TSAR: a fully-distributed Trustless data ShARing platform

Hanqing Wu*, Jiannong Cao*, Shan Jiang*, Ruosong Yang*, Yanni Yang*, Jianfei He†
*The Hong Kong Polytechnic University, Hong Kong, China
†Huawei Technologies Co. Ltd., Shenzhen, China
{cshwu,csjcao,cssjiang,csynyang,csryang}@comp.polyu.edu.hk, jeffrey.he@huawei.com

***Abstract*—Nowadays is the big data era. A large amount of data are generated which can be valuable for business, healthcare, transportation, etc. To promote the dissemination of the valuable data, researchers have been trying to design and develop data sharing platforms. However, the existing platforms fail to address at least one of the three issues: trustworthiness, data heterogeneity, and authenticability. To this end, we propose TSAR, a fully-distributed Trustless data ShARing platform. In detail, we architect TSAR on Blockchain to remove the dependency on reliable third parties, which realizes the trustworthiness. Moreover, we propose a general data schema to represent raw data, which handles the problem of data heterogeneity. Finally, we record the data transaction as well as user-group information on Blockchain to achieve authenticability. To demonstrate the practicability and effectiveness of TSAR, we implement it in a minimal-viable-product fashion and evaluate the performance in terms of throughput and response time.**

***Index Terms*—Blockchain, distributed systems, data sharing, access control**

I. Introduction

Human beings benefit a lot from big data analytics. For example, the companies analyze the behaviors of their customers based on the collected data and produce the products more suitable for the customers [1]; the hospitals take advantage of the gene data and daily data of the individual for more precise disease treatment and prevention [2]; the airports schedule the boarding of thousands of planes more efficiently by fusing data of weather, ground transportation [3], etc.

Concerning big data analytics, usually data from multiple sources are required for one application. Take the taxi as an example. To allocate the taxi drivers more appropriately, various data are needed such as weather data, POI data, traffic data and so on [4] [5] [6]. These data are from various institutions such as the bureau of weather, road transport, and geology, etc . In such case, data sharing is in urgent need to achieve taxi driver allocation.

However, data sharing is performed in a primitive way in most of the big data analytics applications [7]. That is, the companies figure what data is needed and ask other institutions whether they can offer the desired data. Then, the question is why the data owner does not want to publish their data directly. It is because of the following reasons. On the one hand, the institutions who own the data are not aware of the value of data. On the other hand, it is complicated, inefficient, or unsafe for the data owners to publish data. Also, it is not guaranteed that the agency will not leak the data. Indeed, the data owners have an alternative to put the data on their official websites which can be hardly guaranteed that the sites can be found by the demander.

To this end, it is essential and urgent to build a data sharing platform. In research community and market, there are few data sharing platforms [8][9][10]. However, they all suffer from at least one of the following issues. First, they require full trustworthiness from the data owners. Second, the shared data can be heterogeneous, thus requires significant efforts to be managed. Third, the users can have various requirements on the access control policy, for example, to share the data with a specific organization whenever they request it.

In this paper, we propose TSAR, a fully-distributed Trustless data ShARing platform. TSAR addresses the above three issues successfully as follows. TSAR is architected on Blockchain [11], which achieves decentralization. Since there is no third party involved, TSAR is trustless which requires no trust from the data owners. Moreover, we propose a metadata schema for the data owners to publish their data. In this way, only the metadata can be accessed publicly not the raw data stored locally. Finally, TSAR provides a Blockchain-based authentication mechanism, which automates the access control of the shared data. The data owners can specify the access control rules in the shared metadata.

The contributions of this paper are summarized as follows:

- We propose TSAR, a fully-distributed data sharing platform, which addresses three critical issues, namely trustworthiness, data heterogeneity, and lack of automatic access control mechanism, in existing systems.
- A Blockchain-based authentication mechanism is provided in TSAR. It allows the data owners to specify the access control rules in the shared metadata, which enhances the user-friendliness of TSAR.
- We implemented TSAR in a minimal-viable-product fashion. The implementation demonstrates its practicability. We further evaluate the performance of TSAR concerning throughput and response time.

This paper is organized as follows. Section II introduces related works. Section III demonstrates the system and module design, and the architecture of TSAR and three main functions. Section IV shows the system implementation and evaluation. Finally, section V concludes the paper.

978-1-5386-4705-9/18/$31.00 ©2018 IEEE
DOI 10.1109/SMARTCOMP.2018.00028

II. Related Works

A. Data Sharing Platforms and Tools

The need of providing easy-to-use tools for sharing big data has resulted in a number of platforms and tools. The large scale organized communities, like High Energy Physics [12], have already developed their own data management systems which is out of scope for this research. We also consider general file hosting services such as Google Drive [13] out of scope. Four closely related systems that have emerged in the past few years are discussed in detail below: Zenodo [14], CKAN [15], Figshare [16] and IPFS [17].

Zenodo [14] is a research data repository as created and hosted by OpenAIRE and CERN to provide a place for researchers to deposit datasets. Zenodo code is open source, and is built on the foundation of the Invenio digital library. It is a general-purpose open access repository and it supports all type of files. Data can be published under different types of licences and it can be flexible controlled. Zenodo assigns a unique DOI to the data and provides APIs for uploading data and harvesting metadata. The Comprehensive Knowledge Archive Network (CKAN) [15] is a web-based open source management system for the storage and distribution of open data. It has developed into a powerful data catalogue system which mainly used by public institutions seeking to share their data with the general public. CKAN supports permanent URIs for citation, e.g. DOIs, by extension packages. It supports RESTful JSON API with required tools for querying and accessing data. Figshare [16] is an online digital repository where researchers can preserve and share their research outputs, including figures, datasets, images, and videos. In adherence to the principle of open data, it is free to upload content and free to access. Users can upload files in any format, and items are attributed a DOI. Figshare has different functionalities dependant on being authenticated user or not. InterPlanetary File System (IPFS) [17] is a content-addressable, p2p hypermedia distribution protocol. Nodes in the IPFS network form a distributed file system. It is a p2p distributed file system that seeks to connect all computing devices with the same system of files. IPFS could be seen as a single BitTorrent swarm, exchanging objects within one Git repository. In other words, IPFS provides a high throughput content-addressed block storage model, with content-addressed hyperlinks. IPFS combines a distributed hashtable, an incentivized block exchange, and a self-certifying namespace.

B. Traditional Data Sharing and Transaction Models

Since there is a need for data sharing and data transactions, traditional models exist and are providing services for data sharing and trading. However, the traditional models can not protect the security and interests of both data supplier and data demander. Here we classify the traditional data sharing and transaction models into twofold: *Data Hosting Center* (DHC) and *Data Aggregation Center* (DAC).

In DHC model, each agency will host, upload and publish its own data to the central database which is controlled and maintained by the DHC. The DHC is responsible for the data exchanging and trading with external agency. After the data is hosted, the data is completely owned by the DHC. All the follow-up applications of the data are independent from the agency. This model is widely used in the current data sharing platform due to its character of convenience, easily operating and low cost.

In DAC model, the Center links data services through the API interface among agencies. Data agencies do not need to report, upload to the DAC in advance. The data is still owned and managed by the data agencies. When an agency needs to search the data, it will use the real-time interaction with the DAC to send the data request. The DAC will relay and broadcast this request to other agencies. Once other agencies with the target data response to this request and return the data, the DAC will collect all the data and send back to the data demander. However, it is not hard to find that the DAC has the ability and the opportunity to retain the data. The DAC can accumulate the data during sharing, and it will gradually become a DHC.

III. System and Module Design

In this section, we first describe the architecture of TSAR in subsection. III-A. Then, we introduce the three modules, i.e., data publishing, data retrieval, and data sharing in detail from subsection. III-B to subsection. III-D.

A. System Overview

The system architecture is illustrated in Fig. 1. For each user who is using TSAR, he/she uses five local components to perform three network functions. The five components are the raw data, the metadata, the metadata chain, the sharingdata chain, and the TSAR interface, while the three network functions are data publishing, data retrieval, and data sharing.

First, if a user owns some raw data to be shared, the user need to notify the other users in TSAR network that there is a piece of newly published data. The process of data publishing involves with the components of raw data, metadata, and metadata chain. Specifically, the raw data stored by each user locally is transferred into metadata, and the metadata is published on the metadata chain, which is accessible by all the users on TSAR network. The metadata chain is a Blockchain, which stores metadatas as transactions. Second, the function of data retrieval is required when a user wants to search data with some keywords. Finally, when a user wants to get a certain data, data sharing is needed.

B. Data Publishing

In traditional data sharing platform, the users have to upload their raw data or metadata to achieve publishing data. Under this schema, a centralized server collects the uploaded data and display them. This method heavily relies on a trustworthy service provider. By saying trustworthy, it means that the service provider is not supposed to make any modification on the uploaded data. To remove such a centralized service

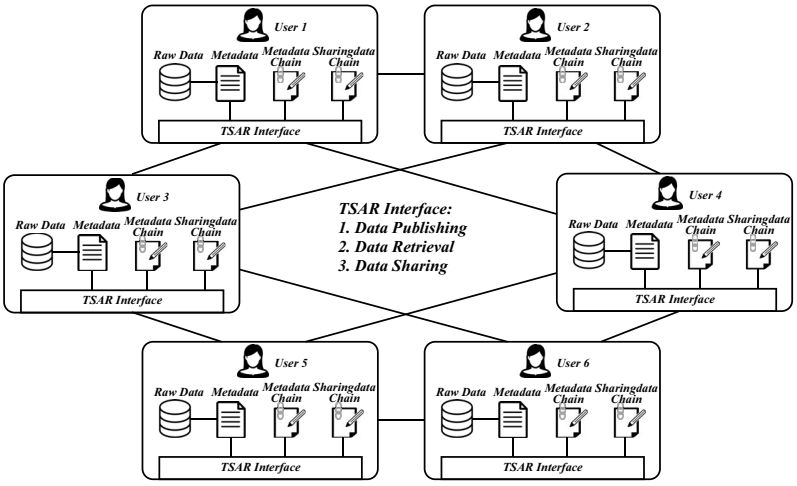

Fig. 1. System Architecture of TSAR

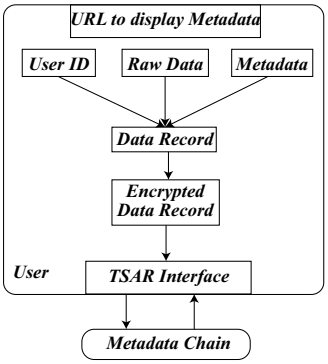

Fig. 2. Flowchart of Data Publishing

provider, we propose to use metadata chain for publishing data in this subsection.

Fig. 2 shows the proposed procedure for a user to publish data which is divided into three steps: 1) packing raw data into data record with signature; 2) broadcasting and verifying the data record; 3) synchronize of metadata chain.

The input of the data publishing procedure is the user's raw data. The raw data can be gigabytes and even terabytes. If the raw data is directly published, it is nearly impossible to guarantee its copyright. Also, it is a huge burden to the network. To this end, we define a new data type, metadata, for purpose of describing and publishing the raw data.

The metadata is a description of the raw data which contains the data schema, a set of keywords, a small amount of sample data, the acquisition time and the data size. The format of the metadata is defined so to fully describe the raw data and for purpose of high-performance retrieval. The size of a piece of metadata is around several hundreds of kilobytes. Compared to the huge-size raw data, it significantly reduces the burden of the network.

After transforming the raw data to metadata, the metadata is published via HTTP service. In this way, everyone in the network can view the metadata via the corresponding URL. Also, the published metadata cannot be modified by others. However, there are still two issues to handle. The first issue is how to make other users in the network aware of the newly published data. The second issue is how to guarantee that the metadata is not modified by its owner after publishing. To address these two issues, we propose a metadatachain mechanism for decentralized recording of data publishing.

After a user generates the URL to display the metadata, a data record which composes of user ID, checksum of raw data, checksum of metadata, and the the URL to display the metadata is generated. The data record is encrypted using the user's private key and broadcast to the whole network using the TSAR interface.

If a user receives a data record, the data record will be verified as follows:

- identify the user using the signature in the record;
- acquire the public key of the data publisher;
- use the public key to decrypt the data record;
- whether the data format is as defined;
- whether the signature is the same with publisher;
- whether the URL contained in the data record is accessible;
- whether the metadata in the URL is as defined;
- whether the metadata checksum is the same with the one in the data record;

The conditions are checked one by one. If there is any unsatisfied condition, the data record will be aborted. Everyone in the network will check every data record to make the data records consistent. If a data record is verified by a user, it will be put into the user's local metadata pool. Note that it does not mean that a data record is published if it is in the metadata pool. At a fixed frequency, the data records in the metadata pool will be packed into MetadataChain. If a data record is packed into the MetadataChain, it is published. Each node in

the network will synchronize the MetadataChain.

C. Data Retrieval

As for data retrieval, there exists a central server to respond to users's query in traditional data sharing platform. And for P2P network, each user sends own query to the neighbor peer and the neighbor peers respond the query and send the query to its neighbor peer. It is obviously, the central one needs a central sever and how to balance load is a difficult problem, and the later one, broadcasting the query to all the peers is a time-consuming operation, and users may get response with large delay.

In our system, we design a totally decentralized one, and do not need server and broadcast the query. As mentioned in former section, each metadata would publish on a Metadata Chain, and users' own client of our system respond to its own query according to the Metadata Chain. The procedure is as following:

1) Client Metadata Chain synchronization
2) Word extension and similarity
3) Data retrieval and show results

For the users who have attended the system will have performed Metadata Chain Synchronization as mentioned in former section. However, for new user, or the user who only want to search the data they need and have not published any data, in their client, they have not synchronize the Metadata Chain. So in data retrieval, client Metadata Chain block Synchronization is the first step. The procedure of Metadata Chain Synchronization is same to the former section. However synchronize the total chain may be a time-consuming cost, we may consider how to avoid download the whole chain in the future.

1) Word Extension and Similarity: The aim of data retrieval is to respond the user's query, and in the Metadata Chain, especially in each metadata recorded in each block, there are some key words to describe the semantic information of the data, the retrieval process actually is to backtrack each metadata and match the query to the key words.

In this section, we will introduce how to extend query key words and how to match the query key words and the metadata key words.

In practical system, like search engine, users query is always short and contain very little information. So only use the query words could not get available results, and common method to solve this problem is to extend the query words with extra knowledge. There are so many human-designed knowledge base such as WordNet[18] which contains synonyms, antonyms, word definitions and so on. In our system, for each query word w_i, we extract the synonyms 1 of the word through wordNet, and use all these words as the query words. And in order to avoid different word form, such as "traffic" and "traffics", we utilize the NLTK [19] interface to get the stemming form of each key words. 2 is the final query keywords set and we use Q to search for the related data.

$$S = \{w_k | w_k \in Syn(w_i)\} \tag{1}$$

$$Q = \{Stem(w) | w \in S\} \tag{2}$$

$$T = \{Stem(w) | w \in R = Syn(w_{tag})\} \tag{3}$$

$$Jacarrd(Q, T) = \frac{Q \cap T}{Q \cup T} \tag{4}$$

Then it is important to match the query key words with each metadata key words. In order to publish more data in each block, each metadata have very little tags as key words, and directly compare these tags with query key words it is also difficult to find the semantical related one. We extend the tags of the metadata using the similar method of that with query words. We could get the final tags T 3. And we use the Jacarrd Similarity 4 to calculate the distance between query key words and metadata tags.

2) Data Retrieval and Show Results: Data retrieval is similar to Search Engine, and we want to return a list of data which may semantically similar to user's query. In this subsection, we introduce the whole procedure of data retrieval and how to rank the data. And in order to speed the retrieval process and according to the locality principle, we build a cache for each user to record recent search results. The key idea of the data retrieval is to traverse the cache data. After getting preliminary results of the semantically similar data, we want to rerank the results so that the data with smaller rank number will be more needed. Following a simple idea, the data published more early, the data will be less important. So we add a weight decay to the data similarity according to the published date, shown in Formula 5. F_S means the final similarity, S means the similarity calculated the former algorithm, D means the publish date of the current data, D_R means the latest publish date during all the list DL.

$$F_S = S * e^{(D - D_r)} \tag{5}$$

D. Data Sharing

The goal of distributed big data sharing platform is to ensure the authenticity of the data, reliability of the sharing mechanism and legality of user behavior. The module of data sharing is composed of two functions: (1) data usage under flexible and safe control; (2) reliable data sharing mechanism. With the above two functions, users can not only freely set different permission modes for sharing data, but protect the data ownership. Meanwhile, for the data requester, they can also guarantee the authenticity for the data they want.

1) Data Usage Mechanism: The data usage module of the system is mainly designed to ensure the controllability and reliability for the data sharing. For the controllability of data sharing, the user can set different permissions to the data requester to obtain and use the data through the ID of the requester. The way of data usage is divided into following two models according to the identity of the data requester.

2) Unlimited Data Usage: Data requesters with unlimited data usage permission search the intended data through MetadataChain, retrieve the data, and send a data-sharing request to the owner. Once the request has been approved by the owner, the intended data can be sent to the requester. Since

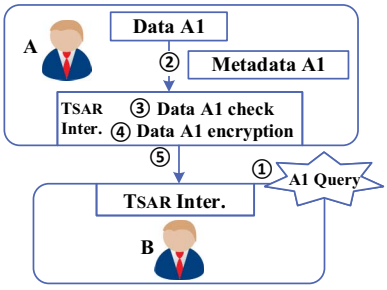

Fig. 3. Mechanism to guarantee data authenticity

TSAR is distributed without centralized party involved, there are times when the data sent by the data owner does not match the original data information posted by the data owner in the Metadata Chain. In order to guarantee the authenticity of the data obtained by the requester, the data authenticity verification function is involved in the TSAR interface, as shown in Fig. 3. After B retrieves Data A1 from the Metadata Chain, he broadcasts his request for Data A1 to the entire network. Data owner A sends the data to B via the TSAR interface after receiving B's request. Before the Data A1 is sent to B, the TSAR interface first checks the data. If the metadata parsed from the data is the same as the metadata A posted before, TSAR will encrypt Data A1 and send it to B. B will get the encrypted data of Data A1. If the metadata parsed by TSAR does not match Metadata A1, the system will reject the data sharing action and ask A to send the original data of Data A1.

3) Limited Data Usage: Users with limited data usage permission cannot directly access the data owner's original data, but can obtain the desired processing result by sending an operation instruction to the data owner. In this case, the system not only needs to ensure that the data owner's data is not maliciously acquired or damaged, but also ensures that the data requester can obtain the data processing result more conveniently and accurately.

4) Data Sharing Mechanism - Sharingdata Chain: Data ownership protection is one of the key functions in distributed big data sharing platform. There are cases when the data requester tampers and republishes the data obtained from other users or even derives profits from it, which seriously infringes the ownership of the data owner. Therefore, through the mechanism of Sharingdata Chain, the system stores the record of data sharing information in the Sharingdata Chain. The data sharing records on the Sharingdata can be regarded as the evidence for the data ownership, and it can also be used for tracking the behavior of data sharing.

The design of Sharingdata Chain follows the concepts of blockchain. Sharingdata Chain makes some innovations on the basis of blockchain technology for the scenario of big data sharing. The main role of Sharingdata Chain is to store the data recording record as a proof for the data ownership. A data sharing record in the Sharingdata Chain contains the following information:

1) Data owner, data requester and their signatures
2) metadata pointer, verification code and URL of the shared data
3) Sharing time and the permission mode
4) Other additional terms

For users with permission of unlimited data usage, the workflow of the Sharingdata Chain is shown in Fig. 4). Data requester B publishes the request and data sharing contract for requesting Data A1 to the entire network. After receiving the request, user A checks and sends the encrypted Data A1 to user B. During this period, A packs the decryption key (Decryption) of the encrypted Data A1 data together with the signed data sharing contract into Sharingdata Chain. After the Sharingdata block containing the data sharing record is authenticated, B can obtain the decryption key and then recover the encrypted Data A1.

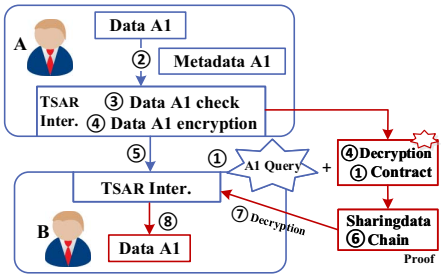

Fig. 4. Workflow of Sharingdata Chain for unlimited data usage

For users with permission of limited data usage (as shown in Fig. 5), data requester B publishes the request and contract for using Data A1 to the whole network and then sends the code and its results on the data sample. Afterwards, user A processes the Data A1 with the code and sends the encrypted processing result to B through TSAR interface. At the same time, user A packs the decryption key (Decryption) together with the signed data sharing contract as a data usage record into the Sharingdata Chain. As long as the block with this record is authenticated, user B will get the decryption key to retrive the encrypted data result.

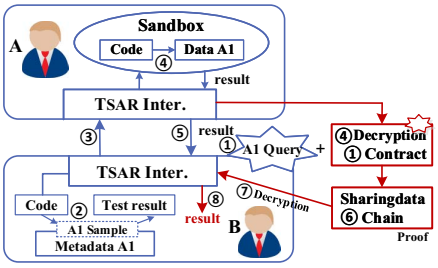

Fig. 5. Workflow of Sharingdata Chain for limited data usage

IV. System Implementation & Evaluation

The key challenge in implementation is the two Blockchains, i.e., metadata chain and sharingdata chain. In TSAR, we implement Blockchain under the framework of

gRPC [20]. A transaction in TSAR is defined to consist of six fields, namely timestamp, source, destination, hash value, type, and body. The timestamp and hash value are the approximate submitting time and the SHA256 [21] hash value of the transaction.

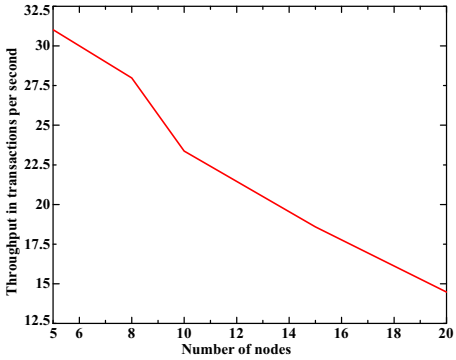

Fig. 6. Throughput v.s. Number of Nodes

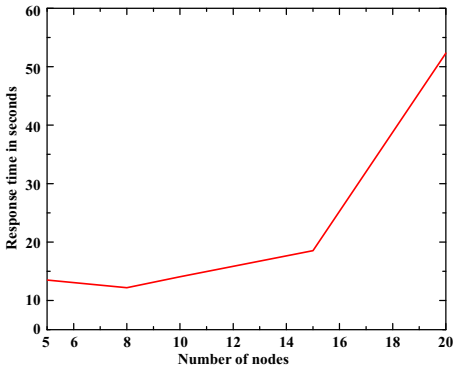

Fig. 7. Response Time v.s. Number of Nodes

We evaluate the performance of the implemented Blockchain by investigating how the number of nodes affects the throughput and response time. We conduct experiments on 5, 8, 10, 15, and 20 nodes respectively. In each set of experiment, each node serves as both client and server. That is, each node generate transactions and pack transactions at the same time. The transaction generation rate for each node is 5 transactions per second. The evaluation result is shown in Fig. 6 and Fig. 7. The evaluation results indicate that as the number of nodes increases, the system throughput decreases and the response time increases. This is true since the system becomes more robust if there are more replica of the data in the network. However, it requires more network resources, which results in degradation of the system performance.

V. Conclusion

In this paper, we propose TSAR, a fully-distributed Trustless data ShARing platform. There are three key innovation points in the design of TSAR. First, we architect TSAR on Blockchain, which removes the need of dependable third parties. Second, we propose to share metadata, which is a description of the data, rather than raw data, which decreases the demand of network resources and copes with the issue of data heterogeneity. Third, we record the data transaction on Blockchain, which achieves non-repudiability. We implement TSAR in a minimal-viable-product fashion and evaluate the system performance concerning throughput and response time. The experimental results indicate the practicability and effectiveness of TSAR.

Acknowledgments

This work is supported by Huawei Technologies Co. Ltd. with project code P15-0540 and RGC CRF with project number CityU C1008-16G.

References

[1] S. Erevelles, N. Fukawa, and L. Swayne, "Big data consumer analytics and the transformation of marketing," *Journal of Business Research*, vol. 69, no. 2, pp. 897–904, 2016.
[2] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health information science and systems*, vol. 2, no. 1, p. 3, 2014.
[3] M. Batty, "Big data, smart cities and city planning," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 274–279, 2013.
[4] W. Li, J. Cao, J. Guan, M. L. Yiu, and S. Zhou, "Efficient retrieval of bounded-cost informative routes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2182–2196, 2017.
[5] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing rfid tags in a time-efficient way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, 2015.
[6] S. Jiang, J. Cao, Y. Liu, J. Chen, and X. Liu, "Programming large-scale multi-robot system with timing constraints," in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–9.
[7] D. Lazer, R. Kennedy, G. King, and A. Vespignani, "The parable of google flu: traps in big data analysis," *Science*, vol. 343, no. 6176, pp. 1203–1205, 2014.
[8] C.-K. Chu, S. S. Chow, W.-G. Tzeng, J. Zhou, and R. H. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE transactions on parallel and distributed systems*, vol. 25, no. 2, pp. 468–477, 2014.
[9] B. Cui, Z. Liu, and L. Wang, "Key-aggregate searchable encryption (kase) for group data sharing via cloud storage," *IEEE Transactions on computers*, vol. 65, no. 8, pp. 2374–2385, 2016.
[10] R. A. Poldrack and K. J. Gorgolewski, "Making big data open: data sharing in neuroimaging," *Nature neuroscience*, vol. 17, no. 11, p. 1510, 2014.
[11] S. Jiang, J. Cao, H. Wu, Y. Yang, M. Ma, and J. He, "Blochie: a blockchain-based platform for healthcare information exchange," in *Smart Computing (SMARTCOMP), 2018 IEEE International Conference on (to appear)*. IEEE, 2018, pp. 1–8.
[12] "High energy physics (hep)," https://science.energy.gov/hep/, accessed: 2018-01-25.
[13] "Google drive," https://www.google.com/drive/, accessed: 2018-01-25.
[14] "Zenodo: a research data repository," https://zenodo.org, note = Accessed: 2018-01-25,.
[15] "Ckan: Comprehensive knowledge archive network, the open-source data portal platform," http://ckan.org, note = Accessed: 2018-01-25,.
[16] "Figshare - credit for all your research," https://figshare.com/, accessed: 2018-01-25.
[17] J. Benet, "Ipfs-content addressed, versioned, p2p file system," *arXiv preprint arXiv:1407.3561*, 2014.
[18] "Wordnet," https://wordnet.princeton.edu/, accessed: 2018-01-25.
[19] "Natural language toolkit," http://www.nltk.org/, accessed: 2018-01-25.
[20] "grpc: A high performance, open-source universal rpc framework." https://grpc.io/, accessed: 2018-01-25.
[21] "Device for and method of one-way cryptographic hashing," https://www.google.com/patents/US6829355, accessed: 2018-01-25.