On Blockchain-Based Anonymized Dataset Distribution Platform

Shinsaku Kiyomoto, Mohammad Shahriar Rahman, Anirban Basu
KDDI Research Inc.
2-1-15, Ohara, Fujimino
Saitama, Japan
Email: kiyomoto@kddi-research.jp

***Abstract*—In this paper, we design a distributed platform for anonymized dataset trading without any centralized trusted third party. The platform consists of peers and consensus-based blockchain mechanism, and each peer acts as a data broker, data receiver, or verifier for blockchain in a data transfer transaction. A data broker collects data from data owners under their consent for data trading. The Privacy Policy Manager (PPM) manages the consent information and confirms them on behalf of data owners, when data distribution is requested from data broker. We implement a prototype system of the platform using an open-source blockchain mechanism, Hyperledger Fabric, and provide evaluation results of the prototype system.**

***Keywords*-Blockchain, Privacy, Anonymized Dataset, Data Trading, Security Protocol**

I. Introduction

Big data analyses including personal data are a key to improving business schemes, and it would be currently believed that combination of personal records from different service domains leads personalized services to a new generation [11], [10]. Privacy is an increasingly important aspect of personal data analyses. Sensitive data, such as medical and healthcare datasets, are recognized as a valuable source of information for personalized services, medical research and statistical trend analysis [1]. However, if personal private information is leaked from the datasets, the service will be regarded as unacceptable by the original owners of the data[3]. Thus, anonymization methods have been considered as possible solutions for protecting privacy information[2]. The Act on the Protection of Personal Information in Japan is currently being reformed and the management of anonymized datasets has been defined in the new act. Under the new act, it is possible to distribute anonymized datasets in B2B scenarios without additional user consents. Anonymized datasets can be distributed between service providers by a secure channel and may not breach the privacy of individuals provided the anonymization mechanism used for generating anonymized datasets complies with anonymization regulations. One important issue on the distribution platform is how to provide verifiable transaction logs of anonymized dataset trading for data owners. The anonymized datasets do not include any identifiers; thus, it is difficult to show evidence that a dataset includes an individual's personal record. Furthermore, it might be impossible to trace who has used their personal records.

In this paper, we propose a block-chain-based distribution scheme for anonymized datasets. The platform consists of peers that act as a data broker, data receiver, and verifier of transactions, and blockchain is used for recording all transactions of anonymized datasets between a data broker to a data receiver. We implement a prototype system of the platform and evaluate it. The rest of the paper is organized as follows; related work and preliminary information are introduced in Section 2 and Section 3 respectively. A construction of a dataset distribution platform is explained in Section 4. Section 5 shows an implementation of the platform and evaluation results for performance and security analysis is presented in the section as well. We conclude this paper in Section 6.

II. Related Work

Blockchains gained popularity through their use in cryptocurrencies, Bitcoin being the most well-known. As of late, however, blockchains have stirred up interest in other types of applications. Tschorsch and Scheuermann in [22] refer to applications of blockchain technologies in a decentralized domain name system[1], abuse prevention of cloud services [21], decentralized cloud storage [24] and anonymous, distributed messaging [23]. McConaghy et al. proposed, in [13] a decentralized database using blockchains capable of a million write operations per second and storage of Peta Bytes of data with sub-second latency. Kosba et al. proposed smart digital contracts which allows anonymous parties to enforce complex agreements [9]. Kishigami et al. proposed, in [6], a decentralized blockchain-based digital content distribution system. A method for decentralized P2P software license validation using blockchain technology has been proposed in [4] to ameliorate software piracy. Qayumi proposed a blockchain-based multi-agent intelligence system for managing and collecting information from highly distributed and very large unstructured datasets in [18]. Sarr et al. discussed a blockchain-based model to handle transactions of social applications by using their access classes in [20]. A decentralized personal data management

[1]Namecoin: https://namecoin.info/.

978-1-5090-5756-6/17/$31.00 ©2017 IEEE
SERA 2017, June 7-9, 2017, London, UK

system that ensures users own and control their data using blockchain has been proposed by Zyskind et al. in [28]. In [26], the authors provide a new certificate format based on blockchain which allows a user to verify a PGP certificate using Bitcoin identity-verification transactions. Drawing on the Bitcoin protocol and an open source middleware system for volunteer and grid computing, the Berkeley Open Infrastructure Network Computing Grid[2], the blockchain finds an unexpected application with a peer-to-peer Internet-based cryptocurrency[3] that aims to provide real benefits to humanity by compensating coin miners for participating in BOINC projects, leading to advances in medicine, biology, climatology, and astrophysics by redirecting the computational power towards BOINC research. Passcard[4] uses a blockchain-tied profile to create a secure and trustless identity which can eventually be used as a sort of digital key to one's identity, potentially replacing passwords. A blockchain-based voting system[5] has also been considered to provide greater transparency in the voting process, with every vote being recorded on the blockchain. In order to verify product authenticity and ethical standards in a supply chain, [25] discusses a blockchain-based solution which is a shared, consensus-based and immutable ledger helping track the origin and the transformations undergone in a supply chain. Counterparty[6] and Factom[7] utilize blockchain to create unalterable records of rights and transactions such that they are irreversible and can be self-executing, without the need (or opportunity) for human intervention. In this paper, we present a purely distributed platform for anonymized dataset trading between service providers.

III. Preliminary

In this section, we present background information about the distribution platform.

A. k-Anonymity

A database table T in which the attributes of each user are denoted in one record is in a public domain and an attacker obtains the table and tries to distinguish the record of an individual. Suppose that a database table T has m records and n attributes $\{A_1, \ldots, A_n\}$. Each record $\mathbf{a^i} = (a_1^i, \ldots, a_n^i)$ can thus be considered as an n-tuple of attribute values, where a_j^i is the value of attribute A_j in record $\mathbf{a^i}$. The database table T itself can thus be regarded as the set of records $T = \{\mathbf{a^i} : 1 \leq i \leq m\}$. The definition of k-anonymity is as follows;

k-Anonymity[19]. A table T is said to have *k-anonymity* if and only if each n-tuple of attribute values $\mathbf{a} \in T$ appears at least k times in T.

[2]BOINC: http://boinc.berkeley.edu/.
[3]GridCoin: http://www.gridcoin.us/.
[4]Onename: https://onename.com/.
[5]Voting on the Blockchain: https://bitcoinfoundation.org/.
[6]Counterparty: http://counterparty.io/.
[7]Factom: http://factom.org/.

Our data distribution platform focuses on general anonymized datasets; but, a k-anonymized dataset is a typical example of the datasets.

B. Privacy Policy Manager (PPM)

The Privacy Policy Manager (PPM) [8] manages an individual's privacy settings and provides information for controlling flows of private data according to those privacy settings. The PPM is built on an entity in a domain, and each separate domain has at least one PPM. Individuals register their privacy settings with a PPM located in a domain to which the individual belongs, and configure the actions to be taken when a service provider requests private information the delivery of which violates the privacy settings. For example, the PPM asks an individual whether the private information should be sent, when the act of sending the information is against the privacy settings of the individual. Individuals are users of a PPM on their domain, and an individual's privacy settings is managed in the database of the PPM, and information flow is controlled based on the privacy settings. Even though the main role of the PPM is the management of privacy settings, the PPM provides another function for tracing of personal data distribution in this paper. The PPM stores a transaction log of personal data transfer from individuals to service providers, and dataset IDs including the individual's personal record. Detailed protocol between the PPM and a data broker is described in later sections.

C. Cryptographic Primitives

We use the following cryptographic primitives in our distribution platform:

- *Hash Function.* $H(x)$ is a one-way cryptographic hash function using a digest value of x.
- *Symmetric Key Encryption.* $Enc(x, y)$ is an symmetric key encryption algorithm for encrypting y using a secret key x. The encrypted message can be decrypted using the same secret key.
- *Digital Signature.* $Sig_x(y)$ is a digital signature of the message y where the signer is x. The signature scheme uses a private key Pr_x of the signer x and anyone can verify the signature using a public key Pu_x of the signer x. A public key certificate $Cert_{Pu_x}$ is issued for the public key Pu_x and the validity of the public key is ensured by the certificate. An existing signature algorithm such as ECDSA [5] is used in the platform.

D. Blockchain Technology

The blockchain technology has drawn significant attention from researchers, engineers, economists and entrepreneurs in recent years. It is a decentralized and byzantine fault-tolerant transaction protocol having the potential to become the backbone of Internet-based connectivity. The blockchain is a general append-only ledger containing all transactions

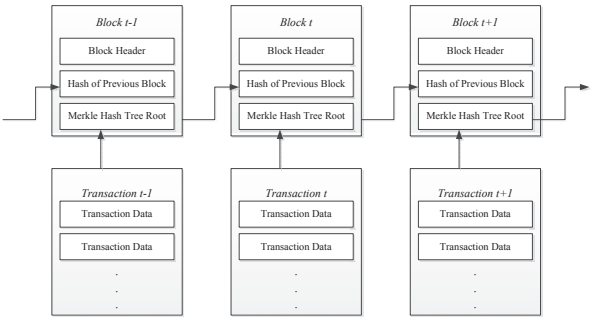

Figure 1. Blockchain Operation

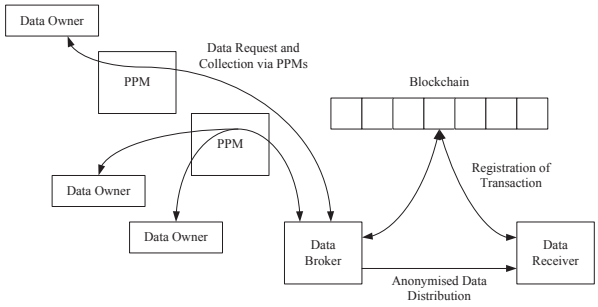

Figure 2. Overview of the Platform

performed since the system started to operate. Such an approach implies that the size of the blockchain is constantly increasing and, for that reason, scalability is the biggest challenge that the system faces.

The blockchain is freely replicated and stored in different nodes of the Bitcoin network, making the Bitcoin a completely distributed system. Transactions in the blockchain are inserted at time periods. Once all the newly generated transactions of the system are collected, they are compiled together in a data structure, called blocks. That block is then inserted at the top of the blockchain. A transaction is called 'confirmed transaction' once the block containing that transaction is inserted in the blockchain. It is possible to check a confirmed transaction in order to prevent double-spending attacks. Addition of a block in the blockchain is a hard problem since it consumes a lot of time, resources and work. Each block in a blockchain is indexed using its hash value. Also, the hash value of the previous block is contained in each new block that is being inserted in the chain. This process ensures that modifying a block from the middle of the chain would be as hard as modifying the rest of the blocks in the chain such that all the hash values are matched. A general blockchain operation mechanism is explained in Figure 1.

We use consensus-based block-chain technologies: that is service providers (data brokers and data receivers) cooperate to make a consensus for each block of the blockchain. Hash values of all transactions are included in Merkle tree hashes[14].

IV. Dataset Distribution Platform

In this section, we present the design of the distribution platform.

A. Overview of Our Scheme

Figure 2 shows an overview of an anonymized datasets distribution scheme. In an anonymized datasets distribution scheme, five entities exist as follows:

- *Data Owner* is an individual who provides his/her personal data to a data broker.
- *Data Broker* is a service provider who collects personal data with a data owner's consent, generates an anonymized dataset and distributes it to data receivers.
- *Data Receiver* is a service provider who uses anonymized datasets and improves their services, provides analysis results, and distributes modified anonymized datasets to other data receivers. Circulation of anonymized datasets from one to another is assumed to be covered in the distribution scheme as well.
- *Blockchain* is a distributed management system of transaction records in the scheme. All service providers (data brokers and data receivers) contribute blockchain operations.
- *PPM* is a privacy policy manager that is described in Section 3. The PPM manages the consent information and confirm them on behalf of data owners, when data distribution is requested from data broker.

The data broker collects data from data owners under confirmations of data owner's consent by PPMs, and then the data broker executes a data transfer protocol with a data receiver. The data broker and the data receiver send transaction data to blockchain in order to establish the transaction. Some peers contribute to make a new block of the blockchain using a consensus-based blockchain protocol.

B. Blockchain Perspective

Our platform uses blockchain as a building block, instead of a central server managed by a trusted third party. To use the blockchain, a scalable and cost-effective platform can be realized. In our platform, players (Data Brokers and Data Receivers) provide a mutual surveillance service using the blockchain technology; thus, no central server managed by a trusted third party (TTP) is required and the cost for the central server can be reduced.

C. Security Requirements

The following security requirements should be considered in a design of the platform:

a) **Verifiability of Data Transaction.** All successful transactions between data broker and data receiver

should be publicly verifiable for any entity.

b) **Secure Transaction.** A dataset should be securely transfered from a data broker to a data receiver.

c) **Traceability of Anonymized Dataset.** Data owners should be able to trace an anonymized dataset that includes their personal data.

d) **Anonymity of Dataset.** The dataset should be anonymized and no reasonable advantage for re-identification occurs during data transfers.

We assume that the system has no vulnerability for its implementation. Vulnerabilities such as software bugs are out of the scope in this paper.

D. Anonymized Dataset Generation

An anonymized dataset is generated from an individual's personal data. The data broker collects personal data with the data owner's consent. We use the Privacy Policy Manager (PPM) for obtaining user consent from data owners. The steps to generate an anonymized dataset is described as follows:

1) The data broker contacts data owners via their own PPMs and obtains consent for anonymized data distribution.
2) Data owners send their personal data to the data broker.
3) The data broker generates an anonymized dataset D_a and its dataset ID ID_{D_a}. The dataset ID ID_{D_a} and a hash value $h_1 = H(D_a)$ are notified to the PPMs via secure channels. The PPM stores the dataset ID and the hash value in the transaction record of each data owner. The data owner can obtain a dataset ID from the PPM.
4) PPM generates a digital signature $Sig_{PPM}(h_0)$ where $h_0 = H(ID_{D_a}||h_1)$. The signature is sent to the data broker as acknowledgment for D_a.

E. Protocol between Data Broker and Data Receiver

We assume that all data brokers and data receivers have a public-private key pair $\{Pu_x, Pr_x\}$ and a public key certificate (PKC) of the public key $Cert_{Pu_x}$ for appending their digital signature of y $Sig_x(y)$. A trading protocol between a data broker and data receiver is described as follows:

1) A data broker A generates an anonymized dataset D_a with dataset ID ID_{D_a} described in the previous section, and encrypts it as $Enc(k_a, D_a)$ where a temporal key k_a is randomly generated, and $Enc(x, y)$ is a symmetric key encryption of y using a secret key x. The data broker calculates two hash values: $h_1 = H(D_a)$ and $h_2 = H(k_a)$. Then, the data broker registers the two hash values and a digital signature of the hash values $Sig_A(h_1||h_2)$ to a blockchain server as a transaction, where the symbol $||$ means concatenation of data. Note that the length of the hash values is at least double the length of the temporal key due to avoid collision of two hash values; for example, SHA-256 [16] is used for generating hash values where 128-bit key encryption by AES [17] is used for the encryption of the anonymized dataset. The signatures of PPMs $Sig_{PPM}(h_0)$ have been received from PPMs, where the transaction is an initial trading of D_a. The signature $Sig_{PPM}(h_0)$ is sent to the blockchain as well.
2) The data broker A sends an encrypted data $Enc(k_a, D_a)$ and ID_{D_a} to a data receiver B.
3) The data receiver B computes the hash value of the encrypted data as $h_3 = H(Enc(k_a, D_a))$ and generates its signature $Sig_B(h_3)$; then sends the hash value h_3 to the blockchain server as a transaction.
4) The data receiver verifies the transaction chain of the dataset ID. Details of this step are explained in a later subsection.
5) The block including a hash value of the transaction data $(h_1, h_2, h_3, Sig_A(h_1||h_2), Sig_B(h_3), (Sig_{PPM}(h_0)))$ in the blockchain is authorized, and the transaction is confirmed. The signatures of PPMs $Sig_{PPM}(h_0)$ are attached where the transaction is an initial trading of D_a. The block is broadcasted to all nodes and the transaction data is distributed to an appropriate node.
6) The data receiver receives h_1 from the blockchain.
7) After confirmation of the transaction, the data broker A sends the temporal key k_a to the data receiver B using a secure channel.
8) The data receiver confirms $H(k_a) = h_2$ in the transaction data, decrypts the encrypted anonymized dataset, and obtains D_a. The data receiver confirms $H(D_a) = h_1$ in the transaction data.

Figure 2 shows the protocol between a data broker and data receiver.

F. Dataset ID generation and Verification of Transaction Chain

Anonymized data distribution between service providers is assumed in the platform, and a unique identifier for each transaction is required for searching a certain transaction. A service provider attaches a dataset ID to an anonymized dataset for identifying each transaction. At the initial transfer of the anonymized dataset, the dataset ID is generated as $ID_{D_a} = r||n||H^n(r||h_1)$, where r is a hash value of concatenated data of all signatures ($r = H(\sum Sig_{PPM})$) from PPMs that provide personal records to the anonymized dataset. The symbol $H^n(x)$ means n-times hash calculation of x. A dataset ID for $t+1$-times transfer is calculated as $ID_{D_a} = r||n-t||H^{n-t}(r||h_1)$, and if $n = t$, the anonymized dataset is no longer transferred to the other service provider. Thus, the limit[s] of transfer can be controlled by the configuration of n. In a transaction chain verification,

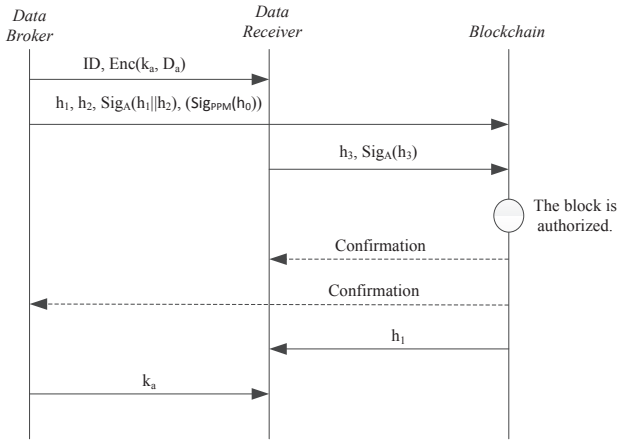

Figure 3. Transaction Protocol

a data receiver calculates $ID_{D_a} = r||n||H^n(r||h_1)$, checks the validity of $Sig_{PPM}(h_0)$ and ID_{D_a} on transaction data, and confirm a hash value of the transaction data in a block of the blockchain. The data receiver can be traceable on the transaction data by computing all hash values for $\{n, n-1, n-2, ..., n-t\}$ and a hash value of the datasets h_1. Distributed Hash Table (DHT) techniques such as Kademila [12] could be applied to realize a more efficient search for a dataset ID in a distributed database. All transaction records consist of four types of data: dataset ID, date and time of the transaction, a block number of the blockchain, which is included in the transaction record, and transaction data. The n-bit prefix of the dataset ID is used as a node ID, where 2^n nodes join the blockchain operations. The service provider of node ID x retains transaction records of the prefix x. A search query is routed to a node according to the prefix of the dataset ID. It is difficult to continuously generate an intentional value for dataset IDs under the assumption that the cryptographic hash function is a secure one-way function.

Case of Datasets Mash-Up. Datasets mash-up means that a data broker obtains two or more datasets and generates a new dataset from the obtained datasets in order to increase the value of the dataset. A new anonymized dataset generated from some different anonymized datasets is considered to be an extension of the general scheme. In this case, a data broker generates a dataset ID and obtains signatures of PPMs related to the datasets, and registers them as an initial transfer. PPMs store the new datasets ID with old dataset ID and the hash value of the new dataset as well. Thus, the dataset is still traceable.

Case of Multiple Distribution. When a data broker wants to distribute the same datasets to another data receiver (or more than two data receivers), a dataset ID for the data receiver should be generated as a new datasets ID and the signatures of PPMs that produce records of the datasets need to be obtained. The data broker traces the chain of dataset IDs to the initial ID, finds the PPMs, and requests new signatures for the new dataset ID. The PPMs store the dataset ID and the hash value in the transaction record for each data owner.

Secure Distribution Untraceable for Third Parties. If we use a keyed-hash function to generate dataset IDs, a transaction is untraceable for third parties; only entities who has a secret key for computing the keyed hash values can trace transactions. This extension restricts entities who can trace transactions for a certain dataset D_a, when only entities trading a dataset D_a and related PPMs share the secret key. This scheme is considered as an optional scheme for our distribution platform.

G. Verification of Anonymized Datasets

The validity of an anonymized dataset is confirmed when the data receiver receives the anonymized dataset. An honest data receiver notifies a third party (such as a personal information protection commission) who is responsible for privacy protection and fairness of data trading, when the data receiver receives an invalid dataset. All valid datasets can be traceable to an initial transaction including PPM's signatures on the blockchain. If the trace fails, the dataset is determined to be invalid. Dataset IDs are used for efficient tracing; the data receiver only computes t hash values and confirms the chain of the transaction in the case of $t + 1$-times transfer. Hash values of anonymized datasets and digital signatures can be verified using PKCs as well.

Tracking by Data Owners. Data owners can obtain dataset IDs from their PPMs and trace data transfers just as the data receiver can. Data owners first search transaction records in their own PPM and then calculate dataset IDs; then they search transaction data in blockchain, by using the dataset IDs.

H. Privacy Risk Analysis on Anonymized Dataset

A data broker generates an anonymized dataset at the first step of the data transfer protocol. Before sending it, the data broker should confirm the privacy risk of the anonymized dataset. There are several existing techniques and indices for evaluation of privacy risks. A basic guideline is k-anonymity; the minimum k should be defined as a regulation of data distribution and the data broker would generate an anonymized dataset to meet the regulation. A common regulation should be defined for the dataset distribution platform.

I. Invalid Trading Inspection

In the distribution platform, a consensus-based blockchain is used for generating scalable verifiable transaction logs. In the consensus protocols, the trustworthiness of that entities who are joining the consensus is the important factor for ensuring the validity of the consensus. We define

the trustworthiness of each entity and the trustworthiness increases according to the contributions of entities. There are several contributions for the distribution platform. Successful consensus increases the trustworthiness of entities who join the consensus. Reporting an invalid transaction or invalid an anonymized dataset is another contribution that increases the trustworthiness of entities. On the other hand, distribution of an invalid anonymized dataset by an entity would decrease the trustworthiness of the entity. The trustworthiness of each entity and reports about invalid transactions are recorded in a block. Thus, detection of invalid anonymized datasets in the system is provided by the voluntary work of entities. If the platform charges a fee of the platform use for each transaction, the fee should be distributed to entities who contribute a successful consensus of the blockchain. We do not assume a collusion attack that an entity acts both roles of and a (fake) invalid trader and an inspector of it or some entities collude to increase trustworthiness of an entity. A new entity should be identified and confirmed its validity, when the entity will join the platform and an entity who conspires invalid trading with other entities can be removed from the platform.

Underground Trading. A malicious service provider will distribute anonymized datasets without blockchain consensus or without the usage of the platform. Such underground trading is difficult to detect; but honest service providers would avoid to receive datasets from malicious service providers because it is an illegal trading. Once a malicious trading is found, the service provider can no longer join the platform and has to pay a penalty. It is not a perfect solution but a reasonable countermeasure same as real trading situations.

On the other hand, datasets may be stolen from the service providers. It is assumed that the service providers store the datasets in their systems securely. However, the datasets may be leaked by a malicious insider. Detecting an insider threat is more difficult than detecting attacks from the outside[27] before the leakage because insiders may know an organization's information protection policies and they may have authorization to access the datasets. Thus, it is important to be able to detect leaked datasets and trace the source of leakage in order to investigate insiders who may be leaking information.

When an invalid distribution dataset (leaked dataset) is found, the detection process compares anonymized datasets in valid transactions with the leaked dataset. If the leaked dataset has properties that are similar to the anonymized dataset, the detection process can output the name of the service provider who finally received the anonymized dataset using an existing technique [7]. Based on the above idea, we can build a distinguishing method: whether an anonymized dataset is illegally obtained by underground trading and whether the original dataset is from datasets being traded on the platform. For example, a data broker generates not only hash values h_1, h_2, but also the property values of a dataset when the data broker registers them to the blockchain, as follows. An anonymized dataset consists of at least k same records that have the same attribute values $a_1^i, ..., a_n^i$ if the dataset is a k-anonymized dataset. First, the number of identical records is counted for each record and the same records are removed from the datasets and then the hash values of each record are calculated (the hash values are different from each other because the datasets have different attributes). The property of the dataset is defined as the hash values of each record and the number of identical records corresponding to it. By using the property values, a leaked dataset can be distinguished.

Note that the objective of the detection process is not to provide strong evidence for the unauthorized leakage of the anonymized dataset, but to investigate a potential malicious/victim service provider. There is no perfect solution for insider threats, but the above detection mechanism may worked as a deterrent measure.

Table I
IMPLEMENTATION ENVIRONMENT

PC	Intel Core i7-6700 CPU 3.40GHz, 32GByte Memory
OS	Ubuntu 16.04 LTS
Language	go 1.7.3 linux/amd64 jquery-3.1.1 bootstrap-3.3.7
Containerization	Docker 1.12.5
Digital Signature Alg.	ECDSA (256bit key)
Encryption Alg.	AES-CTR
Hash Function	SHA-256

V. IMPLEMENTATION AND EVALUATION

In this section, we describe our implementation and performance and security evaluation results.

A. Implementation

We use Hyperledger Fabric Version 0.7.0 as a blockchain library and implement our protocols. Our implementation environment and cryptographic algorithms are shown in Table 1. All server and client programs are containerized as Docker containers. The system overview was described in Figure 2. Each peer can act both roles of data broker and data receiver. Furthermore, the peers join a consensus protocol of the blockchain. The consensus protocol is Batch PBFT in Hyperledger Fabric. The program sizes of peers are 38.53 MByte. The chaincode size is 12.70 MByte.

B. Performance Analysis

Average transaction time of protocols is shown in Table II. Each flow can be executed within 500 msec, and the total transaction time of the data transfer protocol including blockchain operations is 2616 msec.

Table II
AVERAGE TRANSACTION TIME FOR DATA TRADING

Entity	Process	Tran. Time
Data Broker	Sending ID_{D_a} and $Enc(k_a, D_a)$	470 msec
	Generating Signature and Sending it to Blockchain	433 msec
	Sending k_a	143 msec
Data Receiver	Generating a Signature and Sending it to Blockchain	469 msec
	Receiving h_1, k_a	146 msec
Total Trans. Time	-	2616 msec

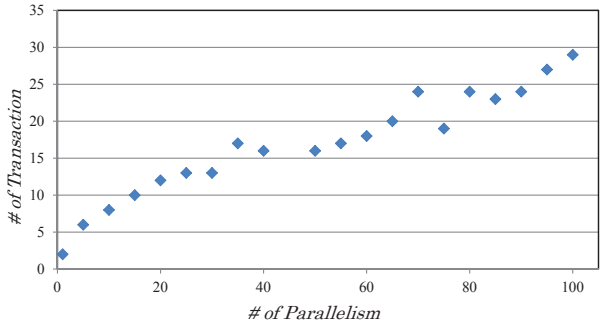

Figure 4. Number of Transaction Per Block in Parallelization

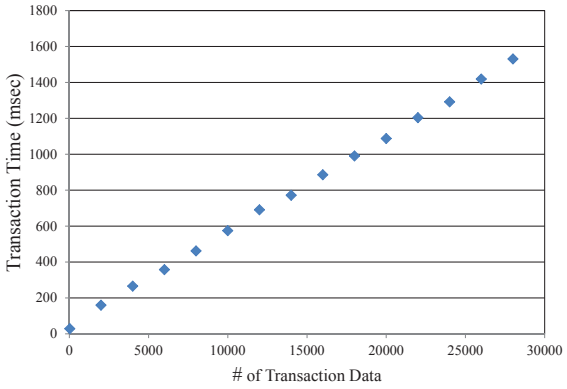

Figure 5. Processing Time for Search of Transaction Data

In our system, a new block is added to a blockchain every second, and the number of transactions stored in a block is shown in Figure 4. By increasing the degree of parallelism, the number of transaction can be increased to 29 per computing node. Figure 5 shows processing time for search of transaction data without parallelization. The processing time is the worst case for each number of transaction data in our experiment. It is feasible for practical use.

C. Security Analysis

Our designed protocols meet security requirements summarized in section IV-C. Verifiability of data transaction is ensured by a blockchain, once transaction is proved and stored in a block. Transaction data includes identifiers of the data broker and the data receiver, their digital signature, dataset ID, and hash values of the data; the transaction data stored in the blockchain is non-repudiable evidence of data trading. Thus, the security requirement a) is satisfied. A transfered dataset is encrypted and a secure channel is used in communications between a data broker and a data receiver. a decryption key is provided to a data receiver via a secure channel, after confirmation of the transaction in the blockchain. Thus, the security requirements b) has been achieved. Dataset IDs for same datasets are generated as hash-chain values; thus it is traceable and verifiable. The hash-chain mechanism achieves restriction of the number of data transfer. Thus, the platform satisfies the security requirement c). Our platform distributes anonymized datasets; thus, anonymity of dataset can be ensured; however, privacy risk should be considered in the case of data mash-up. Thus, a privacy risk should be estimated in a data transfer of mash-up data. The security requirement d) would be achieved under an sufficient privacy risk check. A method of privacy risk analysis is out of scope in the paper; there are some existing technologies for analysis of privacy risk (such as [15]).

VI. CONCLUSION

This paper presented a purely distributed system for anonymized dataset trading. We presented evaluation results of our prototype system using Hyperledger Fabric and confirmed that the platform would be practical. Our security analysis suggested that the platform achieves security requirements for data distribution. In our future research, we will consider an economic model for our platform and consider optimal settings for the platform with regard to realizing a sustainable data trading market.

ACKNOWLEDGEMENT

This work was supported by JST CREST Grant Number JPMJCR1404, Japan.

REFERENCES

[1] N. R. Adam and J. C. Wortmann. Security-control methods for statistical database: a comparative study. *ACM Comp. Surv.*, 21(4):515–556, 1989.

[2] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. k-anonymous data mining: A survey. In *Privacy-Preserving Data Mining: Models and Algorithms*. Springer-Verlag, 2008.

[3] G. Duncan and D. Lambert. The risk of disclosure for microdata. *J. Buisiness & Economic Statistics*, 7:207–217, 1989.

[4] J. Herbert and A. Litchfield. A novel method for decentralised peer-to-peer software license validation using cryptocurrency blockchain technology. In D. Parry, editor, *38th Australasian Computer Science Conference (ACSC 2015)*, volume 159 of *CRPIT*, pages 27–35, Sydney, Australia, 2015. ACS.

[5] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ECDSA). *International Journal of Information Security*, 1(1):36–63, 2001.

[6] J. Kishigami, S. Fujimura, H. Watanabe, A. Nakadaira, and A. Akutsu. The blockchain-based digital content distribution system. In *Big Data and Cloud Computing (BDCloud), 2015 IEEE Fifth International Conference on*, pages 187–190, Aug 2015.

[7] S. Kiyomoto, T. Nakamura, and Y. Miyake. Towards tracing of k-anonymized datasets. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, volume 1, pages 1237–1242, 2015.

[8] Shinsaku Kiyomoto, Toru Nakamura, Haruo Takasaki, Ryu Watanabe, and Yutaka Miyake. Ppm: Privacy policy manager for personalized services. In *Proc. of CD-ARES 2013 Workshops, LNCS*, volume 8128, pages 377–392, 2013.

[9] Ahmed E. Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *2016 IEEE Symposium on Security and Privacy, SP 2016, USA*, page To Appear, 2016.

[10] Alan Lai, Cui Zhang, and Senad Busovaca. 2-square: A web-based enhancement of square privacy and security requirements engineering. *International Journal of Software Innovation*, 1(1):41–53, 2013.

[11] Juhnyoung Lee. A view of clopud computing. *International Journal of Networked and Distributed Computing*, 1(1):2–8, 2013.

[12] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 53–65, 2002.

[13] Trent McConaghy, Rodolphe Marques, Andreas Müller, Dimitri De Jonghe, Troy McConaghy, Greg McMullen, Ryan Henderson, Sylvain Bellemare, and Alberto Granzotto. Bigchaindb: A scalable blockchain database (draft). 2016.

[14] Ralph C. Merkle. A certified digital signature. In *Proceedings on Advances in Cryptology*, CRYPTO '89, pages 218–238, 1989.

[15] Tomoaki Mimoto, Anirban Basu, and Shinsaku Kiyomoto. Towards practical k-anonymization: Correlation-based construction of generalization hierarchy. In *Proceedings of the 13th International Joint Conference on e-Business and Telecommunications (ICETE 2016) - Volume 4: SECRYPT 2016.*, pages 411–418, 2016.

[16] NIST. Secure hash standard (SHS). *Federal Information Processing Standards Publication 180-4*.

[17] NIST. Specification for the advanced encryption standard (AES). *Federal Information Processing Standards Publication 197*.

[18] Karima Qayumi. Multi-agent based intelligence generation from very large datasets. In *2015 IEEE International Conference on Cloud Engineering, IC2E 2015, Tempe, AZ, USA, March 9-13, 2015*, pages 502–504, 2015.

[19] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[20] Idrissa Sarr, Hubert Naacke, and Ibrahima Gueye. Blockchain-based model for social transactions processing. In *Proceedings of 4th International Conference on Data Management Technologies and Applications*, pages 309–315, 2015.

[21] Jakub Szefer and Ruby B Lee. Bitdeposit: Deterring attacks and abuses of cloud computing services through economic measures. In *13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pages 630–635. IEEE, 2013.

[22] Florian Tschorsch and Björn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IACR Cryptology ePrint Archive*, 2015:464, 2015.

[23] Jonathan Warren. Bitmessage: A peer-to-peer message authentication and delivery system. *white paper (27 November 2012), https://bitmessage.org/bitmessage.pdf*, 2012.

[24] Shawn Wilkinson and Jim Lowry. Metadisk: A blockchain-based decentralized file storage application. 2014.

[25] Reid Williams. How bitcoin's technology could make supply chains more transparent, 2015.

[26] Duane Wilson and Giuseppe Ateniese. From pretty good to great: Enhancing PGP using bitcoin and the blockchain. In *Network and System Security - 9th International Conference, NSS 2015, New York, NY, USA, November 3-5, 2015, Proceedings*, pages 368–375, 2015.

[27] W.T. Young, A. Memory, H.G. Goldberg, and T.E. Senator. Detecting unknown insider threat scenarios. In *2014 IEEE Security and Privacy Workshops (SPW)*, pages 277–288, 2014.

[28] Guy Zyskind, Oz Nathan, and Alex Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *2015 IEEE Symposium on Security and Privacy Workshops, SPW 2015, San Jose, CA, USA, May 21-22, 2015*, pages 180–184, 2015.