DUCE: Distributed Usage Control Enforcement for Private Data Sharing in Internet of Things

Na Shi[1], Bo Tang[1], Ravi Sandhu[2], and Qi Li[3]

[1] Sichuan Changhong Electric Co., Ltd.
[2] Institute for Cyber Security, NSF Center for Security and Privacy Enhanced Cloud Computing, and Dept. of Computer Science, University of Texas at San Antonio
[3] Tsinghua University
{na.shi,bo.tang}@changhong.com; ravi.sandhu@utsa.edu;
qli01@tsinghua.edu.cn;

Abstract. The emerging Cloud-Enabled Internet of Things (CEIoT) is becoming increasingly popular since it enables end users to remotely interact with the connected devices, which collect real-world data and share with diverse cloud services. The shared data will often be sensitive as well as private. According to the General Data Protection Regulation (GDPR), the privacy issue should be addressed by the cloud services and subsequent data custodians. In this paper, we propose DUCE, an enforcement model for distributed usage control for data sharing in CEIoT. DUCE leverages both blockchain and Trusted Execution Environment (TEE) technologies to achieve reliable and continuous life-cycle enforcement for cross-domain data sharing scenarios. The core components of DUCE are distributed Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) to enable reliable execution of usage control policies without a centralized trusted authority. Policy administration is also distributed and controlled by the data owner, who can modify the rules anywhere anytime. The policy rules expressed in eXtensible Access Control Markup Language (XACML) are parsed into smart contracts to be executed on the blockchain service. A detailed explanation of the enforcement process is given for an example "delete-after-use" rule. A prototype system is implemented with an open-source permissioned blockchain system and evaluated on an experimental deployment. The results show reasonable performance and scalability overhead in comparison to OAuth 2.0. We believe additional cross-domain data usage control issues can also be addressed by DUCE.

Keywords: Cloud-Enabled Internet of Things · Privacy · Usage Control · Blockchain · Trusted Execution Environment.

1 Introduction

The Internet of Things (IoT) extends the boundary of the familiar Internet by incorporating smart physical objects (things) embedded with sensors, actuators, software and communications hardware, for the purpose of connecting and exchanging data with other devices and systems. The continuing convergence of cloud computing and IoT has brought about the concept of Cloud-Enabled Internet of Things (CEIoT) [4–6, 8] which is a new computing paradigm bringing together the complementary advantages of cloud computing and IoT. In this paradigm, a cloud computing service is used to provide convenient access to online applications and services, and a IoT service is used to enable sensing and control of the physical world, through which increasingly comprehensive data interactions facilitate "smarter" applications for end users.

Applications of CEIoT span a diverse set of consumer, industrial and professional scenarios. Data about healthcare collected and stored in a secure manner can be shared, to provide remote sensing detection services for elderly healthcare [22], to provide patients some facilities through telehealth [2], to promote medical research [9], etc. Contaminated water detected can be shared to prevent users and crops from outbreak of diseases [10, 21]. As for transportation, smart parking service data can be shared and driving habits monitored to provide vehicle owners services such as warranty and insurance discounts for safe driving [12, 13]. The proliferation of data sharing in CEIoT is an emerging trend with great potential and may even become the future of the Internet [11, 28].

CEIoT presents significant privacy concerns especially in consumer-oriented applications [1, 16, 26]. Most shared data collected by user devices is sensitive and private. Despite the fact that access control mechanisms can be used to prevent the data from leakage during an access, the data shared to an external application is not subject to this restriction. Furthermore, the behaviors of the external application cannot be monitored or controlled once the data is shared, whereby the usage of the data may violate articles such as "*Rights to erasure*" in General Data Protection Regulation (GDPR). Thus appropriate privacy preserving mechanisms need to be developed to mitigate this risk and realize the true potential of such data sharing. In this paper, we propose a distributed usage control enforcement model, namely DUCE, to address the aforementioned privacy concerns in CEIoT.

The key contributions of this work are as follows. (i) A DUCE design overview is given with the system components including the distributed PDPs and PEPs. DUCE leverages permissioned blockchain technology to build a trusted relationship between data-sharing parties, whereby the rules and enforcement records are tamper-proof and visible to users. A Trusted Execution Environment (TEE) is used to ensure that the enforcement process of the rules and the usage of user data are trustworthy and controllable by users. (ii) The policy administration model of DUCE is also provided with a policy example of "delete-after-use" in XACML and the policy translation algorithm into Solidity language for smart contracts. (iii) A prototype system is implemented and deployed along with an OAuth 2.0 benchmark system. The end-to-end delay and throughput are evaluated and analyzed to demonstrate the viability of DUCE.

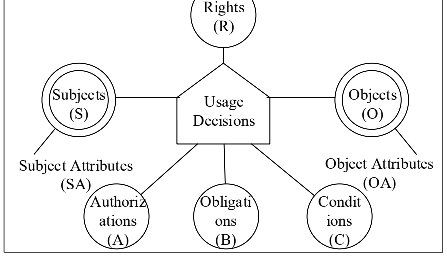
Organization. Section 2 reviews essential technical concepts. Section 3 provides a typical user scenario, the problem statement and the design goals. DUCE is developed in Section 4. Section 5 discusses the experiment results. Related work is summarized in Section 6. Section 7 concludes the paper.

2 Background

Usage Control (UCON). Traditional access control models [27] deal with authorization as sole basis for access decisions and typically focus only on *server-side* controls. The UCON model, namely UCON_{ABC} [24], enables mutability of subject and object attributes, as well as continuity of control on usage of digital resource, and focuses on both *server-side* and *client-side* controls. The basic access control decision in any access control model can be represented as a triple (s, o, r, c), in which s denotes the subject S exercising a right r for object O under conditions c. UCON comprises eight core components, as shown in Fig. 1 to resolve this question. There are three functional predicates that have to be evaluated for usage decisions. The authorizations denote specific rights that a subject may exercise, the obligations denote actions the subject must perform and the conditions denote criteria influenced only by system-wide conditions.
Cloud-Enabled Internet of Things (CEIoT) is a basic IoT three layers architecture [5, 6]. Perception layer includes devices that can perceive and collect data. Middle layer, in which components have functions to transfer data, communication and provide data services. Application layer provides diverse applications to meet the needs of the society and users. IoT devices are typically resource-constrained while close to real data, while cloud computing can provide elastic scalable storage, computing, and analysis. Therefore, the current emerging and widely used architecture called CEIoT integrates the IoT and the cloud, wherein cloud service providers (CSPs) expand services and applications via Internet on the existing foundation based on the above basic IoT.
Distributed Ledger Technology (Blockchain) is a technology [18, 19] linking records expressed as blocks on a chain through cryptography, initially deployed to address the double-spending and currency generation problems of the Bitcoin cryptocurrency [20]. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data usually expressed as a Merkel Tree. For use as a distributed ledger, nodes in a blockchain system are usually managed by a peer-to-peer network, and encouraged to follow protocols for communicating and validating new blocks by incentives. As a decentralized infrastructure and distributed computing paradigm with the characteristics of tamperproof, traceability, and joint maintenance by multiple parties, blockchain has considerable promise for the construction of future IoT systems. As an autonomous application program running in the isolated virtual machine on a blockchain system, the smart contract provides a novel mechanism that can autonomously manage and implement interaction-rules between related parties.

3 Problem Statement and Design Goals

Problem Statement. In this paper, we address data sharing in CEIoT [23]. We assume a typical data usage scenario, as shown in Fig. 2. Suppose Alice acquires a wristband to collect the family's health data, such as heartbeat, exercise and sleep. Alice desires that the device platform of this wristband, can share her data



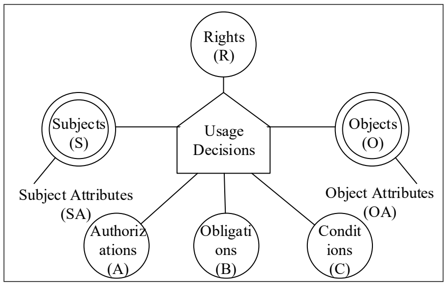


Fig. 1. UCON policy model [24]

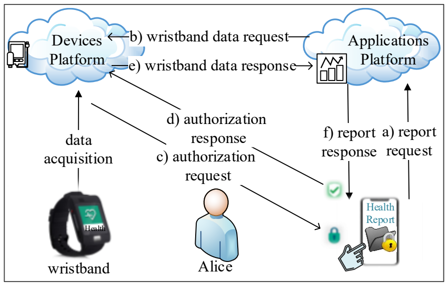
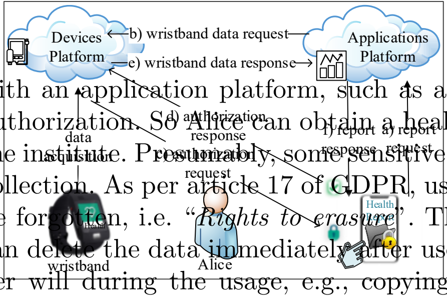


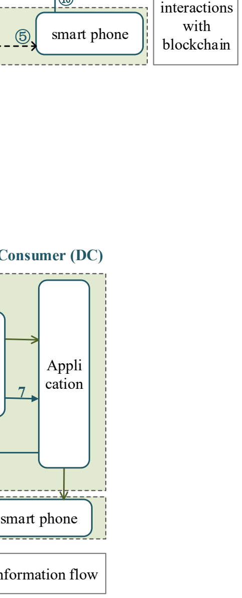
Fig. 2. A private data-sharing scenario



with an application platform, such as a professional health institute, with her authorization. So Alice can obtain a health report after the data is processed by the institute. Presumably, some sensitive and private data is included in her data collection. As per article 17 of GDPR, users' have the right to have their data to be forgotten, i.e. "*Rights to erasure*". Thereby, Alice expects that the institute can delete the data immediately after use, as well as not doing anything against her will during the usage, e.g., copying without her authorization or sharing to others directly in the current system. We note that any solution to such requirements must impose some perimeter restriction whereby all processing of the data takes place within this perimeter. In particular, so-called analog hole operations such as taking photos of display screens or manually copying data are beyond the scope of purely technical solutions.

To satisfy Alice's expectations, the wristband uploads data sporadically to its platform, viz. Devices Platform. Subsequently, a health report request is initiated by Alice via the application button on her smart phone, and the action triggers a professional institute, viz. Applications Platform to send a data request to the devices platform, in steps a) and b). After receiving a data request, the devices platform indicates to Alice that authorization is required through a visualized and unambiguous view in step c). Then, the interface is redirected back to the application with permissions, as indicated in step d). Successful authorization by Alice allows the data to be communicated to applications platform by devices platform in step e). Finally, the data is used to compute a health report which is delivered to Alice by the applications platform in step f).

Design Goals. In the above scenario, Alice wants a professional analysis report, which is a task that a devices platform or a general data storage party cannot fulfill. Thus the data needs to be shared with a third party such as a health institution. Moreover, Alice wants continued control of data usage wherein a "delete-after-use" rule is defined. However, in a distributed architecture, once data is shared, users lose control of the data. Due to mutual untrustworthy relationship between participants, users cannot be sure whether the applications platform follows the rules, and the applications platform cannot prove to users that they did not break the rules and breach privacy.



Event flow
Message flow ()
Data flow
blockchain operation

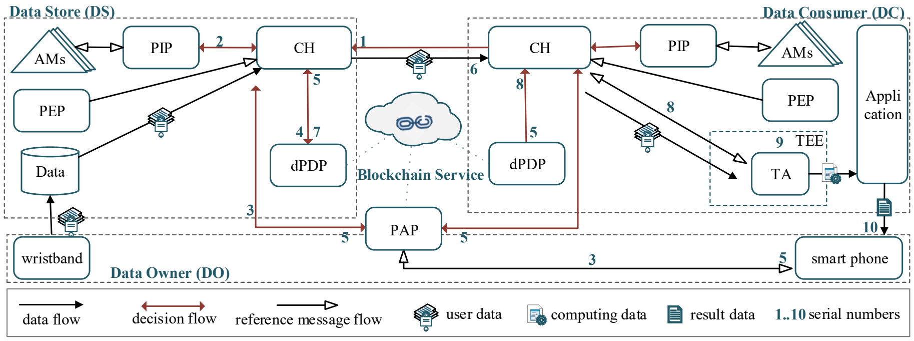


Fig. 3. System overview of DUCE, a trusted and distributed enforcement model.

Therefore, to prevent privacy compromises, a trusted relationship should be established between mutually untrustworthy participants to keep data usage completely visible and in absolute control of users. This motivation drives our goals as follows, and inspires us to design a privacy-preserving distributed usage control enforcement model for data sharing in CEIoT. We recognize the following derivative goals. ***Privacy Preserving*** requires that the shared user data and the keys in authorization used to decrypt this data should be protected. ***Integrity Protection*** requires that the policy defined by users and enforcement records should not be tampered with. ***Traceability*** requires that violations must be able to be traced through enforcement records, and are visible to users.

QA:
1. CH & PEP;
2. TEE & Application;
3. Blockchain Service & dPDPs, PAP;

4 The DUCE Model for Cloud-Enabled IoT

In this section, we present an overview of DUCE and its various components, and develop its enforcement process and an administration model. A system overview of DUCE is given in Fig. 3. A blockchain service is leveraged to construct a trusted and distributed architecture, in which data-sharing participants including data stores, data consumers and data owners are orchestrated and the data usage control rules can be enforced with administration and visibility by data owners via distributed PDPs and PEPs. Moreover, the policy defined by data owners is not only an authorized foundation to share data for a data store, but also a rule constraint for use of data by a data consumer. DUCE connects Data Stores (DSs), Data Consumers (DCs) and Data Owners (DOs) via a Blockchain Service. The system components of DUCE are discussed below.

Policy Enforcement Point (PEP) is a distributed component coupled with protected resources (i.e. the user data stored in a data store), which can intercept usage requests initiated by accessing subjects to trigger a decision through assessing the access request via available attributes, and finally enforce the result returned by an allow or deny decision. In DUCE, the PEP is a distributed engine used to enforce usage requests and perform specific decisions for user data. Additionally, the PEPs incorporate a Context Handler (CH) which plays the role of coordinator in an entire usage policy decision-making process and manages workflow by interacting with all other components.

distributed Policy Decision Point (dPDP) is an adjudicator who makes decisions including allow or deny. It returns the decision to PEPs after the parameters including policy, usage requests, and current available attributes are evaluated. It is an essential aspect of DUCE.

Policy Administration Point (PAP) is a component responsible for management, storage and retrieval service during the evaluation process of usage requests. Meanwhile, PAP can also help decision-makers to define and modify policy, or perform other more complex and related policy management actions.

Attribute Manager (AM) is a component in charge of managing usage, retrieval and update of subject, object and environmental attributes. In DUCE, subject attributes mainly refer to general, authorization, obligation and condition attributes. The object attributes mainly refer to attributes, such as times to use and unique identifiers. The environmental attributes mainly refer to attribute values that are only effected by administrative operations in systems. AMs are not confined to an authorization service, but can be extended to local services, cloud services, or other services in different management domains.

Policy Information Point (PIP) is an interactive interface between diversified AMs, which provides attribute retrieval and update services, whereas attribute sets required for evaluation are collected by different AMs and protocols.

Data Owner (DO) provides PAP and ownership service to IoT devices and user data. DO is responsible for administrating usage policy and can control the entire enforcement by interacting with context handlers in PEPs.

Data Store (DS) provides a hosting service for user data. As a PEP, the data store translates the policy defined by a data owner into a form that can be evaluated by dPDPs, and then determines whether to call the PIP to perform data-sharing based on the distributed evaluation results. Whether or not the data is shared, the data store needs to send a notification to the user through CH. Formally, DS=<CH,PEP,dPDP,PIP,AMs,Data>, where the Data denotes an object. More precisely, we denote CH in DS as $\mathrm{CH}_{\mathrm{DS}}$, thus $\mathrm{CH}_{\mathrm{DS}}$=<$H_{pol}$,$H_{oat}$,$H_{obj}$, H_{not}>, where H_{pol} receives policy defined in XACML from DO, and translates policy into a form that can be evaluated in a dPDP. H_{oat} updates object attributes. H_{obj} handles user data, and H_{not} sends notifications to data owners.

Data Consumer (DC) is a data consumer who enforces data applications services according to policies defined by DO. If a usage request is allowed, DC receives user data and keys through CH. After a data-sharing process is completed, a notification is sent to the data owner through CH. Formally, DC=<CH,PEP,d-PDP,PIP,AMs,TA,Application>, where the Application denotes a subject, viz. data requester, and also performs computing functions by using the data from the TA that excluding sensitive information, then returns a result showing on the smart phone of DO. More precisely, we denote CH in DC as $\mathrm{CH}_{\mathrm{DC}}$, thus $\mathrm{CH}_{\mathrm{DC}}$=<$H_{pol}$,$H_{sat}$,$H_{sub}$,$H_{not}$>, where H_{pol} receives policy and enforce a distributed evaluation directly through the dPDP. H_{sat} updates subject attributes. H_{sub} sends data requests, and H_{not} sends notifications to data owners.

Trusted Agent (TA) is an agent of a TEE belonging to DC, which interacts with an external untrusted environment. In a data-sharing process, the user data received through sharing actions is stored by PIP, and the key is directly delivered to TA for storage in the TEE. It is worth noting that the entire process of providing specific application services and distributed policy evaluation is completed in TEE, and interactions required with the external environment also is completed by TA. If a violation occurs in distributed evaluation processes, CH or TA is triggered to send alerts and notifications to the DO.

Blockchain Service is a service provided by blockchain technology to build a trust relationship among DO, DS, and DC, in charge of providing distributed services including policy decision, policy enforcement, and policy administration.

Next, we illustrate **the enforcement process of DUCE** as follows.

Initialization Phase refers to an initial preparation of DUCE, including service, communication, and data preparation. First of all, the distributed policy-decision services, i.e., nodes of a permissioned blockchain, and the enforcement environments need to be prepared by DS and DC. Then, the communication ability namely CH with data owners need to be prepared in DS and DC, such as P2P network or an internal protocol of DS or DC. Next, a TEE needs to be provided in DC, in which TA can interact with DS and DO. Finally, the user data uploaded by devices should be prepared and retrievable in DS. In this paper, to protect user privacy, we default that the device data is encrypted for storage in DS and can only be decrypted with an authorization of data owners.

Enforcement Phase is divided into the following four segments.

Authorization. After the initialization is completed, the application service that DO wants triggers DC to initiate a data request to DS in step 1. After the request is received, CH retrieves relevant information of authorization on blockchain by executing smart contracts in step 2. There is no authorization information related to this DC since it is in initial status. We assume that authorization in DUCE is instant, i.e. permission will be automatically revoked if the relevant operation is not performed within a limited time, so DC should request authorization every time before an access. Then, DS initiates an authorization request through CH to ask DO for authorizations. After receiving the request through a smart phone, the user authorizes with a defined policy, namely rule in 3. The policy is received and translated into a smart contract, and issued on blockchain by CH in 4. Simultaneously, CH knows the authorization through synchronization in 5.

Operation. The policy-related information on blockchain, <subject,object, right> is used to make a decision to access DC to data by dPDPs. The data is shared to DC by DS through the CH in 6. After receiving the data, DC performs related operations by TA in TEE in 9. The relevant records of operations in 7 are uploaded to blockchain for storage and evaluation subsequently in 8 by CH.

Evaluation. The records stored in an *Operation* are analyzed by executing smart contracts, either make a decision to update the subject, object, and environmental attributes by dPDPs via CH in 8 or enforce a revocation.

4.4 Administrating The Enforcement Model

We use DUCE to realize usage control. In particular, we translate the policy so that we can evaluate the policy by executing smart contracts.

XACML [10] standard is currently the most popular expression policy language to express attribute-based access control (ABAC) policy. The core components of UCON include subject attributes and object attributes. Therefore, we use XACML to help data owners to define policy, and an XACML-based UCON policy is shown in Policy 1.

Policy 1 Usage Control

```
<Policy PolicyID="UCONPolicy">
  <Rule Effect="Permit" RuleID="usage-data-consumer-rule">
    <Target> <AllOf>
      <Match
        MatchID="urn:oasis:names:tc:xacml:1.0:function:date-greater-than">
        <AttributeValue
        DataType="http://www.w3.org/2001/XMLSchema#date">2021-02-08
        </AttributeValue>
        <AttributeDesignator
        AttributeId="urn:oasis:names:tc:xacml:1.0:resource:data-collected-date"
        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:
                    user-wristband-data"
        DataType="http://www.w3.org/2001/XMLSchema#date"
        Issuer="ID_DO"
        MustDeleteAfterUse="true"
        MustMeetSystemCondition="true"/>
      </Match>
    </AllOf> </Target>
  </Rule>
</Policy>
```

Algorithm 2 UCON Policy Translation

```
1: procedure TRANSLATE($xa, sc$)    ▷ translate a XACML file into a smart contract
2:     $rule \leftarrow xa.Rule$
3:     $s \leftarrow rule.Target$
4:     $res' \leftarrow retrieve(rule.\{Category, AttributeID, AttributeValue, Issuer\})$
5:     while $res \in res'$ do traversed    ▷ traverse $res$ to find the data
6:         if ($res$.AttributeValue $\in$ $rule$.MatchID) then
7:             $o \leftarrow res$.AttributeValue
8:     $b \leftarrow rule$.MustDeleteAfterUse
9:     $c \leftarrow rule$.MustMeetSystemCondition
10:    $r \leftarrow rule$.Effect    ▷ parse xacml file to object successfully
11:    $sc \leftarrow$ constructSC()    ▷ begin to construct a smart contract to load object
12:    $uconManager \leftarrow uconManagerContract(rule$.Issuer)
13:    if ($uconManager.AttributeValue \in o$) then
14:        if ($r$=="Permit" && $b$=="ture" && $c$=="true") then
15:            $uconManager$.Permit ← "true"
16:        else
17:            $uconManager$.Permit ← "false"
18:        $sc \leftarrow uconManager$
19:    return $sc$    ▷ translate XACML file into a smart contract successfully
```

In Policy 1, we use "UCONPolicy" to identify the policy. Multiple rules can be defined in one policy, and the Rule tag is used, which contains the specific effect "Permit", and the rule identification namely RuleID. A Rule consists of multiple Target. Each Target can have different and user-defined scope to represent a specific subject of data usage, such as, AllOf means all subjects that meet the following conditions whereas TopOf means the first subject, etc. Following the Target is the Match, which represents a specific object and usage permission and is identified by MatchID. MatchID indicates an object filter, such as date-greater-than is a function to retrieval the user data with creation and updating date met the condition, parameters include tag AttributeValue and tag AttributeDesignator. AttributeValue defines attribute value used as an actual parameter, i.e., date greater than 2021-02-08. AttributeDesignator represents how to single out the specific object, parameters include tag AttributeId which refers to attributes of a resource, tag Category which describes the category of the resource, tag DataType which refers to the type of resource, tag Issuer which refers to the maker or authorizer of this rule (viz., data owner), tag MustDeleteAfterUse which refers to the obligation functional predicate for an evaluation of "delete-after-use" rule in UCON policy model, and tag MustMeetSystemCondition which refers to the condition functional predicate for an evaluation of usage "in TEE". In short, the UCON policy is briefly described as the "Permit" permission is granted to the subject who obeys the two conditions "delete after use" and "use and delete in TEE" by the data owner (Issuer), the subject is allowed to use the user data in "user-wristband-data" category and the date is after "2021-02-08".

Moreover, we use Blockchain Service to make participants trust each other in DUCE. Smart contract is a core component that supports trusted and distributed policy decision-making services of blockchain technology, it ensures functionality and security of policy through automated execution and evaluation. We translate UCON policy into smart contracts, as shown in Algorithm 2.

In Algorithm 2, firstly the XACML file is parsed to get the rule $rule$ and the corresponding authorized subject s. Line 4 shows that the approximate usage object is located by using the parameter to retrieve, and the result is assigned to res'. Starting from line 5, by iterating the result returned by the previous step, the data that meets the conditions are assigned to the object o. Then, the obligation and condition functional predicate, permission are recorded to b, c and r. The XACML file is parsed successfully. Starting from line 11, the function constructSC() is used to construct a smart contract, and a rule manager

Notification. This segment is enforced by CH in PEP and TA. Once the operation the user follows is completed, or related evaluation is triggered, the TA or the CH should notify the user regardless of the conditions met in 5. Definitely, TA should send a notification to users when all the above segments are completed in 10. Thus to be completely visible and controllable to the user.

Administrating The Enforcement Model. In particular, we translate the policy so that we can evaluate the policy by executing smart contracts. We use XACML [3], the most popular expression policy language, to help data owners to define policy, as shown in Policy 1, "UCONPolicy" is used to identify the policy, which is briefly described as the "Permit" permission is granted to the subject who obeys the two conditions "delete after use" and "use and delete in TEE" by the data owner (Issuer), the subject is allowed to use the user data in "user-wristband-data" category and the date is after "2021-02-08".

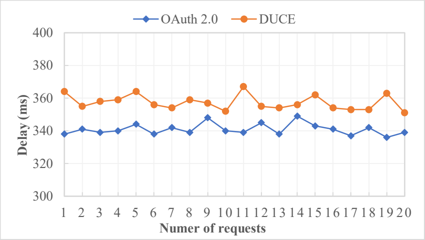
Moreover, smart contract is a core component that supports trusted and distributed policy decision-making services of blockchain technology, it ensures functionality and security of policy through automated execution and evaluation. We translate the policy into smart contracts, that are issued on Blockchain Service through CH by DS and waits execution triggers for data usage decision-making, as shown in Algorithm 2.

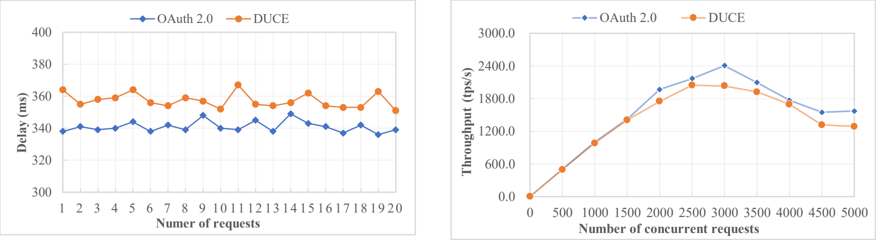
5 Performance Evaluation

Implementation. We prototype the DUCE authentication and authorization service built upon SpringFramework. In particular, we utilize smart contracts enabled in FISCO BCOS[4] to realize Blockchain based authorization in DUCE. We store the accessToken on the blockchain through the CRUD feature of FISCO BCOS, and use Solidity to realize the accessToken authentication service. We replace the authentication service logic of OAuth 2.0 as the DUCE service, which ensures that the authentication process in DUCE is tamper-proof. A user can use an acccessToken stored in the blockchain to get authorized.

Experiment Setup. Our prototype is deployed on the Alibaba Cloud Elastic Compute Service. Also, we use the default OAuth module as a baseline, to im-

[4] http://www.fisco-bcos.org

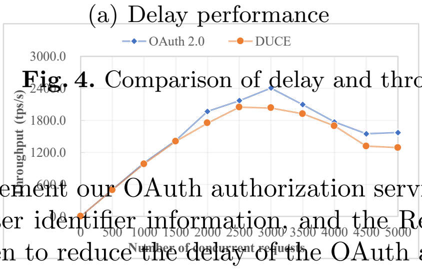


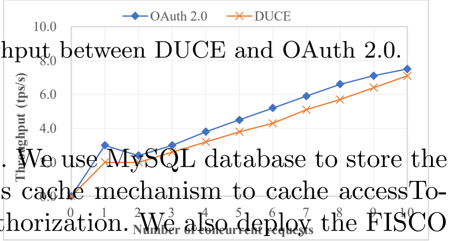


(a) Delay performance

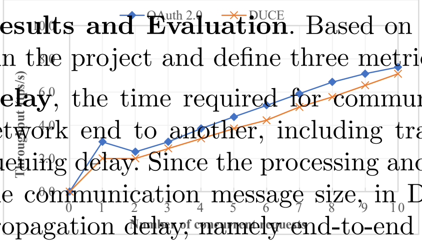
(b) Throughput performance

Fig. 4. Comparison of delay and throughput between DUCE and OAuth 2.0.





plement our OAuth authorization service. We use MySQL database to store the user identifier information, and the Redis cache mechanism to cache accessToken to reduce the delay of the OAuth authorization. We also deploy the FISCO BCOS blockchain service of DUCE on the same cloud server.



Results and Evaluation. Based on the above implementation and setup, we run the project and define three metrics to evaluate performance.

Delay, the time required for communication messages transmitting from one network end to another, including transmission, propagation, processing, and queuing delay. Since the processing and queuing delay are mainly determined by the communication message size, in DUCE, we focus on the transmission and propagation delay, namely end-to-end transfer delay.

Throughput, the maximum request number that the system can handle per unit time, We focus on authorization and authentication throughput in DUCE.

To demonstrate the effectiveness of DUCE, we first use Postman to test the transfer delay, as shown in Fig. 4 (a). Then, we use JMeter to test the throughput of DUCE, as shown in Fig. 4 (b).

Discussion. According to the above experimental results, we find that the realization of authentication and authorization by using blockchain services increases the delay and decreases the throughput. In the experiment, the additional blockchain services requires more time (i.e., 350 ms in the OAuth 2.0 system and 370 ms in the DUCE) to process end-to-end communication than the OAuth 2.0 system. The choice of OAuth 2.0 may be limited to the experimental configuration, and the throughput performance is around 2400tps/s. In DUCE, the selection of different blockchain may result in different throughput, and the throughput reaches about 2000tps/s in FISCO BCOS. Therefore, as the circumstance that there is a same ratio of the peak throughput (y-axis) to number of concurrent requests (x-axis) in DUCE and OAuth 2.0, i.e., both are 80%, the decrease of about 17% (less than 20%) is within the acceptable range. In other words, we demonstrate that compared to the existing widely used solution namely OAuth, DUCE does not introduce excessive overhead, while preventing user privacy from being compromised.

6 Related Work

Privacy preserving of static data refers to a protection of static storage data, methods include the access control mechanism, the encrypted storage and the anonymization of sensitive information. Both academic researchers and industry cloud service providers, such as Microsoft, Amazon, Google, have deployed CEIoT platforms and novel access control models. Google [11] developed GCP-IoTAC, a fine-grained access control model based on attribute extensions, and demonstrated two main use cases which are more privacy-conscious of IoT. Fernández et al. [9] designed a data collection and data sharing model based on the DataBank architecture and implemented it on an open-source platform Privasee. Liu et al. [16] proposed BC-SABE, a blockchain-assisted mechanism with effective revocation and decryption functions based on attribute-based encryption. Xu et al. [30] designed the Key Compromise Resilient Signature (KCRS) system. To protect IoT device data, an authentication framework based on a decentralized ledger namely DIoTA [29] is proposed. Patil et al. [25] used the concept of anonymous tokenization to make up for the shortcomings of current communication technology that cannot protect the anonymity of users.

Privacy preserving of dynamic data refers to prevention of privacy leakage due to improper data usage during data-sharing transmission and computing, the main prevention methods include Federated Learning, Homomorphic Encryption, and Trusted Execution Environment. In order to balance utility and privacy, Ramesh et al. [26] proposed a framework namely proxy re-ciphering as a service that using Fully Homomorphic Encryption and Chameleon Hash to customize the solution to ensure long-term computing with privacy-preserving of device data. Federated Learning can be used to train a global machine-learning model using data distributed across multiple sites without data movement. Choudhury et al. [7] proposed a grammatical method, different from differential privacy, that can support privacy-preserving at the defense level while maximizing the effectiveness of the model. Zhang et al. [31] proposed a system solution called BatchCrypt for the cross-silo federated learning system, which can ensure update of the local gradient is concealed when is aggregated. Zhang et al. [32] designed Cerberus by combining blockchain technology provided distributed data storage and TEE for state maintenance, data storage and off-chain computing in a computing scenario outsourced to edge nodes.

Moreover, Lazouski et al. [15] designed U-XACML to express UCON, implementing a prototype system for evaluation. Marra et al. [14] proposed a realization of usage control in a smart home use case. Ma et al. [17] proposed BlockBDM, a decentralized trust management scheme for IoT big data.

7 Conclusion

Utilizing blockchain or DLT to build a trust relationship between participants in a data-sharing scenario to prevent user privacy leakage is one of the most popular methods. Whereas the IoT device data contains sensitive or private information, combining two new computing models, cloud computing and IoT, can provide users with efficient services with privacy-preserving. To address the problem that applications service or data consumer violates articles such as "*Right to erasure*" in GDPR and leads to user privacy disclosure, we propose DUCE, a trusted and distributed enforcement architecture. In DUCE, blockchain is used to enforce distributed usage control policy to make decisions by distributed PDPs and PEPs, the policy is defined in XACML and translated into smart contracts for automatic execution and evaluation. Utilizing a TEE to limit obligations and conditions, we demonstrate the enforcement process of DUCE, and conducted functional and performance evaluations by comparing our prototype with OAuth 2.0 system. However, DUCE integrates and relies on TEE, thus the protection of user data depends on the security strength of cryptography and TEE. In future work, we devote to research more secure and trusted enforcement models, and figure out methods for encrypted data protection.

References

1. Almolhis, N., Alashjaee, A., Duraibi, S., Alqahtani, F., Moussa, A.: The security issues in iot-cloud: A review. In: 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA). pp. 191–196. IEEE (2020)
2. Alzahrani, B., Irshad, A., Alsubhi, K., Albeshri, A.: A secure and efficient remote patient-monitoring authentication protocol for cloud-iot. International Journal of Communication Systems **33**(11), e4423 (2020)
3. Anderson, A., Nadalin, A., Parducci, B., Engovatov, D., Lockhart, H., et al..: extensible access control markup language (xacml) version 1.0. OASIS (2003)
4. Bhatt, S., Patwa, F., Sandhu, R.: An access control framework for cloud-enabled wearable internet of things. In: 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC). pp. 328–338. IEEE (2017)
5. Bhatt, S., Sandhu, R.: Abac-cc: Attribute-based access control and communication control for internet of things. In: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies. pp. 203–212 (2020)
6. Chen, R., et al.: Trust-based service management for mobile cloud iot systems. IEEE transactions on network and service management **16**(1), 246–263 (2018)
7. Choudhury, O., Gkoulalas-Divanis, A., Salonidis, T., et al.: Anonymizing data for privacy-preserving federated learning. arXiv preprint arXiv:2002.09096 (2020)
8. De Donno, M., Tange, K., et al.: Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog. Ieee Access **7**, 150936–150948 (2019)
9. Fernández, M., Franch Tapia, A., Jaimunk, J., et al.: A data access model for privacy-preserving cloud-iot architectures. In: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies. pp. 191–202 (2020)
10. Foughali, K., Fathallah, K., Frihida, A.: Using cloud iot for disease prevention in precision agriculture. Procedia computer science **130**, 575–582 (2018)
11. Gupta, D., et al.: Access control model for google cloud iot. In: (BigDataSecurity), (HPSC) and (IDS). pp. 198–208. IEEE (2020)
12. He, W., Yan, G., Xu, L.: Developing vehicular data cloud services in the iot environment. IEEE transactions on industrial informatics **10**(2), 1587–1595 (2014)
13. Kianoush, S., et al.: A cloud-iot platform for passive radio sensing: Challenges and application case studies. IEEE Internet of Things Journal **5**(5), 3624–3636 (2018)

14. La Marra, A., Martinelli, F., Mori, P., Saracino, A.: Implementing usage control in internet of things: a smart home use case. In: 2017 IEEE Trustcom/BigDataSE/ICESS. pp. 1056–1063. IEEE (2017)
15. Lazouski, A., Martinelli, F., Mori, P.: A prototype for enforcing usage control policies based on xacml. In: International Conference on Trust, Privacy and Security in Digital Business. pp. 79–92. Springer (2012)
16. Liu, S., Yu, J., et al.: Bc-sabe: Blockchain-aided searchable attribute-based encryption for cloud-iot. IEEE Internet of Things Journal **7**(9), 7851–7867 (2020)
17. Ma, Z., et al.: Blockchain-enabled decentralized trust management and secure usage control of iot big data. IEEE Internet of Things Journal **7**(5), 4000–4015 (2019)
18. Maesa, D., et al.: Blockchain based access control. In: IFIP international conference on distributed applications and interoperable systems. pp. 206–220. Springer (2017)
19. Maesa, D., Mori, P., Ricci, L.: A blockchain based approach for the definition of auditable access control systems. Computers & Security **84**, 93–119 (2019)
20. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Tech. rep. (2019)
21. Nandakumar, L., Sangeeth, M., et al.: Real time water contamination monitor using cloud, iot and embedded platforms. In: 2020 International Conference on Smart Electronics and Communication (ICOSEC). pp. 854–858. IEEE (2020)
22. Neagu, G., et al.: A cloud-iot based sensing service for health monitoring. In: 2017 E-Health and Bioengineering Conference (EHB). pp. 53–56. IEEE (2017)
23. Ouaddah A., E.A., A., O.: Towards a novel privacy-preserving access control model based on blockchain technology in iot. In: Europe and MENA Cooperation Advances in Information and Communication Technologies. p. 520 (2017)
24. Park, J., Sandhu, R.: The uconabc usage control model. ACM transactions on information and system security (TISSEC) **7**(1), 128–174 (2004)
25. Patil, S., Joshi, S., Patil, D.: Enhanced privacy preservation using anonymization in iot-enabled smart homes. In: Smart Intelligent Computing and Applications, pp. 439–454. Springer (2020)
26. Ramesh, S., et al.: An efficient framework for privacy-preserving computations on encrypted iot data. IEEE Internet of Things Journal **7**(9), 8700–8708 (2020)
27. Sandhu, R., Samarati, P.: Access control: principle and practice. IEEE communications magazine **32**(9), 40–48 (1994)
28. Stergiou, C., Psannis, K., Kim, B., Gupta, B.: Secure integration of iot and cloud computing. Future Generation Computer Systems **78**, 964–975 (2018)
29. Xu, L., Chen, L., Gao, Z., et al.: Diota: Decentralized-ledger-based framework for data authenticity protection in iot systems. IEEE Network **34**(1), 38–46 (2020)
30. Xu, L., Chen, L., Gao, Z., Fan, X., Doan, K., Xu, S., Shi, W.: Kcrs: A blockchain-based key compromise resilient signature system. In: International Conference on Blockchain and Trustworthy Systems. pp. 226–239. Springer (2019)
31. Zhang, C., Li, S., Xia, J., Wang, W., Yan, F., Liu, Y.: Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In: 2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20). pp. 493–506 (2020)
32. Zhang, D., Fan, L.: Cerberus: Privacy-preserving computation in edge computing. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 43–49. IEEE (2020)