PACTA: An IoT Data Privacy Regulation Compliance Scheme Using TEE and Blockchain

Yongxin Zhang, Jiacheng Yang, Hong Lei, Zijian Bao, Ning Lu, Wenbo Shi, and Bangdao Chen

***Abstract*—Despite the existence of data privacy regulations, such as the general data protection regulation (GDPR), data leaks in the Internet of Things (IoT) still occur and cause significant harm due to the noncompliance of data users. To address this issue, a notable solution involves recording the process in an open, immutable blockchain and utilizing the trusted execution environment (TEE) for reliable compliance verification. Although substantial progress has been made in designing compliance schemes in recent years, current approaches suffer from various limitations, including compliance incompleteness, regulation faultiness, and privacy leak. This article introduces PACTA, an IoT data privacy regulation compliance scheme that leverages TEE and blockchain technology. In the protocol, PACTA efficiently handles both dynamic and static consent of data owners and utilizes TEE for compliance analysis of requests and processes. By storing encrypted critical data, the blockchain facilitates privacy-preserving audits of the entire compliance process. Additionally, we have designed a challenge–response protocol to address the silent behavior of the TEE. We demonstrate that PACTA effectively enforces regulation compliance while safeguarding privacy. We thoroughly evaluate our implementation's efficiency and effectiveness using Ethereum and Intel SGX platforms.**

***Index Terms*—Blockchain, compliance, general data protection regulation (GDPR), Internet of Things (IoT), trusted execution environment (TEE).**

I. INTRODUCTION

THE Internet of Things (IoT) technology has rapidly advanced, leading to an exponential growth in the quantity of IoT devices, generating massive amounts of data. As the reports forecast, there will be more than 29 billion IoT devices in 2030,[1] and the market size will be over 3,300 billion dollars.[2] Data sharing among users improves the value of data by enabling broader insights, enhancing accuracy and completeness, fostering collaboration and innovation, facilitating the development of data-driven solutions, and driving economic and social benefits [1], [2]. However, it is crucial to balance data sharing with privacy and security considerations, many countries have introduced data privacy protection regulations [3], [4]. The general data protection regulation (GDPR), the most stringent data protection regulation which took effect on May 25, 2018, has been instituted to provide individuals with greater control over their personal data and to promote responsible behavior by organizations in relation to that data.[3] Regardless of the geographic location of organizations, the GDPR applies to any entity that processes the personal data of individuals residing within the EU.

Noncompliance: Despite the ongoing legal and security improvements by both governments and businesses regarding data privacy and protection, data breaches still occur from time to time [5], [6], [7]. On July 28, 2022, IBM released a report *Cost of a data breach 2022* that notes the data breach has reached a high of $4.35M, raising broader concerns about data privacy. On April 15, 2022, a medical software firm fined €1.5M for leaking data of 490k patients, which violated Article 28, 29, and 32 of GDPR.[4] High-impact errors include unauthorized data handling and not storing data in ciphertext. The firm extracted a larger volume of data than required and processed them beyond the instructions given by the data controller (DC). The data is stored in plaintext in a

Manuscript received 29 June 2023; revised 25 September 2023; accepted 27 September 2023. Date of publication 2 October 2023; date of current version 21 February 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2700601; in part by the Finance Science and Technology Project of Hainan Province under Grant ZDKJ2020009; in part by the Hainan Province Science and Technology Special Fund under Grant GHYF2022010; in part by the National Natural Science Foundation of China under Grant 62163011, Grant 62072092, Grant 62072093, and Grant U1708262; in part by the Fundamental Research Funds for the Central Universities under Grant N2023020; in part by the Natural Science Foundation of Hebei Province under Grant F2020501013; in part by the China Postdoctoral Science Foundation under Grant 2019M653568; and in part by the Key Research and Development Project of Hebei Province under Grant 20310702D. *(Corresponding authors: Hong Lei; Zijian Bao.)*

Yongxin Zhang and Jiacheng Yang are with the R&D Department, SSC Holding Company Ltd., Chengmai 571924, China, and also with the Blockchain Research Group, Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China (e-mail: silence_yongxin@163.com; jiacheng.yang.work@gmail.com).

Hong Lei is with the School of Cyberspace Security (School of Cryptography), Hainan University, Haikou 570228, China, and also with the Blockchain Research Group, Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China (e-mail: leiluono1@163.com).

Zijian Bao is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072 China (e-mail: baozijian@whu.edu.cn).

Ning Lu is with the College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China, and also with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: luning@neuq.edu.cn).

Wenbo Shi is with the College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: shiwb@neuq.edu.cn).

Bangdao Chen is with the Blockchain Research Group, Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China (e-mail: bangdao.chen@gmail.com).

Digital Object Identifier 10.1109/JIOT.2023.3321308

[1] Number of IoT connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030.

[2] IoT Market Size.

[3] GDPR.

[4] Health data breach.

publicly accessible area of the server. In terms of people daily life, for example, in research of 532 IoT apps, nearly half of apps hide their data sharing practices with third parties in their privacy policies, and 63.4% of the data processes that involved health and wellness data was inconsistent with the practices disclosed in the privacy policies [8]. Compliance with regulations is not merely a question of lacking technical solutions that can adequately address data privacy requirements and provide the mandated mechanisms. Rather it is a challenge due to the fact that such technical solutions have been designed and implemented under the assumption of weak or unenforced censorship frameworks.

Previous Works: Recognizing the need for enforced regulatory mechanisms to support data privacy protection implementation, scholars have utilized blockchain smart contracts to enforce compliance verification and publicly store procedural data, including requests and compliance results [9], [10], [11], [12]. Blockchain transparency allows for the data it contains to be scrutinized and analyzed by anyone with access to it, providing a level of transparency and accountability that is unparalleled in traditional data management systems. The blockchain maintains an open and distributed ledger, recording all transactions and activities sequentially and permanently [13], [14]. Unlike restricted data management systems, the blockchain is accessible for individuals to independently examine and evaluate recorded data. Each transaction and modification is stored in a block, linked to previous blocks through cryptographic hashes, forming an unalterable chain of information. This structure allows participants to verify authenticity by reviewing the complete transaction history. Transparency fosters accountability, identifying and attributing malicious or fraudulent activities. However, the transparency of the blockchain poses a potential threat to privacy requirements, leading several studies to utilize trusted execution environments (TEEs) for facilitating confidential off-chain computation for blockchain [10], [11], [12]. These works involve executing smart contracts within a TEE and submitting transactions to these contracts. The TEE operator is unable to access any information contained within the TEE due to its physical isolation. This approach enhances the security and privacy of the blockchain network, as sensitive data is protected from unauthorized access. This also points the direction for privacy protection in regulation compliance.

Limitations: However, current solutions come with these limitations.

1) *Compliance Incompleteness:* Compliance completeness refers to ensuring compliance throughout the entire data usage life-cycle, including consent obtainment, data request, and data usage. However, existing works do not cover all three parts. They focus on obtaining consent, but either only take into account when the data owner (DO) is offline, e.g., the work [9] requires the DO to be online to reply requests while the work [15] can only deal with static DO preference to obtain the consent.
2) *Regulation Faultiness:* The purpose of regulation is to establish and enforce rules, standards, and guidelines that govern various aspects of society. It also includes the auditing for the compliance process. Based on its immutable properties, blockchain is a good choice as a trusted trace platform. However, existing blockchain-based compliance schemes only check either request or the data usage program and store part of the data, making it difficult for regulators to reconstruct the complete compliance inspection process, which is not conducive to auditing.
3) *Privacy Leaks:* Privacy and regulation can sometimes be in conflict due to the different goals and considerations they prioritize. As stated in the work [16], even anonymized data may reveal privacy. Some schemes directly put the requests or private requirements on the blockchain to enable accountability. However, it may reveal some sensitive information of both data users (DUs) and DOs.

Our Scheme: In this article, we propose PACTA, an IoT data privacy regulation compliance scheme using TEE and blockchain.[5] The proposed scheme considers both online and offline scenarios. When the DO is online, the cloud-based service provider (CSP) requests consent directly from the DO. However, in the DO's absence, the CSP makes an approval decision based on private requirements indicated in the system's smart contracts to complete the compliance. The TEE deployed by the CSP confidentially verifies compliance with both request and program. The proposed scheme maintains a balance between matters of privacy and regulations by storing policies, requests, and compliance results in ciphertext on the blockchain. Only the regulator can decrypt the compliance results. The scheme also deploys a TEE within the CSP. Finally, the scheme adopts a challenge–response protocol to handle malicious CSP behavior, while time constraints are strictly enforced.

Contributions: Our key contributions are as follows.

1) We present a customized data model. Compared with previous data models, our model takes into account both public and private requirements, while also considering the offline status of DOs, thereby achieving a higher level of compliance comprehensiveness.
2) We utilize TEE for conducting compliance analysis, which is the first approach to incorporate compliance checks for both requests and programs. This framework guarantees confidentiality while enabling a more extensive and thorough detection of compliance. Moreover, the encrypted compliance results are securely logged in the blockchain for regulatory purposes.
3) We design PACTA based on blockchain and TEE. To the best of our knowledge, PACTA is the first privacy regulation compliance scheme that supports comprehensive compliance verification while also striking a balance between privacy and regulation. Additionally, due to the large volume of IoT data, we have restricted the compliance check performed by the CSP. Furthermore, PACTA can be extended to other cloud application domains wherever compliance is necessary.

[5] *Pacta sunt servanda*, (Latin for "agreements must be kept"), is a fundamental rule of the traditional theories of civil law.

4) We conduct security analysis and experimental evaluation of our proposed scheme. PACTA enforces the regulation while protecting privacy. It can solve the malicious CSP by the challenge–response protocol. To illustrate the feasibility of our scheme, we implement the compliance programs using Intel SGX as the TEE and evaluated it on read data sets. The request check takes 21 μs, and the program check consumes 309 μs.

This article is organized as follows. Section II discusses the related work. Section III introduces the system models and assumptions. Section IV gives workflow and challenges. Section V presents the definitions and data model. Section VI details smart contracts and the protocol. Section VII includes the security analysis of PACTA. Section VIII gives the implementation of PACTA along with its evaluation. Section IX discusses the research scope of this article, the limitation of our scheme, and the future work. Section X gives the conclusion.

II. Related Work

A. General Data Protection Regulation

The GDPR aims to create uniform data privacy laws across the European Union, protecting the rights and interests of all EU citizens [9]. The core principles relating to the processing of personal data contain: lawfulness, fairness, and transparency; purpose limitation; data minimization; accuracy; storage limitations; and integrity and confidentiality [16]. The involved roles include data subject (DS), DC, and data processor (DP). The GDPR legislation ensures that the DS completely control over the data stored in the DC, and DC should protect the right of DS when DP processes the data. DS is able to consent and track all the activities on the data containing, what, who, when, how long, and how the data is processed [10], [11]. Only with consent can DC send the target data to DP for processing. The public supervisory authority (SA) is responsible for monitoring the GDPR compliance of roles.

B. Data Privacy Regulation Compliance Schemes

GDPR compliance is the process of ensuring that DC and DP are adhering to the requirements outlined in GDPR regulations. However, the regulations provide high-level guidelines rather than addressing detailed technical implementation [9], [17]. Hence, how to achieve regulation compliance with appropriate technical and organizational measures has become a hot topic. The blockchain technology has been applied to be GDPR-compliant to provide SA with auditable traceability through cryptographic methods and consensus algorithm. Truong et al. presented a solution for managing personal data in compliance with the GDPR, using blockchain technology. It allows for auditable traceability and transparency, and provides a detailed description of the proposed architecture and components. Crucially, this article argues that the use of blockchain technology could provide a GDPR-compliant solution for managing personal data, although there are practical challenges, such as trust and privacy, that need to be addressed to make it feasible [9]. Barati and Rana [11] explored the issues of ensuring compliance with the GDPR in cloud-based service delivery models. A TEE-based container in cloud is used to record all operations, and submits the record to the smart contract for compliance verification. However, the contract deals with the record in plaintext, which may leak DS privacy [16]. Barati et al. [10] proposed a scheme for auditing cloud services used in online healthcare systems to ensure compliance with the GDPR. The online healthcare systems often rely on cloud services to store and process sensitive patient data, which creates a risk of data breaches and other privacy violations. It also uses smart contract to verify the compliance of GDPR through the operation records. However, the data item is public, if the data item involves uric acid (UAC) indicators, the adversary may infer that the patient is undergoing a gout examination, and then carry out the attack. Yeh et al. [12] introduced a novel approach for DS to share files in a peer-to-peer (P2P) network, complying with GDPR. PrivGuard improves the compliance process productivity, using a static analyzer that examines the DU program in the TEE [15]. It focuses on the program compliance with a DS-customized preset privacy preference, ignoring the dynamic consent when DS is online. Ahmad et al. [8] proposed an automated analysis framework for IoT apps' codes and privacy policies to identify compliance gaps. A blockchain-based revocation mechanism is designed to allow DS to revoke access to their data at any time. Based on it, the DS controls over their data and can withdraw consent.

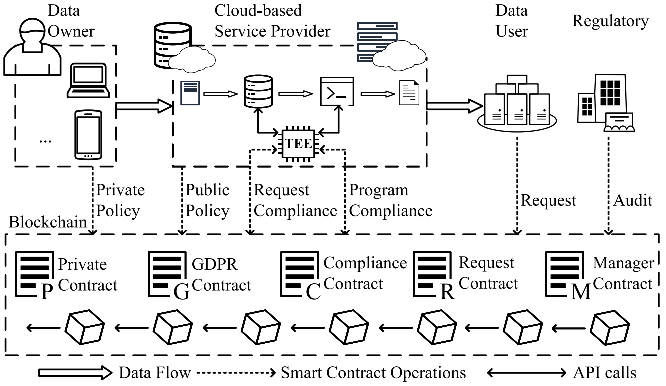


Fig. 1. System model.[6]

III. Models and Assumption

A. System Model and Assumptions

Given the immense amount of IoT data, a significant proportion of current solutions for sharing such data are cloud-based. Therefore, our research, in this article, focuses on ensuring regulation compliance during the process of sharing IoT data within cloud applications. Fig. 1 shows our system model, involving the following.

1) *DO:* The client who owns personal data. The DO allows the CSP to collect its data generated by various IoT devices and decides consent when DUs requiring. A DO is a DS in the context of GDPR.

[6] The figure is drawn with the Microsoft Visio, and the icons are from the Visio library.

2) *CSP:* The service provider collects DO data, employs cloud storage service to store data, uses *TEE-enabled* cloud computing service to process data-based business. The CSP may share DO data with DUs for its commercial interests. A CSP is a DC (sharing DO data with DU) and a DP (processing DO data for business) in the GDPR terminology.
3) *DU:* The DU is an entity that uses the DO data for its commercial purposes. The DU applies to the CSP for DO data or the CSP executes the program and returns the result if the DO consents. In the GDPR terminology, a DU is a DP.
4) *Blockchain Platform (BP):* The BP consists of nodes from all over the world, maintaining a tamper-proof evidence-perpetuation platform via a consensus algorithm. It supports Turing-complete smart contracts for automatic execution. The BP stores the compliance-related data for regulators to audit.
5) *Regulator:* The regulator is responsible for auditing the compliance of the whole process of data sharing through the data stored in the blockchain. A regulator plays the role of an SA in the GDPR terminology.

To model the behavior and the capabilities of every participant of the system, we make the following assumptions.

Blockchain: Our assumption for the blockchain's standard security properties includes the common prefix, chain quality, and chain growth [18], as per previous work [19], [20]. As such, valid transactions will appear in the blocks of the main chain after a certain duration. We use block height to represent the timestamp of data storage on the blockchain, as works [21], [22]. All the entities in our scheme interact with the blockchain, and we use the blockchain public key *pk* as a unique identifier for each entity. The blockchain smart contract verifies the validity of a signature when it receives a transaction, and the process is not detailed in this article.

TEE: We assume that the TEE provides *integrity* and *confidentiality* as guarantees for the program running inside the TEE enclave, in line with TEE-assisted blockchain schemes [19], [23], [24], [25], [26]. The TEE prevents tampering, ensures proper program execution, and does not compromise privacy. Although there are some attacks, such as side-channel and fault injection attacks [27], our scheme can mitigate these based on previous research [28], [29]. However, addressing these attacks is outside the scope of this article. We further assume that the TEE can support *secure remote attestation*, which requires providing a cryptographic proof to a third-party that a program is executing securely within the authentic enclave. Existing industrial TEEs currently support this feature [30], [31].

We emphasize this model can also work for other domains of cloud applications wherever compliance is necessary.

IV. Design

A. Architecture Overview

Our protocol is defined by a collection of phases as follows.

Initialization Phase: In this phase, all entities should generate their cryptographic parameters. They should generate their blockchain public/private keys (*pk*, *sk*) to interact with blockchain. Besides, the regulator will generate keys for them to encrypt and decrypt the information.

Data-Sharing Phase: A data-sharing process roughly contains the following steps. The DO stores data, the DU requests data, the CSP uses its TEE for compliance verification, and the regulator checks compliance results. ① The regulator deploys smart contracts. ② The CSP sets public policy in the GDPR contract (GC), and the DO sets private policy in the private contract (PC) in ciphertext. ③ The DO sends encrypted data to CSP via digital facility, such as smart phone, laptop, and other IoT devices. When DU wants to use the data stored in the CSP, a set of checks needs to be done. ④ The DU sends the request to the request contract (RC). ⑤ The CSP checks whether the request is compliant using TEE. The verification result will be put on the blockchain. ⑥ If compliant, the CSP verifies whether the program is compliant using TEE. The verification result will also be put on the blockchain. ⑦ If compliant, the execution result will be sent to DU.

Auditing Phase: The regulator checks the request and program for compliance, and obtains the compliance result in the contract.

B. Design Challenges

There are some challenges impeding our scheme.

1) *Obtaining DO's Consent:* Following the GDPR rules (Art.18), the personal data can only be processed with the DO's consent. Obtaining the DO consent as early as possible is an important goal of the DU. Some solutions require the DO to stay online. Other solutions use DO configured requirements regardless of whether they are online. This has limitations in the face of various requests. We made a tradeoff. When a request occurs, if the DO is online, the CSP should ask the DO to get consent, otherwise, the CSP uses TEE to execute programs with the DO requirements stored in blockchain. Whether consent comes from the user or the program is packaged in compliance results on the blockchain for auditing.
2) *Privacy Versus Regulation:* The openness, transparency, and tamper proof of blockchain naturally supports regulation. Regulators have easy access to trusted data to learn compliance results. However, data on the blockchain may leak DO's privacy, even solutions employ data items rather than data value [16]. For example, the UAC item hints that the DO might be a gout sufferer. To this end, we store DO requirements and DU requests on contracts in ciphertext. The TEE deployed in CSP runs compliance programs confidentially and sends results to the blockchain contracts for auditing.
3) *Malicious CSP:* The compliance verification program runs in the TEE enclave, which is under the control of CSP. Malicious CSP may modify or replay inputs, falsify outputs, procrastinate execution, and even turn off the TEE. We adopt some security mechanism to treat them. Digital signatures and random numbers

are used to resolve tampering and replay attacks. A challenge–response protocol is designed to deal with procrastination and outage.

V. Definitions and Notations

A. Cryptographic Primitives

Our protocol utilizes an asymmetric encryption scheme ($\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}$), a symmetric encryption scheme ($\mathsf{KGen}, \mathsf{En}, \mathsf{De}$), a signature scheme ($\mathsf{SKGen}, \mathsf{Sign}, \mathsf{Veri}$), and a secure hash function $H(\cdot)$. We identify parties by their public keys. The signing algorithm Sign takes (sk, m) as input, where m denotes the message. The algorithm outputs a signature σ. The verification algorithm Veri takes (σ, m, pk) as inputs and outputs $\top$ if the signature on m is valid and $\perp$ otherwise.

B. Data Model

This section introduces the data model for compliance in data processing. We consider DO private requirements expect the public regulation rules from a privacy perspective. Especially, we are concerned with the practice of the compliance for consent and differentiate between the online status and offline status of the DO.

One of the contracts on the blockchain is responsible to receive the requests from DUs. Requests expect personal data to execute operations for certain purposes. We can define the data usage model to formally constrain compliance with requests.

Definition 1: The data model of a request is a four-tuple: $\mathcal{Q} = \mathcal{S} \times \mathcal{D} \times \mathcal{O} \times \mathcal{P}$, where $\mathcal{Q}$ is a set of requests; $\mathcal{S}$ is a set of subjects (DOs); the set $\mathcal{D}$ contains personal data classes that represent the types of data rather than the values; $\mathcal{O}$ is a set of operations; $\mathcal{P}$ is a set of purposes.

This request refers to the panorama of data processing, i.e., who did what operation on what data and for what purpose. The set D specifies the data type, such as time, address, and role. This is enforced to protect privacy, as defined in GDPR Art.5(1)(f). Recall that the processing is based on the consent defined in GDPR Art.7.(1), to obtain the consent, it is encouraged to inform DU the GDPR compliance status of the request in advance. The following definition gives the compliance.

Definition 2: The compliance of the request is denoted by a boolean function: $\Upsilon_r : Q \to \{\top, \perp\}$, following the rules r. Given a request $req = \langle s, d, o, p\rangle \in \mathcal{Q}$, where $s \in \mathcal{S}$, $d \subseteq \mathcal{D}$, $o \in \mathcal{O}$, and $p \in \mathcal{P}$. A compliance result is defined by $\Upsilon_r(req)$.

Generally, the request $req = \langle s, d, o, p\rangle \in \mathcal{Q}$ states that the DU will execute o on data d of DO s for purpose p and that req is GDPR compliant if $\Upsilon(req) = \top$. While the GDPR is rigorous and broadly applicable, it is not customized to each DO and is not perfect for all scenarios. To this end, we further extend the scope of compliance by increasing the DO's private policies and taking into account the DO's consent processing both online and offline.

Definition 3: Let Υ_G be the boolean function for the GDPR compliance, as defined in Definition 2. Let Υ_{DO} be the boolean function for the DO private requirements compliance, as follows:

$$\Upsilon_{DO}(req) = \begin{cases} \delta \cap \Upsilon_r(req), & \text{online} \\ \Upsilon_r(req), & \text{offline} \end{cases}$$

where, $\delta \to \{\top, \perp\}$ denotes the decision of DO. A compliance result is defined by $\Upsilon(req) = \Upsilon_G(req) \cap \Upsilon_{DO}(req)$.

Given a request $req = \langle s, d, o, p\rangle$, req is compliant if $\Upsilon(req) = \top$. That means the req is GDPR-compliant and DO private requirements compliant. If the DO is offline, Υ_{DO} follows the rule r to determine compliance. If the DO is online, the DO decision δ and the rule r both work together.

Definition 4: The public policy and private policy is a four-tuple same to that of request. The public policy $\mathcal{PO} = \mathcal{S} \times \mathcal{D} \times \mathcal{O} \times \mathcal{P}$, where $\mathcal{S}$ indicates the set of DOs, $\mathcal{D}$ denotes the set of data classes, $\mathcal{O}$ implies the set of data operations, and $\mathcal{P}$ denotes the set of purposes. The $\mathcal{PO}$ contains the public polices, such as GDPR, and it is universal for all applications. The private policy $\mathcal{PP} = \mathcal{S}' \times \mathcal{D}' \times \mathcal{O}' \times \mathcal{P}'$. The $\mathcal{PP}$ contains the DO's private requirements, and it is application-oriented.

For example, a public policy $\mathcal{PO} = \{subjects\ of\ \texttt{PACTA}\} \times \{name, age, gender\} \times \{read, write\} \times \{commerce, research\}$. It permits that DUs can read and write the application PACTA users' name, age, and gender data for commercial or research purposes. The DO (Alice) can create a private policy $\mathcal{PP} = \{Alice\} \times \{name, age\} \times \{read\} \times \{research\}$ to restrain the range and operation of DUs. Within the application PACTA, the $\mathcal{PP}$ permits that the DUs can read Alice's name and age data for research purposes.

The data analysis program *dap* used by DU follow a certain paradigm.

Definition 5: The program is denoted by four modules: $dap = \langle \texttt{readCon}, \texttt{extractRow}, \texttt{extractColumn}, \texttt{calcData}\rangle$. readCon structures the data in certain form, such as table, to a data file *df*. extractRow extracts a row of data from *df* to construct a record *dr*. We say the *dr* indicates the data from a DO. extractColumn extracts some columns from *dr* to a narrow record *ndr*. We say the *ndr* only contains part of data classes of *dr*. calcData uses *ndr* to calculate the result.

VI. Our Scheme

A. Smart Contracts Factory

PACTA contains five smart contracts that record public policies, private policies, data requests, compliance results, available TEEs, and handle the challenge–response protocol. The smart contracts are: GC, PC, RC, compliance contract, and manager contract (MC).

GC: records the public GDPR policies in plaintext so that other DUs can make compliant requests according to the public policy. It contains four basic functions: 1) create; 2) read; 3) update; and 4) delete. Only the CSP can invoke create, update, and delete.

PC: stores DOs' private requirements in ciphertext. It contains four basic functions: 1) create; 2) read; 3) update; and 4) delete. Only the DO can invoke create, update, and delete.

RC: is used to store requests from DUs, as shown in Algorithm 1. To protect the privacy of DOs and DUs, the

Algorithm 1: RC

```
/* The $req$ is in ciphertext, $t$ refers to
   the block height. */
function request($id$, $req$, $r$, $t$)
    require($t < CurrentBlockHeight$)
    Save($id$, $req$, $r$, $t$)
function retrieve($id$)
    return ($req$, $t$)
```

Algorithm 2: Compliance Contract

```
/* The $res$ is in ciphertext, $t$ refers to
   the block height. */
function set($id$, $res$, $t$)
    require($t < CurrentBlockHeight$)
    Save($id$, $req$, $t$)
function get($id$)
    return ($res$, $t$)
```

Algorithm 3: MC

```
Data: LIST is TEE pk list. $\triangle$ is maximum length of
      legal delay.
function challenge($id$, $pk$)
    $t \leftarrow CurrentBlockHeight$
    $(req, t_1) \leftarrow RC$.retrieve($id$)
    $(res, t_2) \leftarrow CC$.get($id$)
    if $t_1 = null \mid t - t_1 \leq \triangle$ then
        Punish DU.
    else if $t_2 = null \mid t_2 - t_1 > \triangle$ then
        Punish CSP according to $pk$.
    else
        Punish DU.
    /* Deposit or reputation can be
       employed to incentive them. */
function register($att$, $pk$)
    if $verify(att)$ then
        LIST.$add(pk)$
```

TABLE I
SUMMARY OF SYMBOLS

Symbols	Meanings
pp	The public parameters.
$H(\cdot)$	The secure hash function.
(pk, sk)	The blockchain public/private key pair.
σ	The signature.
k^{dt}	The key of the DO to encrypt the data.
k^{rq}	The key of the DU to encrypt the request.
k^{rg}	The key of the regulator to decrypt the compliance result.
req	The request sent by the DU.
rep	The response of the DO.
con	The consent of the DO.
t_1	The blockchain height of request.
t_2	The blockchain height of request compliance.
t_3	The blockchain height of program compliance.

request is encrypted. It contains functions: `request` and `retrieve`. The DU invokes `request` to submit the request. The CSP invokes `retrieve` to obtain the request.

Compliance Contract (CC): records request compliance results and program compliance results, as shown in Algorithm 2. Note that we demand the CSP TEE enclave to add the DO consent (i.e., yes, no, and null) in the compliance results. The *null* means that the DO is offline when the DU requiring. A Strawman scheme stores the consent in contracts to prevent the DO from framing other entities. However, the blockchain pseudonym may leak the honest DO privacy [32]. The contract only stores compliance results, hence it contains functions: `set` and `get`.

MC: contains two functions to manage the TEE, as shown in Algorithm 3. The `register` function verifies TEE enclave attestation that proves the TEE enclave correctly runs the compliance programs and records the TEE enclave public key pk_t that proves the availability for the data sharing. The `challenge` function receives the challenge message from DUs. It checks the availability of the CSP TEE.

B. Protocol Description

In this section, we describe phases of PACTA, containing initialization, data sharing, and challenge–response. For readability, Table I lists critical parameters.

1) Initialization: The CSP registers TEE enclave in MC so that other entities can interact with the enclave. Next, regulator, DOs, and DUs generate keys with the enclave for privacy.

① *Deploying Contracts:* The regulator deploys contracts: GC, PC, RC, compliance contract, and MC. Section VI-A describes their functions.

② *Registering TEEs:* CSP enclave generates blockchain key pairs (pk_t, sk_t). The sk_t is stored in the enclave. CSP uses TEE to generate an attestation *at* which proves the enclave runs PACTA program and controls the blockchain public key and private key (pk_t, sk_t). CSP invokes the MC.`register`(pk_t, at) to register. If the $verify(at) = true$, the MC adds pk_t to the LIST. This subphase ensures that all registered enclaves run the compliance programs and that the private key sk_t remains secure and private. Therefore, the enclave does not need to repeat the attestation in the following phases.

③ *Generating Keys:* Entities generate keys with the enclave to encrypt data. They create their blockchain key pairs (pk, sk). Next, the DO creates a key $k^{dt} = \mathsf{KGen}(\mathsf{pp})$ to encrypt the data. The DU creates a key $k^{rq} = \mathsf{KGen}(\mathsf{pp})$ to encrypt the request. The regulator creates a key $k^{rg} = \mathsf{KGen}(\mathsf{pp})$ to encrypt the compliance results, where pp denotes public parameters. The enclave stores the above keys.

2) Data Sharing: The contracts in blockchain record the data-sharing process and compliance results. The DO stores data in the CSP, and the DU requests data and requires results from the CSP. The CSP provides data storage and computation service, and verifies the compliance.

① *Setting Polices:* The CSP and the DO should set the policy in the blockchain. The CSP builds the policy $\mathcal{PO} = \mathcal{S} \times \mathcal{D} \times \mathcal{O} \times \mathcal{P}$, where $\mathcal{S}$ indicates the set of DOs, $\mathcal{D}$ denotes the set of data classes, $\mathcal{O}$ implies the set of data operations, and $\mathcal{P}$ denotes the set of purposes. The CSP sends transaction $tx = (\sigma, addr_{GC}, \texttt{create}, \mathcal{PO})$ to invoke $GC.\texttt{create}$ to set GDPR polices. The DO constructs the private policy $\mathcal{PP} = \mathcal{S}' \times \mathcal{D}' \times \mathcal{O}' \times \mathcal{P}'$, sends transaction $tx = (\sigma, addr_{PC}, \texttt{create}, \mathcal{PP})$ to invoke $PC.\texttt{create}$ to set private policies.

② *Storing Data:* The DO's IoT devices generate data m, and use the key k^{dt} to encrypt m. That is, $m' = \mathsf{En}(m, k^{dt})$. The DO sends the encrypted data m' to the CSP.

③ *Requesting Data:* The DU builds the request $req = \langle s, d, o, p \rangle$, where s indicates the DO, d denotes the set of data classes, o implies the data operation, and p denotes the purpose. Then, DU encrypt the request req with key k^{rq}. That is, $req' = \mathsf{En}(q, k^{rq})$. The DU sends transaction $tx = (\sigma, addr_{RC}, \texttt{request}, id, req', r, t_1)$ to invoke $RC.\texttt{request}$ to request data from CSP, where id represents the request unique identification, r denotes a random number, t_1 indicates the time. Recall that we use the block height as the time in Section III. The RC will check the time, if the t_1 is higher than the current block height, an error is reported.

④ *Checking Requests:* The CSP continuously monitors the blockchain. When RC receives new requests from DUs, the CSP fetches the request req', the public policy $\mathcal{PO}$, the private policy $\mathcal{PP}$. Then, the CSP sends them to the TEE. The TEE performs request compliance program to check the request. First, it decrypts the request req' with k^{rq}. Second, according to the s, the TEE informs the CSP to contact with DO s. If s is online, s sends the response $rep = (\sigma, con)$ to the CSP, where con denotes the consent. Then CSP transfers rep to the TEE. Otherwise, e.g., s dose not reply within the agreed time interval, the CSP transfers $rep = null$ to the TEE. Then the TEE executes the request compliance program, as shown in Algorithm 4. If the request req is not compliant, the TEE encrypts the compliance result $res = \mathsf{En}(k^{rg}, \{res_{req}, rep\})$, where res_{req} denotes the request compliance result. Next, TEE sends the transaction $tx = (\sigma, addr_{CC}, \texttt{set}, id, res, t_2)$ to the blockchain compliance contract, where t_2 indicates the current blockchain height.

⑤ *Checking Programs:* If the request q is compliant, the CSP asks the DU to send the data analysis program. Then, the DU submits the data analysis program dap to the CSP. The CSP transfers dap to the TEE to check. Then the TEE executes the program compliance program, as shown in Algorithm 5. The TEE encrypts the compliance result $res = \mathsf{En}(k^{rg}, \{res_{req}, res_{pro}, rep\})$, where res_{pro} implies the program compliance result. Finally, the CSP sends the transaction $tx = (\sigma, addr_{CC}, \texttt{set}, id, res, t_3)$ to the blockchain compliance contract.

⑥ *Executing the Program:* If the program is compliant, the CSP will execute the program and return the results to the DU. As the actual calculation module of the program may vary depending on the specific business, we have not imposed a time limit for this particular subphase.

Algorithm 4: Request Compliance Program

```
Input: The request $req = (s, d, o, p)$, the data owner
       consent $con$ if online.
Output: The request compliance result $r$.
$d := \top$                                  ▷ The consent in default.
$r_{pg} :=$ Verify($req, PG$)
if $online$ then
    $d := con$                               ▷ Online.
else
    $r_{pr} : = d \wedge$ Verify($req, PR$)     ▷ Offline.
return $r = r_{pg} \wedge r_{pr}$
function Verify($req, R$)
    $r := \top$
    for $v$ in $m$ do
        $r \leftarrow r \wedge R.$contains$(v) ? \top : \bot$
    return $r$
```

Algorithm 5: Program Compliance Program

```
Input: The request $req = (s, d, o, p)$, the data analysis
       program $dap$.
Output: The program compliance result $r$.
$rc, er, ec, cd := dap$                      ▷ Extract modules.
$r = \bot$
$s' :=$ Extract($er$)      ▷ Extract target data owner from
    extractRow.
$d' :=$ Extract($ec$)     ▷ Extract target data classes from
    extractColumn.
if $s' = s$ and $d' \subseteq d$ then
    $r = \top$
return $r$
```

⑦ *Auditing the Results:* The regulator obtains the data from the compliance contract according to the id. Then, he/she uses the key k^{rg} to decrypt the data for auditing.

3) Challenge–Response: If the DU does not receive a timely response to its messages after a request, he/she challenges the CSP on-chain. Therefore, the DU and CSP need to monitor the blockchain for any on-chain challenges. We use the time (block height) to reflect the timeliness of responses. We set two time length endpoints t_{req} and t_{com}. Let $\triangle = t_{com} - t_{req}$ be the maximum length of legal delay. The DU sends the request not earlier than t_{req}; the CSP sends the request compliance result no later than time t_{com}. The challenge–response protocol runs in the following cases. Case 1), the CSP sends the request compliance result after time t_{com}. Case 2), the CSP sends the request compliance result time no later than t_{com}. The former represents the CSP violates the protocol, and the latter implies that the DU is malicious. Since the verification methods of both are consistent, we will take case 1) as an example.

Suppose the DU sends the request req, and the RC records the request at time t_{req}. Then the DU does not observe request compliance results at time t_{com} on compliance contract. As a result, the DU sends the transaction $tx = (\sigma, addr_{MC}, \texttt{challenge}, id, pk)$ to the MC to start the challenge–response protocol. MC obtains current block height as t. MC invokes the RC to get the request time t_1. MC invokes compliance contract to get the compliance time t_2. We give the following possible cases. Case a), if t_1 is null, we say that the DU does not issue a request. Case b), if the $t - t_1 \leq \triangle$, we say that the time limit has not been exceeded and DU's challenge is malicious. If t_1 is reasonable, MC will check t_2. Case c), if t_2 is null, we say the CSP does not respond in time. Otherwise, case d), if $t_2 - t_1 > \triangle$, we say that the CSP exceeded the time limit before giving the compliance result. Then, case e), $t_2 - t_1 < \triangle$, we say that the DU's challenge is malicious. In cases a), b), and e), the DU is malicious. In cases c) and d), the CSP is malicious.

The MC will punish the malicious entity. This can be done in a deposit-based or reputation-based manner, which is not discussed in this article.

VII. Security Analysis

A. Ensuring Immutable Logs

The proposed scheme records all operations in smart contracts on the blockchain. The RC collects all requests from DUs, while the compliance contract records all compliance results from CSP to create an immutable log of actions. The tamper-proof features of the blockchain technology guarantee that the recorded information cannot be altered, creating an auditable log of operations over time.

B. Solving Malicious CSP

In the proposed scheme, we design a challenge–response protocol, explained in Section VI-B3, to handle the malicious CSP. When the DU does not receive a timely response after a request, he/she challenges the CSP using the MC. We use the blockchain height as time constraint to restrict the response of CSP. Suppose the request is sent in time t_1, and the CSP does not give a response in time t_2. The MC obtains the two time points from the RC and the compliance contract, respectively. If $t_2 - t_1 > \triangle$, or the $t_2 = null$, the CSP should be punished. Our scheme can employ deposit-based or reputation-based strategies to solve it.

C. Enforcing Regulation While Protecting Privacy

We use a combination of symmetric encryption algorithm and TEE to protect the privacy of sensitive data involved in the regulatory process. In the proposed scheme, we encrypt the shared data, requests, and compliance results with key k^{dt}, k^{rq}, and k^{rg}, respectively. The TEE is used to execute the request compliance program that contains sensitive information, the adversary cannot obtain the TEE internal data under the assumption in Section III. Furthermore, Pacta uses TEE to run the program compliance program to avoid potential privacy leaks, e.g., the outputs contains the name of the DO. These measures protect entity privacy by ensuring that only authorized regulator can access the compliance results during the auditing process.

D. Ensuring DO's Consent

The DO's consent is crucial in the data analysis process for DO data, and our scheme prioritizes efficient processing and respect for the DO's rights accordingly. The CSP checks the DO's online status and responds accordingly. In the case where the DO is offline, the TEE only verifies the request by checking it against the DO's preset private policies stored in the PC. When the DO is online, the CSP forwards the request and compliance results to the DO for consent. With the DO's consent, the TEE then proceeds to the next operation stage.

VIII. Evaluation

In this section, we evaluate the feasibility of our scheme. We first code smart contracts using Ethereum Solidity, and assess the gas costs. Then, we implement the compliance verification program in Intel SGX to evaluate the time overhead.

A. Implementation and Data Sets

We use Ganache[7] to simulate a local Ethereum blockchain. We use Solidity to write the five smart contracts for each function execution. It is the most used programming languages of Ethereum, and can write contracts with self-executing business logic and embedded in smart contracts. These contracts were tested using Remix, a Web-based open-source Solidity development environment that provides basic compilation, deployment to the local or test network, and execution of contracts. Based on Remix, we can create transactions to interact with these contracts, and obtain the gas costs.

The request compliance program and program compliance program are running in TEE enclaves. Codes are written using the C++ language. We need to emphasize that our scheme is TEE-agnostic, and all commercial TEE that can achieve general-purpose computation and satisfy the memory requirements of the program can perform these programs at the CSP. We choose Intel SGX for our implementation.

We use the public data set, namely, *Deep Healthcare Analysis using BigQuery*[8] The data set contains: 1) public medical data, created by the Centers for Medicare and Medicaid Services and 2) several public data analysis programs, written in Python, for above medical data. These data involve procedures, services, and prescription drugs by inpatient hospitals, outpatient hospitals, physicians, and other providers.

B. Setup

We deploy a test setup with our implementation for performance measurements. The test setup runs in one laptop, which runs Windows11 on an Intel Core i7-9750H CPU @ 2.60 GHz and 8-GB RAM. This CPU supports Intel SGX, since Intel 11th Generation Core Rocket Lake and 12th Generation Core Alder Lake do not support SGX.[9] We use the Intel SGX SDK for Windows.[10] The enclave thread stack

[7] Ganache - Truffle Suite.

[8] Deep Healthcare Analysis using BigQuery.

[9] Where Is a List of Processors that Support Intel Software Guard Extensions (Intel SGX)?

[10] Intel Software Guard Extensions SDK for Windows.

TABLE II
GAS COSTS OF EXECUTING PACTA CONTRACTS

Contracts	Functions	Gas	USD [1]
GDPR Contract	`create`	403,523	18.11
	`update`	83,040	3.73
	`delete`	188,461	8.46
Private Contract	`create`	106,511	4.78
	`update`	37,937	1.70
	`delete`	62,000	2.78
Request Contract	`request`	102,091	4.58
Compliance Contract	`set`	102,091	4.58
Manager Contract	`register`	2,445,021	109.73
	`challenge`	75,614	3.39

[1] The USD costs were estimated based on the prices on Jun. 27, 2023 [33], [34].

size is 256 kB, and the heap size is 1 MB. We use ganache-cli to emulate a local Ethereum blockchain, which runs the contracts deployed by the Remix.

C. Gas Costs

Since we implement smart contracts in Ethereum, interactions with them incur some gas costs, which introduces a certain economic overhead. The gas costs of contracts functions are listed in Table II. Specifically speaking, the regulator deploys the five contracts as stated in Section VI-A. The CSP registers the TEE with function `register` of MC. The CSP stores the public policy with function `create` of GC. The DO sets the private policy with function `create` of PC. The DUs send their requests with the function `request` of RC. The TEE stores the compliance results with function `set` of compliance contract. The DO invokes the function `challenge` of MC if the CSP cannot response in time. Since the `read`, `retrieve`, and `get` are read operations and do not consume gas, we do not list them. The gas overhead is proportional to the data size.[11] The GC policy uses 102 bytes. Each policy occurs 192 bytes + (data classes size × 64 bytes), In this experiment, we set the GC policy as 102 bytes, and the PC policy as 44 bytes. The request and compliance both take 96 bytes. Although the `create` functions of GC and PC are expensive, they will only be called once. Then these policies will be updated later with the cheap `update`. Currently, the Solidity smart contract cannot implement Intel SGX remote attestation, thus we adopt the elliptic curve digital signature algorithm (ECDSA) algorithm to simulate the attestation. The `register` takes 140 bytes, where the public key occupies 64 bytes and the attestation takes 64 bytes. The `challenge` uses 32 bytes.

D. Time Costs

We test the two programs and the attestation generation program. We repeat the experiments 100 times and take the averages. For `register`, the signature costs 1578 microseconds (μs). The request compliance program spends 21 μs for check one request. The program compliance program uses 309 μs for the python codes in data set 2.1 section, since it contains several columns as the data classes in our scheme. The attestation generation program costs 1549 μs, it only needs to be called once during registration. Besides, there are two necessary functions:

1) `sgx_create_enclave` creates an enclave for running above programs.
2) `sgx_destroy_enclave` destroys the enclave when it is not expecting further execution.

The former spends 18 285 μs and the latter uses 873 μs.

IX. DISCUSSION

A. Research Scope

PACTA aims to present a complete compliance scheme for IoT data regulation that ensures regulatory functionality while concurrently addressing privacy concerns.

1) Completeness: Compliance completeness refers to ensuring compliance throughout the entire data life-cycle, from data request to data utilization. To formalize the DO's consent, we establish a comprehensive approach for obtaining consent, encompassing both online and offline scenario. In the data utilization phase, apart from analyzing compliance with data request requirements, it is imperative to conduct compliance analysis on the data analysis program as well. Therefore, the compliance of the data utilization process is thoroughly safeguarded.

2) Regulation: The purpose of regulation is to establish and enforce rules, standards, and guidelines that govern various aspects of society. PACTA leverages blockchain as the trusted platform to record all interactive data, such as *private policies*, *requests*, and *compliance results*, facilitating regulator audits.

3) Privacy: PACTA adopts cryptography techniques and TEE techniques to ensure privacy. First, it employs blockchain to store interactive data in ciphertext, thereby preventing unauthorized entities from accessing the actual data. Additionally, PACTA utilizes TEE to securely execute the request compliance program and program compliance program, safeguarding the confidentiality of *private policies*, *requests*, and *compliance results* from the CSP.

B. Limitations and Mitigation

Our research concentrates on ensuring the integrity of compliance, relying on the security assumptions associated with blockchains and TEEs. However, we acknowledge that matters concerning availability and economic costs in production environments extend beyond the scope of this study. Consequently, the following limitations arise.

1) Right to Erasure: The right to erasure (forgotten) (Art.17) in GDPR regulations requires DC to erase personal data. The immutability of blockchain technology conflicts with the right to erasure. However, there are some works have proposed to rewrite the history of blockchain. Ateniese et al. [35] employed the chameleon hash functions to modify the Bitcoin history. Deuber et al. [36] designed a consensus-based voting to avoid heavy cryptographic tools or trust assumptions. KERB also uses chameleon hashes with monetary penalty to control rewriting privileges and penalize

[11] Gas and fees.

malicious behaviors [37]. Some researchers explore redactable blockchains in decentralized environments [38], [39] and improve security [40]. Our work can employ above blockchain systems to inherit the right to erasure.

2) Security Assumption: Our scheme hinges upon the security and reliability of both the blockchain and TEE. Should these components be compromised, the confidentiality of requests and private requirements, as well as the correctness of compliance judgments, cannot be ensured.

For data from the blockchain. Request compliance verification and program compliance verification run by TEE depend on blockchain data, such as policies and requests. As a result, it is essential to ensure that the blockchain data provided to TEE is consistent with the main chain. However, the TEE cannot access to blockchain data and must rely on data provided by the CSP. This could lead to the generation of fake compliance results, abducted by the CSP. Several methods have been proposed to ensure the correctness and timeliness of blockchain data obtained by TEE [19], [41]. These methods rely on the assumption that honest blockchain nodes have a computing power advantage, preventing malicious blockchain nodes from generating enough blocks to falsify data. Our scheme is also based on this assumption and can be applied to these methods to ensure the correctness of data acquired by TEE.

For the assumption of TEE. Our system can support multiple TEEs in CSP to avoid the single point of failure to improve system availability. Multiple TEEs work together, enabling confidential communication, information coordination, key management, and the execution of request and program compliance verification. Various industrial implementations currently exist, such as Hyperledger Sawtooth [42], Microsoft CCF [43], and Visa's LucidiTEE [44], which leverage TEE technology to build blockchains. These implementations also use Raft or BFT consensus algorithms to maintain a replicated copy of the data among multiple nodes, avoiding a single point of failure. By implementing CSP on these systems, the system availability can be improved.

3) Economic Costs: Despite the convenience offered by Ethereum in facilitating economic payments among participants, the volatile and comparatively expensive nature of ETH poses potential economic disputes. To mitigate these challenges, one viable alternative is the adoption of stablecoins. Take the USTD, for example, which maintains a stable one-to-one exchange rate with the US dollar, thereby serving as a reliable medium of exchange. Additionally, we propose leveraging blockchain layer2 technologies and projects to minimize overhead costs and enhance operational efficiency [45], [46].

C. Future Work

Moving forward, our future research will focus on the following directions.

1) Strengthening the Security Assumptions of TEE: The effectiveness and security of our proposed scheme heavily rely on the confidentiality, integrity, and remote authentication provided by TEE. Addressing the challenges that arise when one or more of these TEE features are compromised is of utmost importance.

2) Cross-Organizational Compliance in a Multinational Context: Given the variations in data privacy laws and regulations across different countries and organizations, ensuring compliance in multinational companies and enterprises involved in cross-border data issues poses a significant challenge. Finding effective strategies for handling cross-organizational data compliance is imperative.

3) Best Practice: Additionally, we are currently in the process of developing a robust privacy regulation compliance framework specifically tailored for Medical IoT data sharing. Our aim is for this framework to serve as an industry best practice.

X. Conclusion

This article introduces PACTA, a compliance scheme for IoT data privacy regulations that leverages TEE and blockchain technology. PACTA is capable of managing both dynamic and static consent, ensuring compliance completeness. By utilizing TEE, PACTA performs compliance checks for both requests and procedures, addressing regulation faultiness. Crucial data is stored in ciphertext on the blockchain to facilitate audits by regulators. The evaluation results demonstrate the efficiency of TEE in verifying compliance, while highlighting the inappropriate cost-effectiveness of using Ethereum for data storage.

Moreover, this framework has the potential to extend to other cloud application domains where compliance is imperative. Nonetheless, several open problems of interest remain. In terms of compliance, the use of a redactable blockchain can better support the right to erasure. Additionally, finding effective strategies for managing cross-organizational data compliance is crucial. Regarding system design, an important challenge lies in reducing the security dependency on TEE, not only for this work but also for any TEE-based system.

References

[1] Y. Zhang, K. Gai, J. Xiao, L. Zhu, and K. R. Choo, ‘‘Blockchain-empowered efficient data sharing in Internet of Things settings,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3422–3436, Dec. 2022.

[2] K. Figueredo, D. Seed, and C. Wang, “A scalable, standards-based approach for IoT data sharing and ecosystem monetization,” *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5645–5652, Apr. 2022.

[3] J. Isaak and M. J. Hanna, “User data privacy: Facebook, Cambridge analytica, and privacy protection,” *Computer*, vol. 51, no. 8, pp. 56–59, Aug. 2018.

[4] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, “Privacy preservation in federated learning: An insightful survey from the GDPR perspective,” *Comput. Security*, vol. 110, Nov. 2021, Art. no. 102402.

[5] Q. Miao, H. Lin, J. Hu, and X. Wang, “An intelligent and privacy-enhanced data sharing strategy for blockchain-empowered Internet of Things,” *Digit. Commun. Netw.*, vol. 8, no. 5, pp. 636–643, Oct. 2022.

[6] Y. Wang, A. Zhang, P. Zhang, Y. Qu, and S. Yu, “Security-aware and privacy-preserving personal health record sharing using consortium blockchain,” *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12014–12028, Jul. 2022.

[7] S. D. Okegbile, J. Cai, and A. S. Alfa, “Performance analysis of blockchain-enabled data-sharing scheme in cloud-edge computing-based IoT networks,” *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21520–21536, Nov. 2022.

[8] J. Ahmad, F. Li, and B. Luo, “IoTPrivComp: A measurement study of privacy compliance in IoT apps,” in *Proc. Euro. Symp. Res. Comput. Security*, 2022, pp. 589–609.

[9] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, “GDPR-compliant personal data management: A blockchain-based solution,” *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1746–1761, 2020.

[10] M. Barati et al., “Privacy-aware cloud auditing for GDPR compliance verification in online healthcare,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 7, pp. 4808–4819, Jul. 2022.

[11] M. Barati and O. F. Rana, “Tracking GDPR compliance in cloud-based service delivery,” *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1498–1511, May/Jun. 2020.

[12] L. Yeh, C. Shen, W. Huang, W. Hsu, and H. Wu, “GDPR-aware revocable P2P file-sharing system over consortium blockchain,” *IEEE Syst. J.*, vol. 16, no. 4, pp. 5234–5245, Dec. 2022.

[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: A survey,” *Int. J. Web Grid Services*, vol. 14, no. 4, pp. 352–375, Oct. 2018.

[14] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, “A survey on privacy protection in blockchain system,” *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Jan. 2019.

[15] L. Wang et al., “PrivGuard: Privacy regulation compliance made easier,” in *Proc. USENIX Security*, Boston, MA, USA, 2022, pp. 3753–3770.

[16] M. Rhahla, S. Allegue, and T. Abdellatif, “Guidelines for GDPR compliance in big data systems,” *J. Inf. Security Appl.*, vol. 61, Sep. 2021, Art. no. 102896.

[17] G. Lax and A. Russo, “A lightweight scheme exploiting social networks for data minimization according to the GDPR,” *IEEE Trans. Comput. Social Syst.*, vol. 8, no. 2, pp. 388–397, Apr. 2021.

[18] J. A. Garay, A. Kiayias, and N. Leonardos, “The bitcoin backbone protocol: Analysis and applications,” in *Proc. Adv. Crypt. EUROCRYPT*, 2015, pp. 281–310.

[19] T. Frassetto et al., “POSE: Practical off-chain smart contract execution,” in *Proc. Ann. Netw. Distri. Syst. Secur. Symp.*, San Diego, CA, USA, 2023, pp. 1–18.

[20] Q. Ren et al., “Cloak: Transitioning states on legacy blockchains using secure and publicly verifiable off-chain multi-party computation,” in *Proc. Ann. Comput. Security Appl. Conf.*, New York, NY, USA, 2022, pp. 117–131.

[21] M. H. Miraz and M. Ali, “Applications of blockchain technology beyond cryptocurrency,” *Ann. Emerg. Tech. Comput.*, vol. 2, no. 1, pp. 1–6, Jan. 2018.

[22] K. Choi, A. Manoj, and J. Bonneau, “SoK: Distributed randomness beacons,” in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, 2023, pp. 75–92.

[23] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, “Town crier: An authenticated data feed for smart contracts,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Vienna, Austria, 2016, pp. 270–282.

[24] I. Bentov, Y. Ji, F. Zhang, L. Breidenbach, P. Daian, and A. Juels, “Tesseract: Real-time cryptocurrency exchange using trusted hardware,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, London, U.K., 2019, pp. 1521–1538.

[25] R. Cheng et al., “Ekiden: A platform for confidentiality-preserving, trustworthy, and Performant smart contracts,” in *Proc. IEEE Euro. Symp. Security Privacy*, Stockholm, Sweden, 2019, pp. 185–200.

[26] P. Das et al., “FastKitten: Practical smart contracts on bitcoin,” in *Proc. USENIX Security*, Santa Clara, CA, USA, 2019, pp. 801–818.

[27] Z. Chen, G. Vasilakis, K. Murdock, E. Dean, D. F. Oswald, and F. D. Garcia, “VoltPillager: Hardware-based fault injection attacks against Intel SGX enclaves using the SVID voltage scaling interface,” in *Proc. USENIX Security*, 2021, pp. 699–716.

[28] W. Zhao, K. Lu, Y. Qi, and S. Qi, “MPTEE: Bringing flexible and efficient memory protection to Intel SGX,” in *Proc. Euro Conf. Comput. Syst.*, Heraklion, Greece, 2020, pp. 1–15.

[29] A. Ahmad, B. Joe, Y. Xiao, Y. Zhang, I. Shin, and B. Lee, “OBFUSCURO: A commodity obfuscation engine on Intel SGX,” in *Proc. Ann. Netw. Distribut. Syst. Security Symp.*, San Diego, CA, USA, 2019, pp. 1–15.

[30] S. Johnson, V. Scarlata, C. Rozas, E. Brickell, and F. Mckeen, “Intel software guard extensions: EPID provisioning attestation services,” Intel, Santa Clara, CA, USA, White Paper, 2016. Accessed: May, 2023. [Online]. Available: https://www.intel.com/content/www/us/en/content-details/671370/intel-sgx-intel-epid-provisioning-and-attestation-services.html

[31] A. M. Devices, “AMD SEV-SNP: Strengthening VM isolation with integrity protection and more,” Intel, Santa Clara, CA, USA, White Paper, 2020. Accessed: May 2023. [Online]. Available: https://www.amd.com/system/files/TechDocs/SEV-SNP-strengthening-vm-isolation-with-integrity-protection-and-more.pdf

[32] P. Chatzigiannis, F. Baldimtsi, and K. Chalkias, “SoK: Auditability and accountability in distributed payment systems,” in *Proc. Int. Conf. Appl. Crypto. Netw. Security*, Kamakura, Japan, 2021, pp. 311–337.

[33] Etherscan. “Ethereum average gas price chart.” Accessed: Jun. 2023. [Online]. Available: https://etherscan.io/chart/gasprice

[34] CoinMarketCap. “Ethereum (ETH) price.” Accessed: Jun. 2023. [Online]. Available: https://coinmarketcap.com/currencies/ethereum/

[35] G. Ateniese, B. Magri, D. Venturi, and E. R. Andrade, “Redactable blockchain–or–rewriting history in bitcoin and friends,” in *Proc. IEEE Euro. Symp. Security Privacy*, Paris, France, 2017, pp. 111–126.

[36] D. Deuber, B. Magri, and S. A. K. Thyagarajan, “Redactable blockchain in the permissionless setting,” in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, 2019, pp. 124–138.

[37] S. Xu, J. Ning, J. Ma, X. Huang, and R. H. Deng, “K-time modifiable and epoch-based redactable blockchain,” *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4507–4520, 2021.

[38] M. Jia et al., “Redactable blockchain from decentralized chameleon hash functions,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2771–2783, 2022.

[39] J. Ma, S. Xu, J. Ning, X. Huang, and R. H. Deng, “Redactable blockchain in decentralized setting,” *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1227–1242, 2022.

[40] J. Li, H. Ma, J. Wang, Z. Song, W. Xu, and R. Zhang, “Wolverine: A scalable and transaction-consistent redactable permissionless blockchain,” *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 1653–1666, 2023.

[41] A. R. Choudhuri, M. Green, A. Jain, G. Kaptchuk, and I. Miers, “Fairness in an unfair world: Fair multiparty computation from public bulletin boards,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Dallas, TX, USA, 2017, pp. 719–728.

[42] “Hyperledger sawtooth.’ Accessed: May 2023. [Online]. Available: https://sawtooth.hyperledger.org/

[43] A. Shamis et al., “IA-CCF: Individual accountability for permissioned ledgers,” in *Proc. USENIX Symp. Netw. Syst. Des. Implement.*, Renton, WA, USA, 2022, pp. 467–491.

[44] R. Sinha, S. Gaddam, and R. Kumaresan, “LucidiTEE: A TEE-blockchain system for policy-compliant multiparty computation with fairness,” IACR, Bellevue, WA, USA, Rep. 2019/178, 2019. [Online]. Available: https://eprint.iacr.org/2019/178

[45] L. Gudgeon, P. Moreno-Sanchez, S. Roos, P. McCorry, and A. Gervais, “SoK: Layer-two blockchain protocols,” in *Proc. Finan. Crypt. Data Security Int. Conf.*, 2020, pp. 201–226.

[46] L. Aumayr, S. A. K. Thyagarajan, G. Malavolta, P. Moreno-Sanchez, and M. Maffei, “Sleepy channels: Bi-directional payment channels without watchtowers,” in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, Los Angeles, CA, USA, 2022, pp. 179–192.

Yongxin Zhang received the bachelor’s degree from Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2018, and the master’s degree from Northeastern University, Shenyang, China, in 2021.

He is currently a Researcher with the Blockchain Research Group, Oxford-Hainan Blockchain Research Institute, Chengmai, China, and also a Researcher with the R&D Department, SSC Holding Company Ltd., Chengmai. His research interests include blockchain technology.

Jiacheng Yang received the bachelor’s degree from Hainan University, Haikou, China, in 2022.

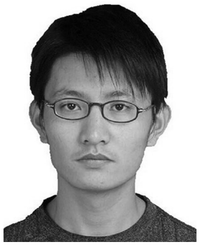
He is currently a Research Assistant with the Blockchain Research Group, Oxford-Hainan Blockchain Research Institute, Chengmai, China, and also a Researcher with the R&D Department, SSC Holding Company Ltd., Chengmai. His research interests include blockchain technology and trusted hardware technology.

Hong Lei received the bachelor's and master's degrees from Beihang University, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree from Michigan State University (MSU), East Lansing, MI, USA, in May 2015.

He continued as a Postdoctoral Fellow with Smart Microsystem Laboratory, MSU. He joined Schweitzer Engineering Laboratory, Pullman, WA, USA, in 2016, and then joined the Department of Electrical and Computer Engineering as a Tenure-Track Assistant Professor with Portland State University, Portland, OR, USA, in July 2018. He was appointed as the Associate Dean of Oxford-Hainan Blockchain Research Institute, Chengmai, China, in June 2019. He is currently a Professor with Hainan University, Haikou, China, and doing researches on TEE and Blockchain.

Zijian Bao received the M.S. degree in computer application technology from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree with the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

His research interests include cryptographic protocols.



Ning Lu received the M.S. degree from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2009, and the Ph.D. from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, in 2013.

He is currently an Associate Professor with Northeastern University, Shenyang, China. His current research interests include artificial intelligence security, data security and privacy protection, and denial-of-service attack defense.

Wenbo Shi received the M.S. and Ph.D. degrees from Inha University, Incheon, South Korea, in 2007 and 2010, respectively.

He is currently a Professor with Northeastern University at Qinhuangdao, Qinhuangdao, China. His research interests include the cryptographic protocol, cloud computing security, artificial intelligence security, data security, privacy protection, and denial-of-service attack defense.

Bangdao Chen received the bachelor's degree from the Department of Computer Science, Shanghai Jiao Tong University, Shanghai, China, in 2006, and the M.Sc. and D.Phil. degrees in computer science from the University of Oxford, Oxford, U.K, in 2007 and 2013, respectively.

His main research directions were decentralized identification and authentication, payment security, and communication security. He is currently mainly engaged in cyber security, and blockchain related technology research and product development.