International Journal of Engineering Sciences &Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164

IJESRT

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

ACCOUNTABILITY AGAINST DISHONEST BIG DATA BUYERS AND SELLERS

M.S.Yasmine Fathima[*1] **& R.Sahila Devi**[2]
[*1&2]Rohini College of Engineering and Technology, India

DOI: 10.5281/zenodo.2595870

ABSTRACT

In this article, we proposed a set of accountable protocols denoted as AccounTrade for big data trading among dishonest consumers. For attaining secure big data trading environment. Throughout the trading (i.e., buying and selling of datasets) bookkeeping and accountability are achieved against consumers. Examines the consumer's responsibilities in the data trading, and then designed AccountTrade to achieve accountability against dishonest consumers that are likely to deviate from their responsibility. Specially uniqueness index is defined and proposed it is a new measure of data uniqueness. Parallelization has been defined against overhead comes from the data file conversion to membership vector.

KEYWORDS: AccontTrade, Uniqueness index, Accountability, Datatrading.

1. INTRODUCTION

A Big data is a term that describes the large volume of data – both structured and unstructured – that inundates a business on a day-to-day basis. But it's not the amount of data that's important. It's what organizations do with the data that matters. Big data can be analyzed for insights that lead to better decisions and strategic business moves.With a number of new technologies integrated into our daily lives, such as mobile and social networking applications, and Internet of Thing (IoT)-based smart-world systems (smart grid, smart transportation, smart city, and others), massive amounts of data will be collected. The different kinds of sensors and smart devices generate large datasets continuously from all aspects and domains. Thus, unprecedented, comprehensive, and complex data, namely big data, becomes more valuable. Furthermore, with the advancement of data analytics provided by machine learning and data mining techniques, and the computing capabilities supported by cloud and edge computing infrastructures, the potential values of the generated big data become more impressive. Thus, big data is the impetus of the next waves of productivity growth. Nonetheless, there are a number of significant challenges, including data collection, storage, analysis, sharing, updating, and others. To maximize the utility of the data collected, one viable solution is to design an effective big data trading market that allows data owners and consumers (i.e., buyers) to carry out data trading effectively and securely. The importance of big data doesn't revolve around how much data you have, but what you do with it. You can take data from any source and analyze it to find answers that enable 1) cost reductions, 2) time reductions, 3) new product development and optimized offerings, and 4) smart decision making. When you combine big data with high-powered analytics.

In finance, market data is price and trade-related data for a financial instrument reported by a trading venue such as a stock exchange. Market data allows trader **s** and investors to know the latest price and see historical trends for instruments such as equities, fixed-income products, derivatives and currencies. On a daily basis, consumers engage in a variety of online and offline activities that reveal personal information about them. Some typical activities include using a mobile device, shopping for a home or car, subscribing to a magazine, making a purchase at a store or through a catalog, browsing the Internet, responding to a survey in order to get a coupon, using social media, subscribing to online news sites, or entering a sweepstakes. As consumers engage in these daily activities, the entities they interact with collect information about them and, in many instances, provide or sell that information to data brokers. Trading of digital datasets has become a trend as the trading of digital datasets became a promising business in the big data era. Although profits lie in big data, organizations

POSSESSING large-scale datasets (companies or research institutes) do not participate in the data trading due to serious concerns in user-generated data. One of the major concerns is that we do not have accountability in the digital data trading. The concerns are particularly huge due to the non-physical nature of the digital dataset – replication and delivery are almost costless when compared to physical commodities. Anxieties arise at the broker side: data owners worry that brokers may illegally disclose or resell the datasets they outsourced to the brokers Anxieties arise at the consumer side as well: dishonest consumers may illegally resell the purchased datasets. Addressing the first issue is possible because that was one of FTC's main concerns and FTC has managed to detect and punish dishonest data brokers already achieving accountability in small-size systems has shown to be possible. Therefore, it is possible to monitor the broker's side. The second issue is hard to address. It is reasonable to monitor the broker's side but it is hardly acceptable to monitor the consumers. Firstly, it is not lawful to lively monitor individual consumers' behaviour because of the privacy implications. Secondly, the history of Internet suggests that any service requires installation of a heavy monitoring system cannot survive because it leads to bad user experiences.

Several challenges exist in designing Account Trade. Firstly, the boundary for whether sale is legitimate or not is hard to define. There are two reasons for that: (1) dishonest sellers may bring perturbation into others' datasets before they try to resell them; defining how much overlap will make a dataset copy of another one is challenging; (2) data brokers buy and sell a huge number of large-scale datasets, but illegal resale monitoring involves scanning the entire dataset that they posses.

This article propose a set of accountable protocols know as AccountTrade for big data trading hosted by brokers. AcountTrade enables broker to maintain accountability against dishonest consumers throughout the trading by detecting the misbehaviour. Misbehaviour defined in this article are tax evasion, denial of purchase and resell of others dataset. Also proposed to detect blatant copy in the dataset uploaded by the owners, by detecting whether the uploaded one is derived from already existing one.

We have the following contributions:

- We define formal models of accountability (symbolic and computational ones) for big data trading, and we design accountable protocols Upload, Examine, and Download that are provably accountable.
- We efficiently detect illegal resale by defining and proposing the uniqueness index that is consistent with state-of-the-art data similarity comparison for different types. Notably, no such mechanism is available for table-type datasets, and existing mechanisms for JSON-like datasets are not scalable. Therefore, we present novel mechanisms to efficiently measure the similarity for those types.
- AccountTrade is highly scalable with both number and volume of datasets. The extra overhead introduced at the users' side remains constant regardless of the data brokers' scale, and the extra overhead at the brokers' side grows linearly with the number of datasets only. The volumes of existing datasets do not contribute to the extra overhead.

2. DEFINITION AND MODELS

Data Trading with Brokers

There are three entities: brokers, sellers, and buyers. Each entity has its own trading-related responsibilities.

Broker: Brokers provide shopping services in general (product listing, description, payment, delivery, etc.). Besides, they are in charge of book-keeping for accounting purposes (i.e., recording trading transactions), and they also need to define what type of transaction is considered as reselling and should be prohibited.

Seller: Sellers are required to sell only the datasets that are collected/generated by themselves, and they should not resell others' datasets by slightly perturbing them. Also, they have to correctly file the tax report regarding the dataset transaction. They should not interrupt brokers' book-keeping.

Buyer: Buyers should not disturb brokers' book-keeping. Some areas are important but orthogonal to ours. Description of dataset's quality/utility for buyers is complementary. Furthermore, we let sellers set the prices, but more sophisticated pricing mechanism may be considered. The accountability we study for data trading is independent from them.

Adversary Model & Channel Assumption
Malicious users: Users may try to deviate from the responsibilities described above. Namely, they may e.g., disrupt the brokers' data trading service, deny cleared transactions (i.e., paid and sold) and resell previously purchased datasets. A user is defined as a dishonest user if he avoided any of the trading related responsibilities, and such behaviour (either selling or buying) is denoted as misbehaviour. Note that, when illegally selling previously purchased datasets, attackers may try to perturb the dataset to bypass copy detection mechanisms. Trusted brokers: We assume the brokers can be trusted, e.g., the role is played by the organizations that are strictly supervised with great transparency or commercial companies with high reputation. Similar assumptions can be found and the assumption that the brokers will be strictly supervised is also consistent with the FTC's recent action. Channel assumption: We assume both buyers and sellers interact with the broker via secure communication channels. The communication is encrypted and decrypted with pre distributed keys to guarantee that the dataset is not open to the public. This also implies authentication is in place since the broker needs to use the correct entity's key for communication.

Accountability Model
The modelling in is inherited to define a formal accountability model for Account Trade. Our accountable protocols are characterized by two properties Fairness: honest entities are never blamed. Goal-centered completeness: if accountability is not preserved due to malicious entities' misbehaviour, at least one of them is blamed. General completeness, which states that all misbehaving entities must be blamed, is impossible to satisfy because "some misbehaviour cannot be observed by any honest party".Account Trade also requires individual accountability, which states that it must be able to correctly blame one or more parties unambiguously, rather than to blame a group without knowing the exact misbehaving person. We define two formal models of accountability with different purposes. Symbolic individual accountability is defined in a setting where all building blocks are abstracted as ideal black boxes. The symbolic model is amenable to automatic security verification protocols, e.g., who verify whether security flaws exist. Then, computational individual accountability without the abstraction is defined to give a quantitative analysis of individual accountability guarantee.

Symbolic Individual Accountability: A verdict is a Boolean formula which includes propositions having the form dis(e), where dis(e) is a statement "the entity e misbehaved". If the broker states = dis(A) ^dis(B), it means the broker blames A and B, and the blame is fair if A and B indeed misbehaved. A run r is an actual run of a protocol. We use the expression r) to denote that evaluates to true in the run r. Then, if a run r contains misbehavior(s) and describes the misbehaved entities in r, we call ϕ= (r =>ψ) an accountability constraint of r because the broker must state ψ after observing the run r. We use ϕ to denote the set of all accountability constraints of all possible runs in a given protocol P , denoted as P 's accountability property. We say an entity J ensures after observing a run r if either no misbehaviour occurred in r or J states ψ and (r=>ψ) € ϕ.

Definition 1 (Symbolic). Let P be a protocol, J be its entity, and be P 's accountability property. We say P is individually accountable w.r.t. J if

Fairness: verdicts stated by J all evaluate to true, Goal-centred completeness: for every run r of P , J ensures after observing it, and Individual accountability: the only logical operators in J's verdicts are '^'.

Computational Individual Accountability: The computational version is similar to the symbolic one except that we consider the leveraged building blocks may be imperfect. For example, there are always negligible chances for the attacker to break (almost all) cryptographic tools (e.g., by a random guess or with negligible advantage), and the leveraged predictive models can hardly be perfect regarding the precision and recall. By reusing the notations in the symbolic model, we present the following definition.

Definition 2 (Computational). Let P be a protocol, J be its entity, and ϕ be P 's accountability property. We say P is individually accountable w.r.t. J if Fairness: for any verdict ψ stated by J, Pr[ψ= F] is bounded byή.

Goal-centered completeness: for any run r of P , Pr[:(J ensures)] is bounded by x and

Individual accountability: the only logical operators in J's verdicts are '^'.

The expression [entity : inputg] → P [fentity : outputg] is used to define the input and output from different entities
in the protocol P , and ⊥ indicates a null argument. AccountTrade is composed of **Upload, Examine, and**
Download

Upload: This protocol is executed between a seller who wishes to sell her dataset d and the broker. The seller generates a post postt at time t which is posted at the public bulletin board so that the broker can book-keep the transaction and achieve individual accountability.

[Seller : d; Broker :⊥] !Upload [Seller :⊥; Broker : d; postt]

Examine: The broker examines whether the dataset is derived from existing ones with this protocol. He generates a set of MinHash values mh (d) for the dataset d, and they are used to calculate the uniqueness index of d, UD(d), over the entire database D containing all already-uploaded datasets.

[Broker : d] !Examine [Broker : mhπ (d)π ; UD(d)]

Download: This protocol is executed between a buyer and the broker. The buyer generates and posts postt at the bulletin board similar to Upload protocol.

[Buyer :⊥; Broker : d] !Upload [Buyer : d; Broker : postt]

3. SPECIFICATION AND ACCOUNTTRADE

A. Building Blocks

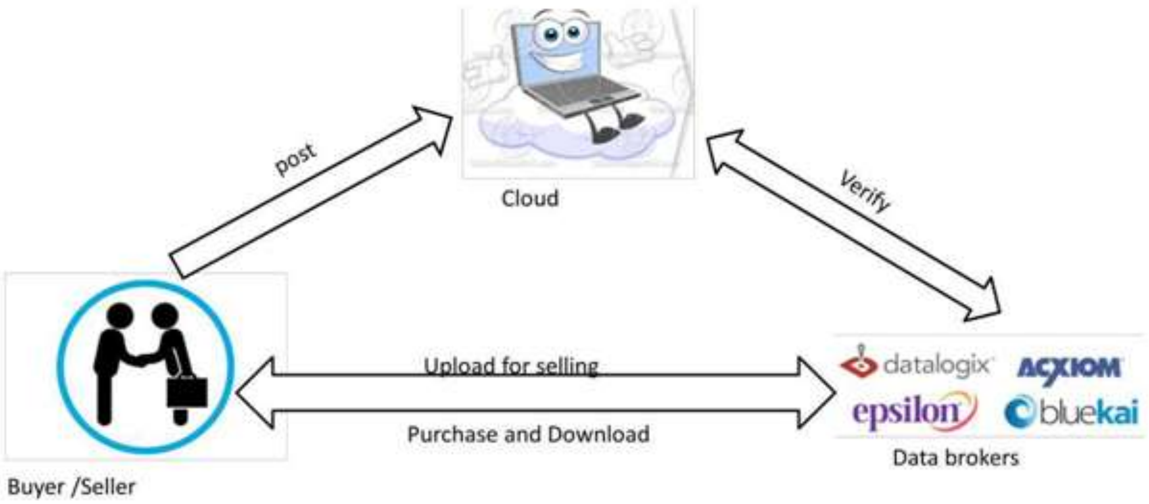
Cryptographic hash: Suppose ∑ is a set of characters. We employ a cryptographic hash function H {0,1}*→∑ where k is a pre-defined system-wide parameter. The hash function

maps any bitstring to a string of length k. **Digital signature**: A secure digital signature scheme is lever-aged to let an entity E sign on a message m € ∑ Produced signature is denoted by sigE(m), and it is used to verify the integrity of m. We also let every signature secret key be bound to a specific user so that the signature can be used to prove E's ownership for accountability purpose. For the simplicity, we omit the signature verification in the protocol specifications.

Append-only bulletin board: A bulletin board with 'append' and 'read' privileges only has been employed as the source of trust in systems requiring accountability or verifiability , It is a public broadcast channel with memory where any party can post messages by appending them to her own area, and she can see anyone's posts as well. A posted message is denoted as a 'post' hereafter. With the building blocks, we present the architecture of AccountTrade as in Fig. 1. Buyers/sellers post posts at the bulletin board whenever they buy/sell datasets, and brokers verify the corresponding records exist before accepting/releasing datasets. After accepting a dataset, the brokers examine it before finally listing it for selling.

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

Figure:



Architecture of AccountTrade

B. Upload for Sale

When a seller A wants to upload a dataset to sell it, she follows the Upload protocol and posts her declaration postt at the bulletin board at time t. Then, she sends the upload request along with H(d) to the broker. The broker finds the corresponding post from the bulletin board and blames A if none is found, because it is evident that she has tried to avoid being book-kept. If the broker sees the post, he accepts A's request and retrieves the dataset. Then, the broker checks whether the hash of received dataset is identical to the one posted at the bulletin board and blames A if not. Finally, the broker generates and publishes the description of the dataset d (e.g., its contents, price, H(d)).

C. Dataset Examination

If the upload is successful, the broker checks whether a similar dataset has been uploaded before. To do so, we propose uniqueness index, which is indicative of the amount of overlaps between a given set S and a set of sets

S = {S1,S2,........Sn.}

Definition 3 (Uniqueness index). Given a set S = {S1,S2,........Sn }of the uniqueness index of Sx over the set S is
defined as US(Sx) = 1 maxS2Sf (S; Sx)g, where (S; Sx) is a normalized similarity function describing how unique

Sx is when compared to S, defined as:

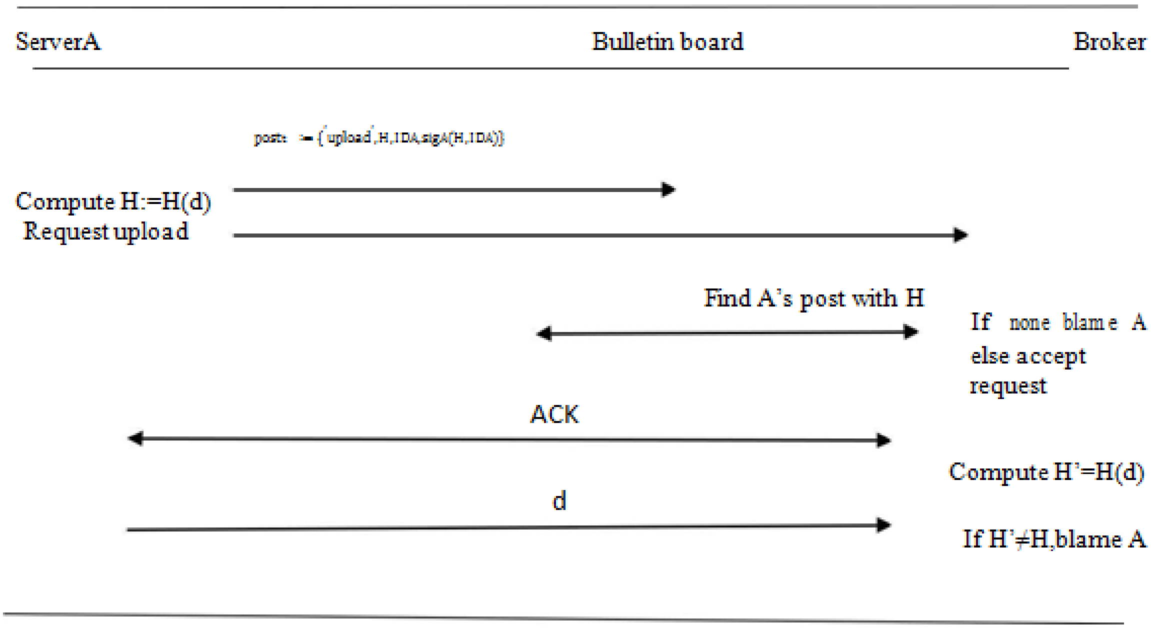
) max(| |, | |)
min(|
|, | |)

J(S1; S2) denotes Jaccard Index, which is statistical mea-surement of the similarity of two given sets, defined as

J(S1; S2)= $^{1\cap}$ ^{2} Then, we define selling of a dataset d asre-selling if U (d) > and as valid selling if

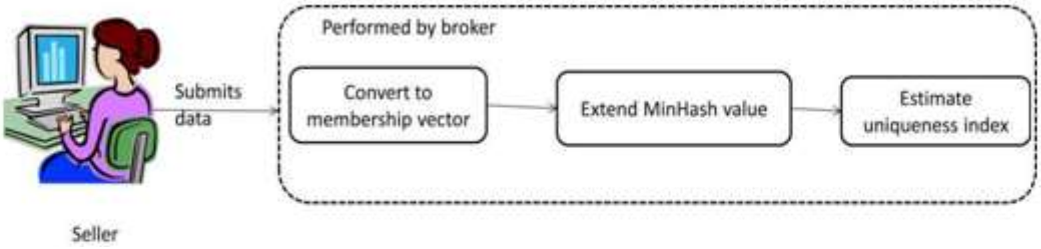
UD(d) < low, where D is the database of datasets the broker possesses, d is the dataset to be examined, and high; low refer to two threshold values for decision making. If the uniqueness index is between the two threshold values, the broker can manually inspect the dataset with human labor. The reason we define and use this uniqueness index in dataset re-selling detection is manifold. Firstly, it intuitively measures how many elements of Sx are similar to the ele-ments in the entire set S, and the multiplier after the Jaccard Index guarantees the index is equal to 1 when Sx is a subset/superset of any set in S. Secondly, in many existing similarity comparison approaches in information retrieval, the datasets are considered as sets of elements (k-grams for texts, feature descriptors for images, and key frames for videos), and therefore the proposed uniqueness index is consistent with them (reviewed in xVII). Thirdly, there is no known similarity comparison mechanism for table-type datasets, and similarity comparison of JSON-like datasets are hardly scalable.

Figure



Upload protocol between a seller A with ID IDA and the broker for uploading dataset

The flow is sketched below to calculate the uniqueness of the document. For a given dataset d, we first convert it to a membership vector. Then, we calculate the MinHash values of the membership vector, which will be used to estimate the uniqueness index.



Uniqunessindex calculation performed by broker

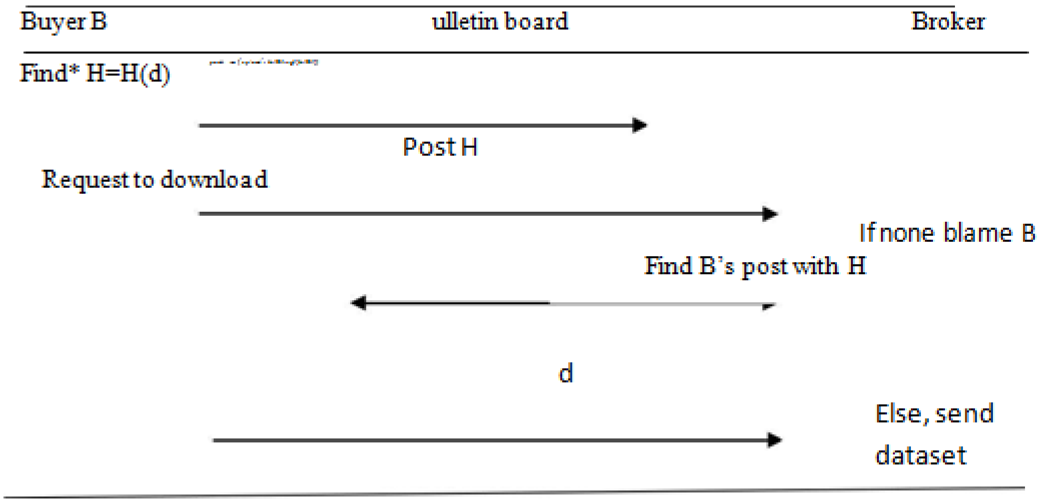
D. Download after Purchase

When a buyer B want to get access to certain dataset d (after reading the description provided by the broker), first he pay for it to the broker and then follows the Download protocol . he posts a declaration postt first at the bulletin board at time t, and then he initiates the download request by sending H(d) to the broker, where H(d) is available in the dataset provided by the broker. The broker finds the corresponding post from the bulletin board

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

and blames B if none is found, because it is evident that he has tried to avoid being book-kept. If the broker sees the post, he accepts B's download request and sends the dataset to B.

Figure



Download protocol between a buyerB with ID IDB and broker for downloading dataset d

E. Parallelization

The most intensive overhead comes from the 1) data file I/O; 2) conversion to membership vector; and 3) generating M MinHash values. the overall execution time isreduced by introducing parallelization.In which Data is logically partitioned into chunks (except JSON/XML types) so that each processor only reads the designated chunk. We carefully design the partitioning among processors so that the membership vector will not be incomplete due to the partitioning. Finally, we compute M MinHash values in parallel since each value is independent of one another. Note that processors can read one file simultaneously in 'Read Only' mode and that both Windows and Linux file systems support random access.

F. Accountability Properties of AccountTrade

Upload

J1: where A is the one who sent the upload request. If the post postt matching H does not exist, the broker states

dis(A) J2: If the posted hash H in postt is different from the calculated hash H^0, the broker
states dis(A). Examine

J3: If the calculated uniqueness index is very low, the dataset is derived from already-uploaded ones, the broker states dis(A) where A is the one who uploaded the dataset.
Download
J4: Same as J1 except that dis(B) is stated instead, where B is the one who sent the request. J1 detects a dishonest seller who tries to refuse a sale transaction, and J2 further prevents a dishonest seller from declaring a wrong dataset. J3 detects reselling, and J4 detects a dishonest buyer who tries to refuse a purchase transaction.

4. CONCLUSION

This paper presents AccountTrade which assurance correct book-keeping and achieves accountability in the big data trading among dishonest consumers. In data transaction AccountTrade blames dishonest consumers if they deviate from their responsibilities. To achieve accountability against dishonest sellers who may resell others' datasets to find the uniqueness of document – uniqueness index – which is efficiently computable. We formally defined two accountability models. we also evaluated the performance and QoS using real-world datasets in our implemented testbed.

REFERENCES

[1] Data markets compared – a look at data market offerings from four providers. Goo.gl/k3qZsj.

[2] Ftc charges data broker with facilitating the theft of millions of dollars from consumers' accounts. Goo.gl/7ygm7Q.

[3] Ftc charges data brokers with helping scammer take more than $7 million from consumers' accounts. Goo.gl/kZMmXn.

[4] Ftc complaint offers lessons for data broker industry. goo.gl/csBYA3.

[5] Multimedia computing and computer vision lab. goo.gl/pbKeCj

[6] R. Ara´ujo, S. Foulle, and J. Traor´e. A practical and secure coercionresistant scheme for remote elections. In Dagstuhl Seminar Proceedings. Schloss Dagstuhl-Leibniz-Zentrum f¨ur Informatik, 2008.

[7] D. Baltieri, R. Vezzani, and R. Cucchiara. Sarc3d: a new 3d body model for people tracking and re-identification. In ICIAP, pages 197– 206. Springer, 2011.

[8] B. Blanchet. Automatic verification of security protocols in the symbolic model: The verifier proverif. In FOSAD, pages 54–87. Springer, 2014.

[9] B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM, 13(7):422–426, 1970.

[10] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In SIGMOD, volume 24, pages 398–409. ACM, 1995.

[11] A. Z. Broder. On the resemblance and containment of documents. In Compression and Complexity of Sequences, pages 21–29. IEEE, 1997.

[12] A. Z. Broder, S. C. Glassman, M. S. Manasse, and G. Zweig. Syntactic clustering of the web. Computer Networks and ISDN Systems, 29(8):1157–1166, 1997.

[13] X. Cao, Y. Chen, and K. R. Liu. An iterative auction mechanism for data trading. In ICASSP, pages 5850–5854. IEEE, 2017. M. S. Charikar. Similarity estimation techniques from rounding algorithms. In STOC, pages 380–388. ACM, 2002. O. Chum, J. Philbin, M. Isard, and A. Zisserman. Scalable near identical image and shot detection. In CIVR, pages 549–556. ACM, 2007.

[14] F. T. Commission et al. Data brokers: A call for transparency and accountability. 2014

[15] S. A. Cook. The complexity of theorem-proving procedures. In STOC, pages 151–158. ACM, 1971.

[16] S. Delgado-Segura, C. P´erez-Sol`a, G. Navarro-Arribas, and J. Herrera- Joancomart´ı. A fair protocol for data trading based on bitcoin transactions. Future Generation Computer Systems, 2017.

[17] M. Douze, H. J´egou, and C. Schmid. An image-based approach to video copy detection with spatio temporal post-filtering. Transactions on Multimedia, 12(4):257–266, 2010.

[18] L. Fan, P. Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. TON, 8(3):281–293, 2000.