, from data acquisition to presenting analytics-driven query responses to users.

VII. A WAY FORWARD

There is a requirement for an integrated framework to ensure end-to-end privacy protection by design based on user preferences. Data owners need to know what information can

be collected by an app and how that information will be processed. Users must always be in the picture of who has access to their data, for what purpose data is being used, and for how long it will be used. The users are also concerned about how their data will be disposed of once not required by the app/service provider.

Fig. 15 provides an overview of the various facets encompassing a comprehensive solution to safeguard user privacy. The proposed solution is a generalized one and can be tailored according to the specific needs of an application. The term "end-to-end" denotes all phases of the data life cycle, including data acquisition, transmission, storage, processing, and the provision of business analytics. Focusing on the data acquisition phase, notable challenges include edge device security, data integrity, excessive data collection, and adherence to users' consent. Addressing these challenges requires adopting a robust trust management framework specifically tailored for IoT devices (serving as data sources) to uphold the security and integrity of the said devices.

In the realm of user privacy and protection against excessive data collection by applications, researchers proposed a smart human-computer interface (HCI) [116] and [117]. The HCI plays a crucial role in ensuring that users are fully informed about an app's privacy policy, the types of sensitive data it will collect and process, and the measures taken to secure the data, all before app installation. Should a user decline access to certain private data, the HCI must actively monitor and generate alerts for any violations. However, app developers must devise a method for showing app permissions and related alerts that do not compromise functionality [118]. Furthermore, the user interface should incorporate a scalable notification mechanism to enable users to take preemptive measures. For example, mitigation options should be appropriately presented based on the level of privacy risk involved. Like the alert interface, the mitigation interface should improve itself based on the user's previous choices and may offer options to configure automatic mitigation action plans for specific scenarios.

To ensure data privacy during transmission and storage, it is essential to segregate PII from non-PII and share the classified data with the requisite security measures. In this regard, PrivySharing [45] can be further improved to process PII and non-PII on separate blockchain channels. Moreover, for enhanced security, PII can be secured using LDP, lightweight homomorphic encryption, or CP-ABE. We recommend LDP and homomorphic encryption so that during the training and analytics stages desired value should be attained from the collected data while preserving user privacy. In addition, user-defined access control rules can be integrated into the blockchain smart contracts to realize data owners' consent for data sharing. The appropriate data classification and segregation measures, along with the access control rules embedded in smart contracts, will surely protect against PII leakage and unauthorized sharing without users' consent.

Subsequently, PPML techniques, such as Federated Learning (FL), can be employed during semantic labeling and training of ML models on data sets. One such technology to watch is Carbyne Stack [35], which uses open source MPC to offload computations to multi-party virtual clouds. Regardless of the underlying platform or technology, the adaptability of PPML techniques may require regular remodeling to accommodate emerging advances [119]. It should be noted that many PPML techniques introduce additional communication and computation overheads, which may impede the efficient utilization of the vast amounts of available data. Moreover, there should be a mechanism to manage trust among FL nodes to protect against maliciously/erroneously trained models. This can be realized by running every virtual cloud instance (as in Carbyne Stack) in a trusted execution environment. In addition, we also need statistical disclosure control and protection against unauthorized sharing of processed data.

In PPML, an essential requirement is a provision for data owners to exercise complete opt out capabilities for data collection or selectively revoke permission to collect specific PII. In scenarios where user data are deleted or unauthorized, ML models must have the adaptability to prevent the need for retraining. For example, researchers in [120] proposed a machine unlearning algorithm that incrementally unlearns without requiring the entire training process to start anew. Such techniques hold promise in achieving "right to be forgotten" as mandated by the GDPR. Additionally, the privacy model should be capable of generating privacy-preserving responses to address user queries. The queries or responses can be communicated to the end users in a privacy-preserving manner using CP-ABE.

It is highly recommended that the concept of PE be incorporated throughout every stage of the system/software development life cycle, including the design phase. Additionally, it is advisable to support the end-to-end privacy-preserving solution with blockchain or a distributed ledger technology (DLT), as it offers transparency and facilitates compliance with GDPR and other pertinent anti-trust laws. There may be questions about the proposed approach's scalability, computation overhead, and feasibility. It is imperative to note that not every distributed ledger is a blockchain. Hence, a permissioned DLT such as Corda [121] can be used to avoid the scalability issues of a typical blockchain technology. Corda allows enterprises to customize the network topology per their business objectives and security and privacy requirements. Unlike the broadcast of transactions in blockchain or other DLTs, Corda implements a peer-to-peer (P2P) transaction model, in which a transaction is only shared with the concerned nodes. For example, if Alice shares her location information with Bob, the transaction will only be sent to Bob. No other node in the network needs to be aware of this transaction. Hence, all nodes in a Corda network store only the transactions directly shared with them by other nodes. Consequently, the nodes avoid storing transaction data that are not related to

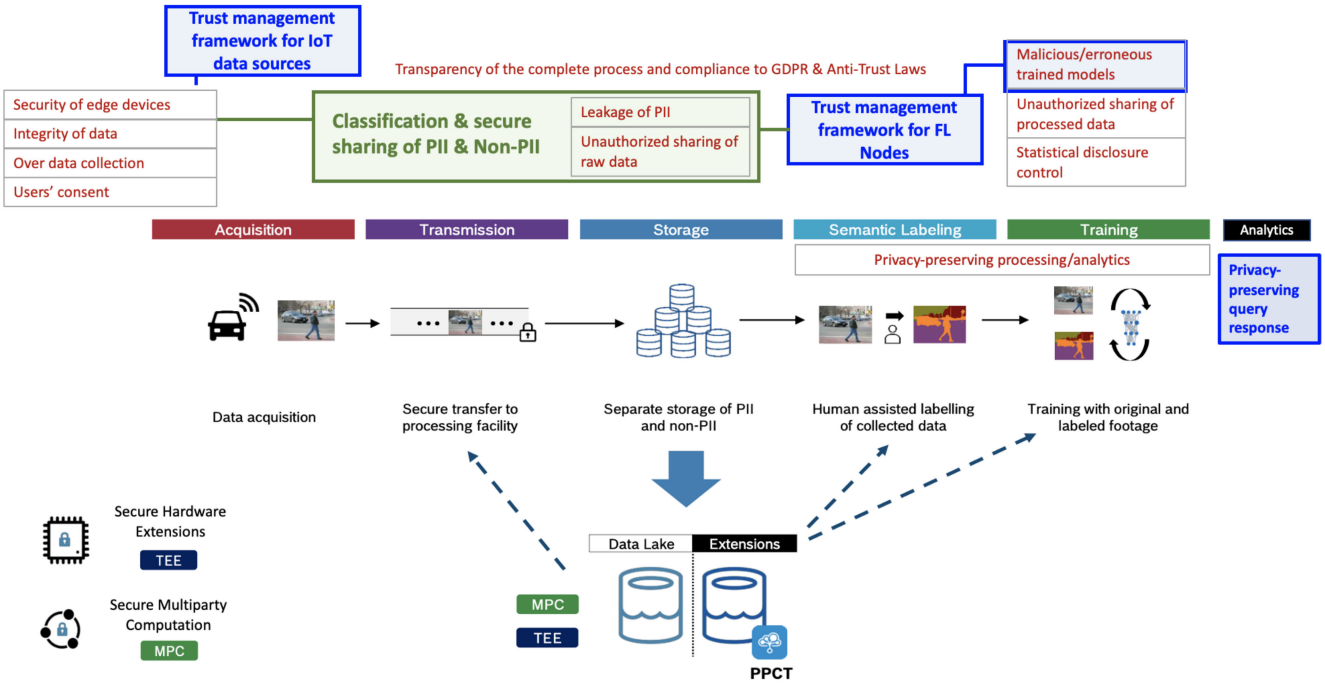

FIGURE 15. End-to-end protection of users' privacy.

TABLE 5. Data security, privacy, and performance issues.

Concerns	Proposed Methodology
Unauthorized data collection	Data owner's consent for data collection
Over-data collection	Data subject to select data assets for sharing
Data misuse	User-defined access control or RBAC
Unauthorized data deletion	Only data owner to delete his digital/data assets
Transparency of the complete process	Decentralized Autonomous Organization (DAO)
Data forgery and injection	DLT (Corda)
Transparency in data collection and sharing	Smart contracts
Compliance to privacy regulations	Smart contracts
Security of PII during storage and processing	Semi-homomorphic encryption (Paillier Transform)
Security of transactions during transmission	AMQP over TLS
Disclosure of user ID	Processing of anonymized data
Privacy-preserving computations	FL, SMPC
Erroneous/malicious model training	Trust management among FL nodes
Scalability (with regard to the number of network nodes and the increase in ledger size)	P2P DLT (Corda)
Users' right to be forgotten	Machine unlearning algorithm, data owners to delete their unwanted data assets
Privacy leakage in query-response	Generation of privacy-preserving responses secured by CP-ABE
Privacy-by-design	Use PE at every stage

them. Similarly, P2P transaction communication in Corda ensures low communication complexity compared to the broadcast nature of various blockchain/DLT technologies such as Ethereum, Bitcoin, and IoTA.

Concerning computation overheads of processing encrypted data, instead of fully homomorphic encryption, semi-homomorphic encryption such as Paillier Transform can be used. The Paillier Transform preserves user privacy by facilitating limited analytics based on addition operations on the encrypted data. Regarding security of transactions during transmission, Corda uses the Advanced Message Queuing Protocol (AMQP) over the TLS protocol. AMQP performs exceptionally well under high load conditions and ensures reliable message delivery. In addition, TLS guarantees data

confidentiality and integrity during transmission. Table. 5, presents possible methodology to resolve various data security, privacy and performance issues in the proposed model.

VIII. CONCLUSION

Significant reliance on Big Data as a strategic and economic asset has placed users at the forefront of privacy threats. Tech companies and government organizations compete to access, store, and utilize user data to achieve business or strategic goals without giving much importance to data owners' privacy. Therefore, a need was felt to review current SOTA privacy-preserving data processing techniques to identify their weaknesses and strengths, including compliance with privacy regulations. Every scheme has a unique way of processing PII and non-PII. However, most reviewed methods do not specify the exact mechanism and technical details of segregating PII and non-PII. Moreover, existing solutions offer fewer privacy guarantees with significant performance overheads. Consequently, instead of transparently implementing data privacy/security by design, the data owners are assumed to make the right choice or configure the desired privacy settings. Similarly, smart analytics providers employing PPML techniques may store PII in trusted data lakes or warehouses.

In general, there needs to be more awareness among system/software developers regarding PE practices/principles. Hence, more research must be done to devise a mechanism to translate the privacy requirements per GDPR, anti-trust laws, and users' preferences into engineering activities. In this way, system/software developers would be clear about the technical aspects of engineering privacy-by-design apps. In summary, numerous challenges must be overcome to claim a complete privacy-aware system/application. Key issues include building a privacy-aware database system, technical translation of privacy policies, and resolving trust issues related to ML/PPML models. Consequently, based on a thorough gap analysis, we propose a possible way to help develop an end-to-end, transparent, privacy-preserving data processing framework that can be employed in diverse applications. The proposed approach aims to empower data owners to make an informed decision to secure their data and privacy at any stage of the data lifecycle, from data acquisition to transmission, storage, processing, and query-response phase.

REFERENCES

[1] Y. Li and X. Yan, "Analyze the protection of personal data in the big data environment from the perspective of the enterprise," in *Proc. Int. Conf. Social Sci. Econ. Develop. (ICSSED)*. Wuhan, China: Atlantis Press, 2022, pp. 66–71.

[2] B. Jacquelyn. (2022). *How Much Data is Created Every Day in 2020*. Accessed: Jan. 10, 2024. [Online]. Available: https://techjury.net/blog/how-much-data-is-created-every-day/

[3] C. L. Goi, "The dark side of customer analytics: The ethics of retailing," *Asian J. Bus. Ethics*, vol. 10, no. 2, pp. 411–423, Nov. 2021.

[4] M. A. Hossain, S. Akter, and V. Yanamandram, "Revisiting customer analytics capability for data-driven retailing," *J. Retailing Consum. Services*, vol. 56, Sep. 2020, Art. no. 102187.

[5] B. Kitchens, D. Dobolyi, J. Li, and A. Abbasi, "Advanced customer analytics: Strategic value through integration of relationship-oriented big data," *J. Manage. Inf. Syst.*, vol. 35, no. 2, pp. 540–574, May 2018, doi: 10.1080/07421222.2018.1451957.

[6] M. Alexandra and G. Ben. (2019). *Facebook Understood How Dangerous the Trump-Linked Data Firm Cambridge Analytica Could be Much Earlier Than it Previously Said. Here's Everything That's Happened Up Until Now*. Accessed: Jan. 10, 2024. [Online]. Available: https://www.businessinsider.com/cambridge-analytica-a-guide-to-the-trump-linked-data-firm-that-harvested-50-million-facebook-profiles-2018-3

[7] (2022). *Australia Blames Cyber Criminals in Russia for Medibank Data Breach*. Accessed: Jan. 10, 2024. [Online]. Available: https://edition.cnn.com/2022/11/11/tech/medibank-australia-ransomware-attack-intl-hnk/index.html

[8] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[9] L. Guan, J. Xu, S. Wang, X. Xing, L. Lin, H. Huang, P. Liu, and W. Lee, "From physical to cyber: Escalating protection for personalized auto insurance," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM*, Stanford, CA, USA, Nov. 2016, pp. 42–55, doi: 10.1145/2994551.2994573.

[10] L. Yang, A. Humayed, and F. Li, "A multi-cloud based privacy-preserving data publishing scheme for the Internet of Things," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl. (ACSAC)*, Los Angeles, CA, USA, Dec. 2016, pp. 30–39, doi: 10.1145/2991079.2991127.

[11] X. Zhang, X. Wang, R. Slavin, T. Breaux, and J. Niu, "How does misconfiguration of analytic services compromise mobile privacy?" in *Proc. IEEE/ACM 42nd Int. Conf. Softw. Eng. (ICSE)*, Seoul, South Korea, Oct. 2020, pp. 1572–1583, doi: 10.1145/3377811.3380401.

[12] T. Taylor and T. Craig. (2018). *Marriott Discloses Massive Data Breach Affecting Up to 500 Million Guests*. Accessed: Jan. 10, 2024. [Online]. Available: https://www.washingtonpost.com/business/2018/11/30/marriott-discloses-massive-data-breach-impacting-million-guests/

[13] M. Finck and F. Pallas, "They who must not be identified–distinguishing personal from non-personal data under the GDPR," *SSRN Electron. J.*, vol. 10, no. 1, pp. 11–36, 2020.

[14] K. El Emam and C. Alvarez, "A critical appraisal of the article 29 working party opinion 05/2014 on data anonymization techniques," *Int. Data Privacy Law*, vol. 5, no. 1, pp. 73–87, Feb. 2015, doi: 10.1093/idpl/ipu033.

[15] S. Spiekermann, "The challenges of privacy by design," *Commun. ACM*, vol. 55, no. 7, pp. 38–40, Jul. 2012, doi: 10.1145/2209249.2209263.

[16] C. J. Hoofnagle, B. van der Sloot, and F. Z. Borgesius, "The European union general data protection regulation: What it is and what it means," *Inf. Commun. Technol. Law*, vol. 28, no. 1, pp. 65–98, Feb. 2019.

[17] X. Junke and T. Ying, "Legal protection of personal data in China," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput. (DASC)*, Calgary, AB, Canada, Oct. 2021, pp. 837–842.

[18] (2023). *Digital Personal Data Protection Bill 2022*. Accessed: Jan. 17, 2024. [Online]. Available: https://kpmg.com/in/en/home/insights/2022/12/privacy-digital-personal-data-protection-bill-2022.html

[19] E. L. Harding, J. J. Vanto, R. Clark, L. H. Ji, and S. C. Ainsworth, "Understanding the scope and impact of the California consumer privacy act of 2018," *J. Data Protection Privacy*, vol. 2, no. 3, pp. 234–253, Jan. 2019.

[20] T. Taylor and T. Craig. (2021). *Australian Privacy Act—1988*. Accessed: Jan. 11, 2024. [Online]. Available: https://www.ag.gov.au/rights-and-protections/privacy

[21] D. Sam, K. Nithya, S. D. Kanmani, A. Sheeba, A. S. Ebenezer, B. U. Maheswari, and J. D. Amesh, "Survey of risks and threats in online learning applications," in *Secure Data Management for Online Learning Applications*. Boca Raton, FL, USA: CRC Press, Apr. 2023, pp. 31–47.

[22] M. M. Ogonji, G. Okeyo, and J. M. Wafula, "A survey on privacy and security of Internet of Things," *Comput. Sci. Rev.*, vol. 38, Nov. 2020, Art. no. 100312. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574013720304123

[23] A. A. A. Sen, F. A. Eassa, K. Jambi, and M. Yamin, "Preserving privacy in Internet of Things: A survey," *Int. J. Inf. Technol.*, vol. 10, pp. 189–200, Feb. 2018.

[24] K. Abouelmehdi, A. Beni-Hessane, and H. Khaloufi, "Big healthcare data: Preserving security and privacy," *J. Big Data*, vol. 5, no. 1, pp. 1–18, Jan. 2018.

[25] M. A. Sahi, H. Abbas, K. Saleem, X. Yang, A. Derhab, M. A. Orgun, W. Iqbal, I. Rashid, and A. Yaseen, "Privacy preservation in e-healthcare environments: State of the art and future directions," *IEEE Access*, vol. 6, pp. 464–478, 2018.

[26] (2023). *Internet of Things (IoT) Market*. Accessed: Jan. 21, 2024. [Online]. Available: https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307

[27] W. Ben. (2023). *What is GDPR, the Eu's New Data Protection Law*. Accessed: Jan. 15, 2024. [Online]. Available: https://gdpr.eu/what-is-gdpr/

[28] (2023). *Data Protection Laws of China*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.dlapiperdataprotection.com/index.html?c=CN&t=law

[29] B. Rob. (2023). *California Consumer Privacy Act*. Accessed: Jan. 17, 2024. [Online]. Available: https://oag.ca.gov/privacy/ccpa

[30] K. Kurapati, "Proactive and intelligent healthcare management using IoT," in *Proc. Int. Conf. Adv. Comput., Commun. Appl. Informat. (ACCAI)*, Chennai, India, Jan. 2022, pp. 1–7.

[31] H. S. Jennath, V. S. Anoop, and S. Asharaf, "Blockchain for healthcare: Securing patient data and enabling trusted artificial intelligence," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 6, no. 3, p. 15, 2020, doi: 10.9781/ijimai.2020.07.002.

[32] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A conundrum of permissions: Installing applications on an Android smartphone," in *Proc. Int. Conf. Financial Cryptography Data Security*. Berlin, Germany: Springer, 2012, pp. 68–79.

[33] I. Wagner, Y. He, D. Rosenberg, and H. Janicke, "User interface design for privacy awareness in eHealth technologies," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2016, pp. 38–43.

[34] M. Al-Rubaie and J. M. Chang, "Privacy-preserving machine learning: Threats and solutions," *IEEE Security Privacy*, vol. 17, no. 2, pp. 49–58, Mar. 2019.

[35] (2022). *Carbyne Stack*. Accessed: Jan. 12, 2024. [Online]. Available: https://carbynestack.io

[36] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel, "FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps," in *Proc. 35th ACM SIGPLAN Conf. Program. Lang. Design Implement.*, Jun. 2014, pp. 259–269, doi: 10.1145/2594291.2594299.

[37] M. I. Gordon, D. Kim, J. Perkins, L. Gilham, N. Nguyen, and M. Rinard, "Information-flow analysis of Android applications in DroidSafe," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2015, vol. 15, no. 201, pp. 1–16.

[38] S. Chitkara, N. Gothoskar, S. Harish, J. I. Hong, and Y. Agarwal, "Does this app really need my location? Context-aware privacy management for smartphones," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–22, Sep. 2017, doi: 10.1145/3132029.

[39] M. Bokhorst. (2017). *XPrivacy—The Ultimate, Yet Easy to Use, Privacy Manager*. Accessed: Jan. 11, 2024. [Online]. Available: https://github.com/M66B/XPrivacy

[40] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, "Privacy oracle: A system for finding application leaks with black box differential testing," in *Proc. 15th ACM Conf. Comput. Commun. Security*, Alexandria, VA, USA, Oct. 2008, pp. 279–288, doi: 10.1145/1455770.1455806.

[41] A. Continella, Y. Fratantonio, M. Lindorfer, A. Puccetti, A. Zand, C. Kruegel, and G. Vigna, "Obfuscation-resilient privacy leak detection for mobile apps through differential analysis," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2017, pp. 10–14722.

[42] G. Srivastava, K. Bhuwalka, S. Kumar Sahoo, S. Chitkara, K. Ku, M. Fredrikson, J. Hong, and Y. Agarwal, "PrivacyProxy: Leveraging crowdsourcing and in situ traffic analysis to detect and mitigate information leakage," 2018, *arXiv:1708.06384*.

[43] A. Deshpande, "Sypse: Privacy-first data management through pseudonymization and partitioning," in *Proc. ACM 11th Annu. Conf. Innov. Data Syst. Res. (CIDR)*, Chaminade, CA, USA, 2021, pp. 1–8.

[44] N. Al-Zaben, M. M. Hassan Onik, J. Yang, N.-Y. Lee, and C.-S. Kim, "General data protection regulation complied blockchain architecture for personally identifiable information management," in *Proc. Int. Conf. Comput., Electron. Commun. Eng. (iCCECE)*, Aug. 2018, pp. 77–82.

[45] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "PrivyShare: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Comput. Secur.*, vol. 88, Jan. 2020, Art. no. 101653.

[46] Z. A. Pardos and K. Kao, "MoocRP: An open-source analytics platform," in *Proc. 2nd ACM Conf. Learn. Scale*, Vancouver, BC, Canada, Mar. 2015, pp. 103–110, doi: 10.1145/2724660.2724683.

[47] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Jose, CA, USA, May 2017, pp. 3–18.

[48] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–33, Dec. 2017, doi: 10.1145/3124441.

[49] M. A. Will and R. K. Ko, "A guide to homomorphic encryption," in *The Cloud Security Ecosystem*. Boston, MA, USA: Syngress, 2015, ch. 5, pp. 101–127. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128015957000057

[50] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 1053–1066, Jun. 2012.

[51] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in *Proc. 20th USENIX Secur. Symp. (USENIX Secur.)*, San Francisco, CA, USA, 2011, pp. 1–16. [Online]. Available: https://www.usenix.org/conference/usenix-security-11/faster-secure-two-party-computation-using-garbled-circuits

[52] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Commun. ACM*, vol. 28, no. 6, pp. 637–647, Jun. 1985, doi: 10.1145/3812.3818.

[53] V. K. Yadav, N. Andola, S. Verma, and S. Venkatesan, "A survey of oblivious transfer protocol," *ACM Comput. Surv.*, vol. 54, no. 10s, pp. 1–37, Jan. 2022, doi: 10.1145/3503045.

[54] L.-J. Pang and Y.-M. Wang, "A new (t, n) multi-secret sharing scheme based on Shamir's secret sharing," *Appl. Math. Comput.*, vol. 167, no. 2, pp. 840–848, Aug. 2005.

[55] D. Bogdanov, L. Kamm, S. Laur, and V. Sokk, "Implementation and evaluation of an algorithm for cryptographically private principal component analysis on genomic data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 5, pp. 1427–1432, Sep. 2018.

[56] Cybernetica. (2022). *Sharemind*. Accessed: Jan. 15, 2024. [Online]. Available: https://sharemind.cyber.ee

[57] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct. 2017, pp. 1175–1191, doi: 10.1145/3133956.3133982.

[58] V. Costan, I. Lebedev, and S. Devadas, "Secure processors—Part I: Background, taxonomy for secure enclaves and Intel SGX architecture," *Found. Trends Electron. Design Autom.*, vol. 11, nos. 1–2, pp. 1–248, 2017.

[59] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, "Oblivious multi-party machine learning on trusted processors," in *Proc. 25th USENIX Secur. Symp. (USENIX Secur.)*, Austin, TX, USA, 2016, pp. 619–636. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/ohrimenko

[60] C. Dwork, K. Talwar, A. Thakurta, and L. Zhang, "Analyze gauss: Optimal bounds for privacy-preserving principal component analysis," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, New York, NY, USA, May 2014, pp. 11–20, doi: 10.1145/2591796.2591883.

[61] M. Hardt and E. Price, "The noisy power method: A meta algorithm with applications," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds., Red Hook, NY, USA: Curran Associates, 2014, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2014/file/729c68884bd359ade15d5f163166738a-Paper.pdf

[62] M. Abadi, A. Chu, I. Goodfellow, H. B. Mcmahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Vienna, Austria, Oct. 2016, pp. 308–318, doi: 10.1145/2976749.2978318.

[63] K. Chaudhuri, A. D. Sarwate, and K. Sinha, "A near-optimal algorithm for differentially-private principal components," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2905–2943, Jan. 2013.

[64] R. Iyengar, J. P. Near, D. Song, O. Thakkar, A. Thakurta, and L. Wang, "Towards practical differentially private convex optimization," in *Proc. IEEE Symp. Security Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 299–316.

[65] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," in *Proc. 25th Annu. Conf. Learn. Theory*, in Proceedings of Machine Learning Research, vol. 23, S. Mannor, N. Srebro, and R. C. Williamson, Eds., Edinburgh, Scotland: PMLR, Jun. 2012, pp. 25.1–25.40. [Online]. Available: https://proceedings.mlr.press/v23/kifer12.html

[66] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *J. Amer. Stat. Assoc.*, vol. 60, no. 309, pp. 63–69, Mar. 1965, doi: 10.1080/01621459.1965.10480775.

[67] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Scottsdale, AZ, USA, Nov. 2014, pp. 1054–1067, doi: 10.1145/2660267.2660348.

[68] B. Cyphers and K. Veeramachaneni, "AnonML: Locally private machine learning over a network of peers," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Tokyo, Japan, Oct. 2017, pp. 549–560.

[69] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 92–106, Jan. 2006.

[70] X. Jiang, Z. Ji, S. Wang, N. Mohammed, S. Cheng, and L. Ohno-Machado, "Differential-private data publishing through component analysis," *Trans. Data Privacy*, vol. 6, no. 1, p. 19, Apr. 2013.

[71] S. Y. Kung, "Compressive privacy: From information/estimation theory to machine learning [lecture notes]," *IEEE Signal Process. Mag.*, vol. 34, no. 1, pp. 94–112, Jan. 2017.

[72] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proc. Cryptol. Conf. (CRYPTO)*, R. Safavi-Naini and R. Canetti, Eds., Santa Barbara, CA, USA: Springer, 2012, pp. 643–662.

[73] I. Damgård, K. Damgård, K. Nielsen, P. S. Nordholt, and T. Toft, "Confidential benchmarking based on multiparty computation," in *Proc. Financial Cryptogr. Data Security*, J. Grossklags and B. Preneel, Eds., Christ Church, Barbados: Springer, 2017, pp. 169–187.

[74] (2022). *Istio*. Accessed: Jan. 12, 2024. [Online]. Available: https://istio.io

[75] (2022). *Knative*. Accessed: Jan. 13, 2024. [Online]. Available: https://knative.dev/docs/

[76] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes, "ReCon: Revealing and controlling PII leaks in mobile network traffic," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Services*, Singapore, Jun. 2016, pp. 361–374, doi: 10.1145/2906388.2906392.

[77] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill, "Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem," in *Proc. 25th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, 2018, pp. 1–15.

[78] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, "Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings," in *Proc. 10th Symp. Usable Privacy Secur. (SOUPS)*. Menlo Park, CA, USA: USENIX Association, 2014, pp. 199–212. [Online]. Available: https://www.usenix.org/conference/soups2014/proceedings/presentation/lin

[79] C. Gibler, J. Crussell, J. Erickson, and H. Chen, "Androidleaks: Automatically detecting potential privacy leaks in Android applications on a large scale," in *Proc. Trust Trustworthy Comput.*, S. Katzenbeisser, E. Weippl, L. J. Camp, M. Volkamer, M. Reiter, and X. Zhang, Eds., Vienna, Austria: Springer, 2012, pp. 291–307.

[80] C. PrivacyGrade. (2015). *Grading the Privacy of Smartphone Apps*. Accessed: Jan. 11, 2024. [Online]. Available: http://privacygrade.org/

[81] H. Wang, Y. Li, Y. Guo, Y. Agarwal, and J. I. Hong, "Understanding the purpose of permission use in mobile apps," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, pp. 1–40, Jul. 2017, doi: 10.1145/3086677.

[82] M. Ahmad, V. Costamagna, B. Crispo, F. Bergadano, and Y. Zhauniarovich, "StaDART: Addressing the problem of dynamic code updates in the security analysis of Android applications," *J. Syst. Softw.*, vol. 159, Jan. 2020, Art. no. 110386.

[83] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "MockDroid: Trading privacy for application functionality on smartphones," in *Proc. 12th Workshop Mobile Comput. Syst. Appl. (HotMobile)*, Phoenix, AZ, USA, Mar. 2011, pp. 49–54, doi: 10.1145/2184489.2184500.

[84] W. Enck, P. Gilbert, S. Han, V. Tendulkar, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. Comput. Syst.*, vol. 32, no. 2, pp. 1–29, Jun. 2014, doi: 10.1145/2619091.

[85] K. Micinski, D. Votipka, R. Stevens, N. Kofinas, M. L. Mazurek, and J. S. Foster, "User interactions and permission use on android," in *Proc. CHI Conf. Human Factors Comput. Syst.* Denver, CO, USA: Association for Computing Machinery, May 2017, pp. 362–373.

[86] A. Le, J. Varmarken, S. Langhoff, A. Shuba, M. Gjoka, and A. Markopoulou, "AntMonitor: A system for monitoring from mobile devices," in *Proc. ACM SIGCOMM Workshop Crowdsourcing Crowdsharing Big (Internet) Data*, Aug. 2015, pp. 15–20, doi: 10.1145/2787394.2787396.

[87] Y. Song and U. Hengartner, "PrivacyGuard: A VPN-based platform to detect information leakage on Android devices," in *Proc. 5th Annu. ACM CCS Workshop Security Privacy Smartphones Mobile Devices*. Denver, CO, USA: Association for Computing Machinery, Oct. 2015, pp. 15–26, doi: 10.1145/2808117.2808120.

[88] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson, "Haystack: A multi-purpose mobile vantage point in user space," 2015, *arXiv:1510.01419*.

[89] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. IEEE Symp. Security Privacy (SP)*, Oakland, CA, USA, May 2008, pp. 111–125.

[90] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Sci. Rep.*, vol. 3, no. 1, pp. 1–5, Mar. 2013.

[91] J. Camenisch and A. Lehmann, "(Un)linkable pseudonyms for governmental databases," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Denver, CO, USA, 2015, pp. 1467–1479, doi: 10.1145/2810103.2813658.

[92] M. Hintze and K. El Emam, "Comparing the benefits of pseudonymisation and anonymisation under the GDPR," *J. Data Protection Privacy*, vol. 2, no. 2, pp. 145–158, Dec. 2018. [Online]. Available: https://www.ingentaconnect.com/content/hsp/jdpp/2018/00000002/00000002/art00005

[93] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "CryptDB: Protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Syst. Princ. (SOSP)*, Cascais, Portugal, Oct. 2011, pp. 85–100, doi: 10.1145/2043556.2043566.

[94] F. Schuster, M. Costa, C. Fournet, C. Gkantsidis, M. Peinado, G. Mainar-Ruiz, and M. Russinovich, "VC3: Trustworthy data analytics in the cloud using SGX," in *Proc. IEEE Symp. Security Privacy*, San Jose, CA, USA, May 2015, pp. 38–54.

[95] edX. (2022). *Start Learning From the World's Best Institutions*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.edx.org

[96] Coursera. (2022). *Learn Without Limits*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.coursera.org

[97] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Security Privacy*, Berkeley, CA, USA, May 2007, pp. 321–334.

[98] M. García, Y. Samuel, Á. Ramiro, and J. M. del, "A metamodel for privacy engineering methods," in *Proc. 3rd Int. Workshop Privacy Eng. Co-Located 38th IEEE Symp. Security Privacy*, vol. 1873, San Jose, CA, USA, May 2017, pp. 41–48.

[99] M. Hansen, M. Jensen, and M. Rost, "Protection goals for privacy engineering," in *Proc. IEEE Security Privacy Workshops*, San Jose, CA, USA, May 2015, pp. 159–166.

[100] Y. Al-Slais, "Privacy engineering methodologies: A survey," in *Proc. Int. Conf. Innov. Intell. Informat., Comput. Technol. (3ICT)*, Sakheer, Bahrain, Dec. 2020, pp. 1–6.

[101] (2022). *Internet Privacy Engineering Network—European Data Protection Supervisor*. Accessed: Jan. 13, 2024. [Online]. Available: https://edps.europa.eu/data-protection/ipen-internet-privacy-engineering-network_en

[102] S. Gutwirth, R. Leenes, P. De Hert, and Y. Poullet, *European Data Protection: Coming of Age*, vol. 16. Dordrecht, The Netherlands: Springer, 2013.

[103] A. Cavoukian. (2010). *The 7 Foundational Principles: Implementation and Mapping of Fair Information Practices*. Accessed: Jan. 13, 2024. [Online]. Available: https://gpsbydesigncentre.com/wp-content/uploads/2022/02/312239.pdf

[104] M. Alshammari and A. Simpson, "Towards an effective privacy impact and risk assessment methodology: Risk assessment," in *Proc. Trust, Privacy Security Digit. Bus.*, S. Furnell, H. Mouratidis, and G. Pernul, Eds., Regensburg, Germany: Springer, 2018, pp. 85–99.

[105] R. Gellert, "Understanding the notion of risk in the general data protection regulation," *Comput. Law Secur. Rev.*, vol. 34, no. 2, pp. 279–288, Apr. 2018.

[106] S. J. De and D. Le Métayer, "Priam: A privacy risk analysis methodology," in *Proc. Data Privacy Manage. Security Assurance*, G. Livraga, V. Torra, A. Aldini, F. Martinelli, and N. Suri, Eds., Heraklion, Greece. Cham, Switzerland: Springer, 2016, pp. 221–229.

[107] N. Shevchenko, T. A. Chick, P. O'Riordan, T. P. Scanlon, and C. Woody, "Threat modeling: A summary of available methods," Softw. Eng. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Jul. 2018, pp. 1–26. [Online]. Available: https://insights.sei.cmu.edu/documents/569/2018_019_001_524597.pdf

[108] J.-H. Hoepman, "Privacy design strategies," in *Proc. ICT Syst. Security Privacy Protection*, N. Cuppens-Boulahia, F. Cuppens, S. Jajodia, A. A. El Kalam, and T. Sans, Eds., Marrakech, Morocco: Springer, 2014, pp. 446–459.

[109] K. Beckers, S. Faßbender, M. Heisel, and R. Meis, "A problem-based approach for computer-aided privacy threat identification," in *Proc. Privacy Technol. Policy*, B. Preneel and D. Ikonomou, Eds., Limassol, Cyprus: Springer, 2014, pp. 1–16.

[110] R. Meis, "Problem-based consideration of privacy-relevant domain knowledge," in *Proc. Privacy Identity Manage. Emerg. Services Technol.*, M. Hansen, J.-H. Hoepman, R. Leenes, and D. Whitehouse, Eds., Nijmegen, The Netherlands: Springer, 2014, pp. 150–164.

[111] S. Abu-Nimeh and N. R. Mead, "Combining security and privacy in requirements engineering," in *Information Assurance and Security Technologies for Risk Assessment and Threat Management: Advances*. Hershey, PA, USA: IGI Global, 2012, pp. 273–290.

[112] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2009.

[113] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proc. 3rd ACM Cloud Comput. Secur. workshop (CCSW)*, Chicago, IL, USA, Oct. 2011, pp. 113–124, doi: 10.1145/2046660.2046682.

[114] S. Spiekermann, J. Korunovska, and M. Langheinrich, "Inside the organization: Why privacy and security engineering is a challenge for engineers," *Proc. IEEE*, vol. 107, no. 3, pp. 600–615, Mar. 2019.

[115] A. Ceross and A. Simpson, "Rethinking the proposition of privacy engineering," in *Proc. New Security Paradigms Workshop (NSPW)*, Windsor, U.K., Aug. 2018, pp. 89–102, doi: 10.1145/3285002.3285006.

[116] G. Iachello and J. Hong, "End-user privacy in human–computer interaction," *Found. Trends Human-Comput. Interact.*, vol. 1, no. 1, pp. 1–137, Oct. 2007, doi: 10.1561/1100000004.

[117] A. S. Patrick and S. Kenny, "From privacy legislation to interface design: Implementing information privacy in human–computer interactions," in *Proc. Int. Workshop Privacy Enhancing Technol.*, R. Dingledine, Ed., Dresden, Germany: Springer, Mar. 2003, pp. 107–124.

[118] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang, "Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, Pittsburgh, PA, USA, Sep. 2012, pp. 501–510, doi: 10.1145/2370216.2370290.

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

[120] Y. Cao and J. Yang, "Towards making systems forget with machine unlearning," in *Proc. IEEE Symp. Secur. Privacy*, San Jose, CA, USA, May 2015, pp. 463–480.

[121] *Corda*. Accessed: May 12, 2024. [Online]. Available: https://training.corda.net/corda-fundamentals/introduction/

IMRAN MAKHDOOM (Senior Member, IEEE) received the master's degree in information security from the National University of Sciences and Technology, Pakistan, in 2015, and the Ph.D. degree from the University of Technology Sydney, in 2020. He is a Postdoctoral Researcher with the University of Technology Sydney. He was a Food Agility Scholar, from 2019 to 2020, and has made a valuable contribution to data security and privacy in the Food Tech/Agri Tech. He has published several papers in some of the prestigious journals and conferences. His research interests include blockchain, the Internet of Things, distributed consensus, the networks, and computer security.

MEHRAN ABOLHASAN (Senior Member, IEEE) received the B.E. degree in computer engineering and the Ph.D. degree in telecommunications from the University of Wollongong, in 1999 and 2003, respectively. He has over 20 years of experience in research and development and serving in various research leadership roles. Some of these previous roles include, the Director of research programs with the Faculty of Engineering and IT; the Deputy Head of School for Research with the School of Electrical and Data Engineering; and the Laboratory Director of Telecommunication and IT Research Institute, University of Wollongong. He is currently a Leader of the Intelligent Networks and Applications Labortory, Global Big Data Technology Center, Faculty of Engineering and IT, University of Technology Sydney. He has authored more than 180 international publications and has won over seven million dollars in research funding. His current research interests include 5G/6G wireless networks, software-defined networking, tactile internet, intelligent transportation systems (ITS), the Internet of Things (IoT), wireless mesh, wireless body area networks, and sensor networks.

JUSTIN LIPMAN (Senior Member, IEEE) fosters innovation in connected technologies and thrives at the intersection of academia and industry. As an Industry Associate Professor and the Director of the RFCT Laboratory, UTS, he draws on more than 12 years of expertise leading research and development with Intel and Alcatel spearheading research. Previously, he was the Deputy Chief Scientist of the Food Agility Cooperative Research Center and the Director of research translation with UTS. Having secured more than 28M research funding, he drives innovation across RF, cybersecurity, the IoT, digital agriculture, smart cities, and data privacy. He actively shapes future connected systems through standards development. A dedicated bridge builder, his expertise fuels impactful industry collaborations translating research into real-world solutions, positioning him at the forefront of innovation. He holds 24 U.S. patents and has published more than 100 peer-reviewed papers in top-tier conferences and journals.

NEGIN SHARIATI (Senior Member, IEEE) received the Ph.D. degree in electrical-electronics and communication technologies from the Royal Melbourne Institute of Technology (RMIT), Melbourne, Australia, in 2016.

She is a Senior Lecturer with the School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney (UTS), Australia. She has established the State-of-the-Art RF and Communication Technologies (RFCT) Research Laboratory, UTS, in 2018, where she is currently the Co-Director and leads research and development in RF technologies, sustainable sensing, energy harvesting, low-power Internet of Things, and AgTech. She also leads the sensing innovations constellation with Food Agility Corporative Research Center (CRC), enabling new innovations in agriculture technologies by focusing on three key interrelated streams, such as sensing, energy, and connectivity. Since 2018, she has been held a joint appointment as a Senior Lecturer with Hokkaido University, externally engaged in research and teaching activities in Japan. She attracted more than 3M dollars' worth of research funding across a several CRC and industry projects, where she has taken the Lead Chief Investigator (CI) role and also contributed as a member of the CI Team. She was with industry as an Electrical and Electronics Engineer, from 2009 to 2012. Her research interests include RF-electronics circuits and systems, sensors, antennas, RF energy harvesting, simultaneous wireless information and power transfer, and wireless sensor networks. She was a recipient of the Outstanding Research Award, in 2021; the IoT Awards Australia; and the IEEE Victorian Section Best Research Paper Award, in 2015.

DANIEL FRANKLIN (Member, IEEE) received the Bachelor of Engineering degree (Hons.) in electrical from the University of Wollongong, in 1999, and the Ph.D. degree in telecommunications engineering, ''Enhancements to Channel Models, DMT Modulation and Coding for Channels Subject to Impulsive Noise,'' from the University of Wollongong, in 2007. He is currently a Senior Lecturer with the School of Electrical and Data Engineering, University of Technology Sydney. His current research and commercial interests are split between radiation engineering, including positron emission tomography, computed tomography, radiotherapy, radiation dosimetry, pharmacokinetic modeling, and telecommunications engineering, including security applications and network protocols, multimedia, image and signal processing, wired and wireless communication systems, and communications protocols.

MASSIMO PICCARDI (Senior Member, IEEE) is a Professor of natural language processing (NLP), computer vision, and machine learning with the Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), where he joined in 2002. At UTS, he is the Head of Discipline and SEDE Signal Processing and Analytics, and a Leader for the Big Data Analytics Program of the Global Big Data Technologies Center, an international center of excellence for the development of enabling technologies for big data. Over his career, he has been the author or co-author of over a 108 scientific papers in international journals and conference proceedings and several book chapters. Since relocating to Australia from Italy, in 2002, he has been the principal investigator of many advanced research projects including two Australian Research Council (ARC) Discovery Projects and an ARC Linkage Project, and a chief investigator in another Linkage Project and four Linkage Infrastructure Projects. He has received significant funding from several Cooperative Research Centers (CMCRC, FACRC, and DHCRC). He is a member of IEEE Computer and Systems, Man and Cybernetics Societies, the International Association for Pattern Recognition, and the Association for Computational Linguistics. He serves as an Associate Editor for IEEE TRANSACTIONS ON BIG DATA and a member of the editorial board for *Artificial Intelligence in Medicine*.