Privacy-friendly Blockchain based Data Trading and Tracking

Zhenan Wu*, Han Zheng*, Lan Zhang*, Xiang-Yang Li*
*School of Computer Science and Technology
University of Science and Technology of China

Abstract—**Data trading has become a promising way to bridge numerous data islands and benefit a wide range of data driven applications. However, due to the high risk of data privacy, the easy-to-copy feature of data, and potential malicious behaviors of users, there still lacks data trading solutions to provide privacy protection, copyright/ownership protection and auditability of user behaviors. In this work, we propose a blockchain and smart contract based framework to keep tamper-proof records of data trading transactions, which also preserves users' identity, behavior and data privacy. Data fingerprint is introduced to track data and protect data ownership. A set of protocols are also designed to prevent dishonest behaviors and ensure the security of data trading.**

I. Introduction

The rapid development of artificial intelligence and big data science and technology while providing new methods for the natural sciences and social sciences, brings us great value through making good use of massive data. While there is a strong demand for large datasets, there is still a large portion of data that is isolated and not fully utilized. To get the most out of the value of these data, the importance of data sharing and trading have become more prominent. Many data trading platforms have emerged, such as Qlik, CitizenMe, Microsoft Azure Marketplace, and DataExchange. Existing platforms typically require the data owner to submit the original data and its description to the platform for sale. As a special kind of commodity, however, data has the characteristics of easy copying, difficult to protect copyright, and high risk of privacy and security. Establishing an efficient, secure and privacy-preserving data trading market still faces enormous challenges.

Many traditional technologies like Digital Rights Management [6], watermark [4] and forensics have been designed to protect data copyright/ownership. Recently, there are some work considering user and data privacy during data trading using encryption or hash methods, e.g., [5], [7]. [3] uses an append-only bulletin board to record transactions to support acountable data trading. There still lacks a desgn which achieves effcient, scalable, acountable and privacy-preserving data trading.

In this work, we propose a privacy-friendly blockchain and smart contract based data trading framework, which achieves scalability, auditability and acountability. Our framework keeps tamper-proof records of data trading transactions. It protects users' identities, data and trading behaviors from any unauthorized party including the platform. Our design supports tracibility of data (copy right protection) by introducing the data fingerprint, which is a combination of features extracted from data and robust to minor data modifications. Our system also achieves auditability of malicious behaviors, including denying the sale or purchaseis, illegal reselling data, selling fabricated data.

II. System Overview

In this work, we design a blockchain based data trading and tracking platform, which not only supports undeniable and traceable trading, but also considers data privacy and ownership.

There are three parties involved in the data trading process.

1) **Seller:** is the owner of the data and wants to sell the data for profit. In this work, we consider two typical ways of selling data: 1) directly selling the data ownership, and the buyer will own the data and be allowed to resell the data; 2) selling the right to use the data, and the buyer will have the right to conduct allowed queries on the data and be charged for every query.
2) **Buyer:** is the data consumer, who would like to purchase the ownership or use right of his/her desired data.
3) **Platform:** provides services on a permanent basis between buyers and sellers to expedite sales. A platform can server as a broker or a reseller or both in transactions. If the platform works as a broker, it helps each buyer to find sellers who possess the buyer's desired data and gets paid for each successful sale. In this case, the seller won't give his/her original data to the platform except some description information or features of data and the platform doesn't take ownership of the data being sold. If the platform works as a reseller, it purchases data from sellers and resells the data to buyers. The platform records all transactions on the ledger.

A. Adversary Model

For the adversary model, we consider the widely adopted one with malicious users and semi-honest platform. A malicious seller could provide fabricated description information or features that do not conform to the data for sale. The seller could also resell the data whose ownership has already been sold. A malicious buyer could refuse to pay for the data he/she has received. The data trading platform is usually strictly supervised by authorities. It will follow all protocols honestly

978-1-7281-4024-7/19/$31.00 ©2019 IEEE
DOI 10.1109/BIGCOM.2019.00040

to guarantee the quality of service but try to learn as much information as possible from all transactions.

B. Design Overview

According to the role of the platform, we design our system to support the following two trading scenarios:

Scenario 1: platform servers as a broker. In this scenario, each seller hosts his/her own data and provides features of data to the platform. A buyer chooses data on the platform by browsing or searching features of data. The platform charges brokerage fee for each sale.

Scenario 2: platform servers as a reseller. In this scenario, the platform purchases data from data sellers or other data sources, and then sells data to buyers and pocket the difference as its fee.

We design our system to fulfill both functional and privacy requirements in the above two scenarios, that is our system should provides efficient data sell and purchase, sufficient protection to users' privacy as well as auditability to malicious behaviors. Our design is based on a blockchain and smart contracts. Each seller or buyer or the platform, has his/her/its own account with a pair of public key and private key, where the public key also serves as the account address. Each data for sale is bound to a unique smart contract. Each contract is kept in a special contract account with a public key as its account address. In both scenarios, our system supports two ways for the buyers to find their target data: (1) a buyer can browse various data features to find his/her desired data and obtain the contract address of the data; (2) or a buyer can upload a request including features of his/her desired data, according to which the platform searches and offers a list of contract addresses whose bound data have similar features. During a transaction, participants call different contract functions to complete a series of operations. A transaction could be a sale of data ownership or a sale of right to use data. Even for the same data, due to the parallelism of calling contract, the parallelism of transactions can be guaranteed. Specially, unlike Bitcoin or other blockchain-based pure-on-chain transaction platforms, our system combines an on-chain distributed ledger with an off-chain centralized data-contract table and a send-receive data transmission record board. Our whole design ensures the security of transaction records meanwhile provides the convenience of quick retrieval. Since each invoked contract operation will be written in an tamper-proof log on the blockchain, a variety of bad behaviors, including denying selling, denying buying and illegal reselling, can be detected and proven easily through querying the log.

We will present our system in detail in the next section, including all operations, possible cases requiring verification and corresponding treatments in both two trading scenarios.

III. System Design

A. Scenario 1: Platform serves as a broker

Figure 1 presents the system design and work flow of a transaction in Scenario 1, which includes the following steps.

(1)-(3) Data preparation. A seller collects data $\mathbb{D}$ for sale and stores it locally. The fingerprint $f_{\mathbb{D},S}$ of $\mathbb{D}$ is generated by the seller using the fingerprint generator and encrypted with the platform's public key Pk_P automatically.

(4)-(8) Put data on the market. The seller encrypts the information of data $\mathbb{D}$ with the platform's public key. The information includes the data fingerprint $f_{\mathbb{D},S}$ and the data description $d_{\mathbb{D}}$. The seller sends the ciphertext $E_{Pk_P}(f_{\mathbb{D},S}, d_{\mathbb{D}})$ to the platform. The fingerprint is required for data transmission verification, while the description is optional. Receiving the ciphertext from the seller, the platform decrypts it and compares the fingerprint with records in the data storage to ensure the seller has the right to sell the data, i.e., currently, the seller is the owner of the data and the data isn't on the market yet. Specifically, the platform maintains a "data-contract" table which records each historical smart contract's address, the fingerprint and description of the contract's corresponding data, and the account of the data's owner. Note that each data for sale is bound to a unique smart contract. The platform compares the fingerprint $f_{\mathbb{D},S}$ with fingerprints in the "data-contract" table. If no existing fingerprint matches $f_{\mathbb{D},S}$, i.e., the similarity between $f_{\mathbb{D},S}$ and any existing fingerprint doesn't exceed a threshold, the platform will generates a new smart contract for $f_{\mathbb{D},S}$ on the blockchain, which records $f_{\mathbb{D},S}$, $d_{\mathbb{D}}$ and $Pk_{\mathbb{D},C}$. $Pk_{\mathbb{D},C}$ is the public key for this contract and also the account address of the contract. If there already exists a matched fingerprint, the platform won't approve this sale request. The price of the data is determined by the pricing module. There are various pricing strategies, including but not limited to seller pricing, negotiated pricing and auction. The seller can choose his/her preferred pricing strategies for data ownership and use right. The determined price will be recorded in the smart contract. The seller can pay the management fee to the platform through calling the charge function in the contract. The management fee is optional and determined by the platform. Now $\mathbb{D}$ is on the market for sale.

(9)-(12) Data purchase. As aforementioned, there are two ways for a buyer to find his/her desired data, whose processes are presented in (9-1) and (9-2) respectively. In the first way, a buyer chooses data by browsing descriptions of all data for sale. In the second way, a buyer needs to provide a request fingerprint f_d, which can be extracted from some sample data d. In this case, the buyer wants to find data similar to the sample data. The buyer sends the request encrypted with the platform's public key, i.e., $E_{Pk_P}(f_d)$, to the platform. The platform decrypts it and searches the "data-contract" table for a list of similar data $\{s_1, s_2, cdots, s_n\}$. The platform sends the buyer the feedback $E_{Pk_B}(Pk_{s_1,C}, Pk_{s_2,C}, \cdots, Pk_{s_n,C})$ encrypted with the buyer's public key, which consists of public keys of the similar data's corresponding contracts. The buyer can call these contracts to get the description and price of each data and decide the data $mathbbT$ to purchase. Note that, in this step, the buyer is not allowed to access any data fingerprint. The buyer calls the payment function in the contract of $mathbbT$ and then the payment will be transferred from the buyer's account to the contract account. Meanwhile

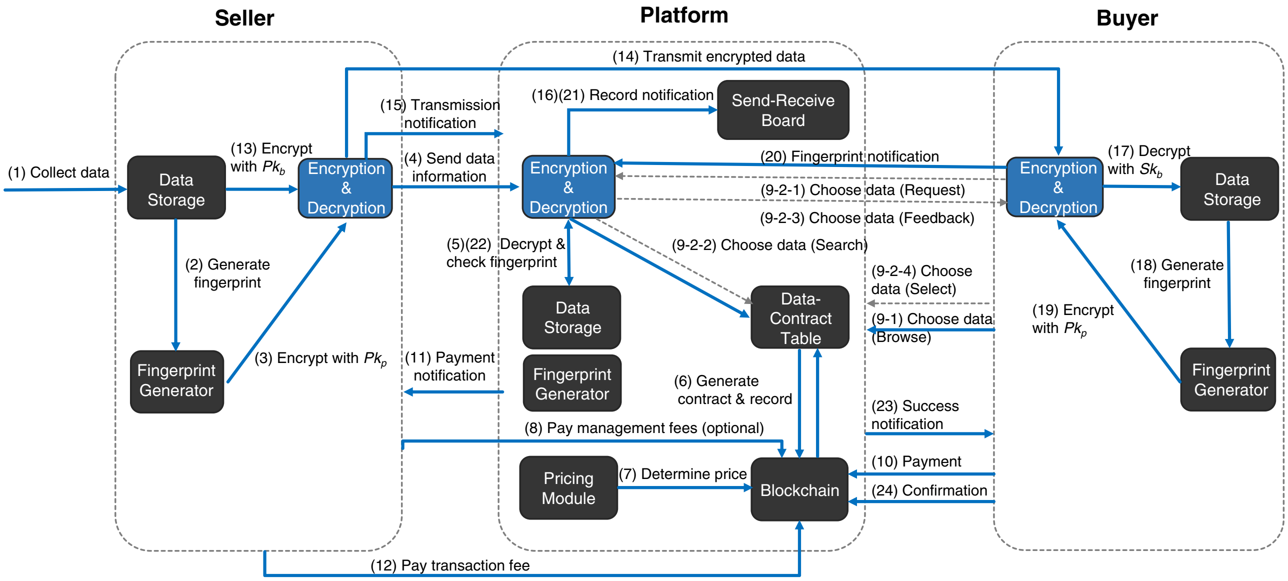

Seller
Platform
Buyer
(14) Transmit encrypted data
(15) Transmission notification
(16)(21) Record notification
Send-Receive Board
(1) Collect data
Data Storage
(13) Encrypt with Pk_b
Encryption & Decryption
(4) Send data information
Encryption & Decryption
(20) Fingerprint notification
(9-2-1) Choose data (Request)
(9-2-3) Choose data (Feedback)
Encryption & Decryption
(17) Decrypt with Sk_b
Data Storage
(2) Generate fingerprint
(5)(22) Decrypt & check fingerprint
(9-2-2) Choose data (Search)
(9-2-4) Choose data (Select)
(18) Generate fingerprint
Data Storage
Data-Contract Table
(9-1) Choose data (Browse)
(19) Encrypt with Pk_p
Fingerprint Generator
(3) Encrypt with Pk_p
(11) Payment notification
Fingerprint Generator
(6) Generate contract & record
Fingerprint Generator
(23) Success notification
(8) Pay management fees (optional)
(10) Payment
Pricing Module
(7) Determine price
Blockchain
(24) Confirmation
(12) Pay transaction fee

Fig. 1. System design and work flow in Scenario 1, where the platform serves as a broker and supervisor.

the seller will be notified about the buyer's payment and he/she will also transfer the transaction fee to the contract account by invoking the contract's charge function.

(13)-(21) Data transmission: The seller encrypts data $\mathbb{D}$ with buyer's public key and transmits $E_{Pk_B}(\mathbb{D})$ to the buyer through the platform. Once the platform completes relaying the data, the seller is required to send an encrypted notification $E_{Pk_P}(Pk_S, Pk_B, f_{\mathbb{D},S}, Pk_{\mathbb{D},C}, t_1)$ to the platform. t_1 is the time stamp when the transmission is completed. The platform decrypts the notification and records it in the send-receive board. Meanwhile, the buyer decrypts the received data with his/her own private key. He/she generates the data fingerprint $f_{\mathbb{D},B}$ using the same fingerprint generator as the data seller and sends encrypted notification $E_{Pk_P}(Pk_S, Pk_B, f_{\mathbb{D},B}, Pk_{\mathbb{D},C}, t_2)$ to the platform. The platform also decrypts and records it in the send-receive board.

(22)-(24) Transaction confirmation. The platform checks whether the two fingerprints $f_{\mathbb{D},S}$ and $f_{\mathbb{D},B}$ from the above two notifications are consistent. If they are the same, the platform will notify the buyer that the transaction is successful and the buyer will call the contract to confirm this transaction. The payments kept in the contract, including the buyer's payment for the data and the seller's payment for the transaction, will be transferred to the seller's account and the platform's account respectively. The ownership or the use right of $\mathbb{D}$ will be updated in the "data-contract" table and the contract accordingly.

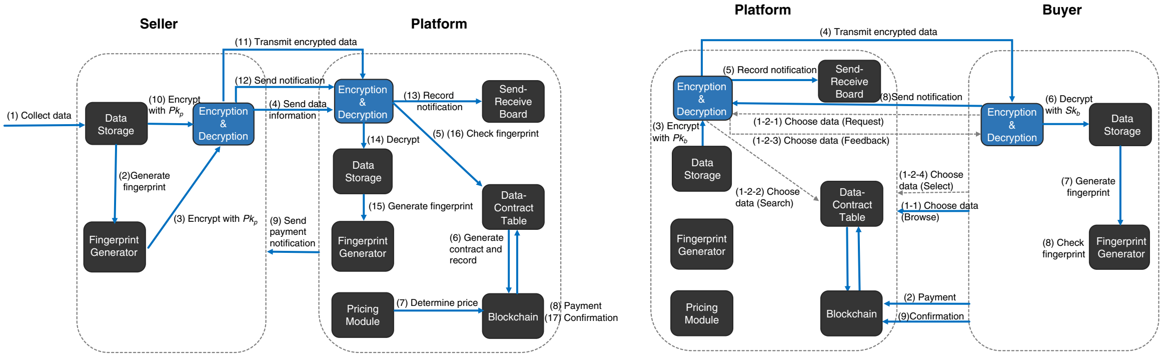
After going through all above steps, the first trade of data $\mathbb{D}$ is completed. Thereafter, the trade of the same data $\mathbb{D}$ will be conducted from step (9) to step (24).

B. *Scenario 2: platform servers as a reseller*

Generally, the framework design of Scenario 2 is similar to the Scenario 1. In Scenario 2, there are two sub-scenarios: (1) the platform works as a buyer to purchase data from a seller as showed in Figure 2(a); (2) the platform works as a seller to sell data to a buyer as showed in Figure 2(b). The main differences with Scenario 1 are (1) in the first sub-scenario, the platform needs to pay for the data and the seller sends both data fingerprint and original data encrypted with the platform's public key to the platform; (2) in the second sub-scenario, for the premise that the platform is honest, the buyer dose not need to send fingerprint of the received data to the platform for consistence checking.

C. *Fingerprint Generator*

In order to detect the potential malicious behaviors, such as transmitting fabricated data and illegally reselling data, we introduce a data fingerprint generator to extract an identifiable vector for each data. The distance between fingerprints can also be used to measure the content similarity between data. Traditional hash method like SHA2 cannot fulfil the requirement for malicious behavior detection, since a seller could make minor modifications to the data to disguise it as a new data for reselling. Therefore, the fingerprint should be robust enough to resist minor data modifications. We concatenate multiple feature vectors extracted from the data to form its fingerprint. Taking image data as an example, we extract the following three typical image features for similarity measurement. (1) Color histogram characterizes the color distribution of an image. (2) Average Hash (aHash) is a perceptual hash algorithm. First, we convert an image to a 32×32 matrix through Discrete Cosine Transform (DCT). The 8×8 upper left corner of the matrix is remained to represent



(a) The platform buys data from a seller.

(b) The platform sells data to a buyer.

Fig. 2. System design and work flow in Scenario 2, where the platform serves as a reseller.

the low-frequency part of the image. The feature vector is calculated based on whether each value in the low-frequency matrix is above or below their average value, resulting in a 64-bit binary feature vector. (3) Structural Similarity Index (SSIM) mainly characterises the brightness, contrast, and structure information of an image. Concatenating these three features yields the fingerprint of an image. When measuring the similarity of two images using their fingerprints, we simply calculate the weighted sum of similarities of three features. Any more advanced data fingerprint can be adopted by our system.

IV. System Analysis and Evaluation

A. Resistance To Malicious Behaviors

We consider five possible malicious behaviors and analyze the ability of our system to resist those behaviors. Some reputation strategy can be adopted to punish malicious users, which is out of the scope of this work.

1) Denying the sale or purchase: A seller or buyer cannot deny his/her sale or purchase behavior, because all the on-chain operations of calling contracts are recorded in the tamper-proof log and all the off-chain operations of data transmission are recorded in the send-receive board maintained by the honest platform.

2) Illegally reselling data: A seller cannot resell the data whose ownership has already been sold to the other user, even the seller makes some minor modifications to the data. This is achieved by comparing the data's fingerprint with fingerprints recorded in all existing contracts (step (5)).

3) Inconsistent data and fingerprint: There are three cases when data and fingerprint are inconsistent: the seller providing fake fingerprint in step (4), the seller transmitting fabricated data in step (14) and the buyer providing fake fingerprint in step (20). All these three types of inconsistency can be catched in the "check fingerprint" step (step(22)). When the inconsistency happens, the platform needs the seller to upload the data encrypted with the platform's public key, i.e., $E_{Pk_P}(\mathbb{D}')$. The platform decrypts the data and generates the fingerprint $f_{\mathbb{D}',P}$. If $f_{\mathbb{D}',P}$ is consistent with $f_{\mathbb{D},P}$, data $\mathbb{D}$ is the correct data and the platform will transmit $E_{Pk_B}(\mathbb{D}')$ to the buyer. The payment for transaction and data will be transferred to the platform and the seller respectively. If $f_{\mathbb{D}',P}$ is inconsistent with $f_{\mathbb{D},P}$, the seller is malicious and the transaction will be terminated with the payment returned to the buyer. A malicious buyer cannot get any benefit since he/she needs to pay for the data as long as the seller is honest. For a malicious seller, providing fabricated data will fail the current transaction and providing fake fingerprint will cause failures of all his/her transactions. Failed transactions will cause economic loss to the seller, which works as a punishment to his/her malicious behavior, since he/she pays management fee to the platform. It is still an open problem to automatically exam the consistency of the data and its description.

4) Seller refuses to transmit data: If the seller doesn't transmit data to the buyer after the buyer has paid, the platform will be notified by periodically comparing payment records in contracts and transmission records in the send-receive board. When it passes the data transmission deadline, the platform will call the contract to return the payment to the buyer and stop the transaction mandatorily.

5) Buyer refuses to send transaction confirmation: For each transaction, there is a payment transfer deadline written in the contract. If a buyer doesn't send a confirmation after he/she receives data, a mandatory payment transfer function will be triggered after the deadline.

B. Privacy

We considered three types of privacy, including identity privacy, trading behavior privacy and data privacy. Identity privacy of users is ensured by the anonymity of the blockchain. Each user only reveals his/her public key instead of the real identity. All communications are conducted in an encrypted manner between the seller/buyer and the platform, and the seller and buyer do not contact each other directly. A third party can learn nothing by eavesdropping the communication. Only the platform can learn the selling behavior, purchasing behavior, payment behavior and transmission behavior. Since the identities of sellers and buyers are unknown, their behavior privacy is preserved. For data privacy, our system adopts the

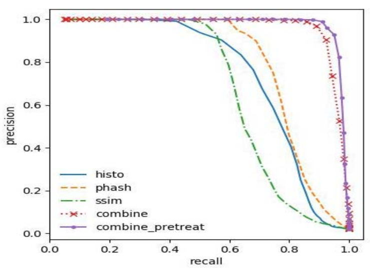


Fig. 3. Precision and recall for fingerprint based pirated image detection.

paid first strategy, thus any party including the platform cannot learn the original data except the buyer who has purchased the data ownership. Moreover, any user cannot even access the data fingerprint before purchasing the data.

C. *Data Fingerprint Evaluation*

In this work, we propose to extract identifiable features from data as its fingerprint. We take image data as an example and introduce a fingerprint design in Section III-C. Here we evaluate the accuracy and robustness of the image fingerprint. We adopt the NUS_WIDE image dataset with 269,648 images [1]. By applying random minor modifications, including cropping, adding noise and changing brightness, we generate 20 pirated images for each image. Using the original image as the query to search its pirated images, Figure 3 present the precision and recall with different fingerprint similarity threshold. Compared with using a single feature (color histogram or pHush or SSIM) as fingerprint, the combination of three features achieves both high precision and recall. When the recall is 95%, the precision is about 98%.

V. Related Work

Many traditional methods like Digital Rights Management (DRM) [6] and watermark [4] have been proposed to protect copyright of data. DRM is a integration project involving technology, law and business at all levels to ensure digital contents will be legally used by authorized users. Watermark technology embeds copyright and authorization information into the data. Forensics technology can also be adopted for copy right protection, which mainly focuses on investigation on data history. [2] implements data management and accountability in the cloud environment and tracks the usage of user data in the cloud. Recently, there are some work addressing privacy issues in data trading. TPDM [5] integrates truthfulness and privacy preservation in data trading, which is structured internally in an Encrypt-then-Sign fashion using somewhat homomorphic encryption and identity based signature. TPDM needs the service provider to collect massive raw data from the data contributors. [7] designed a crowdsourcing based image dataset purchasing framework which preserves data privacy. [3] is most related to our study, which mainly uses the append-only bulletin board to record transactions. They also design a series of protocols that ensure transactions to be fair and detects illegal data efficiently at upload stage.

VI. Conclusion

To meet the urgent demands for privacy-preserving and trustworthy data trading, we propose, design and implement a blockchain and smart contracts based data trading system with data tracking and illegal behavior detecting functions. It enables two trading scenarios with privacy protection against any unauthorized party including the trading platform. An effective fingerprint method is designed to detect the modified image data, thus protect data copyright. Our evaluation on a large image dataset shows the efficacy of our system.

Acknowledgments

This work is supported by the National Key R&D Program of China 2017YFB1003003, NSF China under Grants No. 61822209, 61751211, 61572281, 61520106007, and the Fundamental Research Funds for the Central Universities.

References

[1] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, “Nus-wide: A real-world web image database from national university of singapore,” in *Proc. of ACM Conf. on Image and Video Retrieval (CIVR’09)*, Santorini, Greece., July 8-10, 2009.

[2] M. Felici, T. Koulouris, and S. Pearson, “Accountability for data governance in cloud ecosystems,” in *IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 2. IEEE, 2013, pp. 327–332.

[3] T. Jung, X.-Y. Li, W. Huang, J. Qian, L. Chen, J. Han, J. Hou, and C. Su, “Accounttrade: Accountable protocols for big data trading against dishonest consumers,” in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[4] A. Khan, A. Siddiqa, S. Munib, and S. A. Malik, “A recent survey of reversible watermarking techniques,” *Information sciences*, vol. 279, pp. 251–272, 2014.

[5] C. Niu, Z. Zheng, F. Wu, X. Gao, and G. Chen, “Trading data in good faith: Integrating truthfulness and privacy preservation in data markets,” in *IEEE 33rd International Conference on Data Engineering (ICDE)*. IEEE, 2017, pp. 223–226.

[6] B. Rosenblatt, B. Trippe, S. Mooney *et al.*, “Digital rights management,” *New York*, 2002.

[7] L. Zhang, Y. Li, X. Xiao, X.-Y. Li, J. Wang, A. Zhou, and Q. Li, “Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2735–2743.