BDmark: A Blockchain-driven Approach to Big Data Watermarking

Swagatika Sahoo[1], Rishu Roshan[2], Vikash Singh[2], and Raju Halder[1]

[1] Indian Institute of Technology Patna, India
{swagatika_1921cs03,halder}@iitp.ac.in
[2] Indian Institute of Information Technology Guwahati, India
{rishuroshan.1998, vik625singh}@gmail.com

Abstract. Big data, as a driving force to the business growth, creates a new paradigm that encourages large number of start-ups and less-known data brokers to adopt data monetization as their key role in the data marketplace. As a pitfall, such data-driven scenarios make big data prone to various threats, such as ownership claiming, illegal reselling, tampering, etc. Unfortunately, existing watermarking solutions are ill-suited to big data due to a number of challenging factors, such as V's of big data, involvement of multiple owners, incremental watermarking, large cover-size and limited watermark-capacity, non-interference, etc. This paper presents a novel approach BDmark that provides a transparent immutable audit trail for data movement in big data monetizing scenarios, by exploiting the power of both watermarking and blockchain technologies. We describe in detail how our approach overcomes the above-mentioned challenging factors. As a proof of concept, we present a prototype implementation of the proposed system using Java and Solidity on Ethereum platform and the experimental results on smart contracts show a linear growth of gas consumption w.r.t. input data size. To the best of our knowledge, this is the first proposal which deals with watermarking issues in the context of big data.

Keywords: Digital Watermarking · Big Data · Blockchain · Smart Contract.

1 Introduction

With explosive increase of global data over the last decade, big data has become a powerful resource which is playing a leading role in the transformation of existing business models [3]. This targets not only business analytics process, but also academic, research, and other decision making activities. Thanks to the growth of social media, the increasing number of smart connected devices, ubiquitous network access, and many others. This new paradigm, on the other hand, offers an establishment of large number of start-ups companies who sell and purchase our personal data on a daily basis. Few, among many others, include Datacamp, Datawallet, Dawex, etc [3]. Even most of us are unknown

[3] https://www.datacamp.com, https://www.datawallet.com/, https://www.dawex.com/en/

about the existence of less-known data-brokers who gain profit by gathering, aggregating, analyzing, hoarding, commodifying, trading or using personal data without our knowledge or consent [11]. There are many other situations where data monetization is an integral and indispensable part of a system in the form of data-as-a-service model [14]. Some interesting fields spawned and co-existing with the use of big data are machine learning, deep learning, artificial intelligence, data-science, etc., which may demand training dataset in a pay-per-use fashion. In this context, example like census data collection shows its relevancy, where the task is outsourced to a large number of organizations in a hierarchical fashion to collect data locally at individual-levels and then to combine them together towards the higher levels, covering data-collection over a large geographical areas.

The above data driven scenarios give rise to a major concern related to our fundamental right to privacy and security. Moreover, such data is prone to various activities, such as piracy, illegal reselling, tampering, illegal redistribution, ownership claiming, forgery, theft, misappropriation, etc. Digital watermarking has emerged as a promising technique to address these challenges [4]. A watermark is considered to be some kind of information that is embedded into underlying data and is extracted later to prove the absence of above-mentioned activities. In this process, a watermark W is embedded into an original data-piece using a private key K which is known only to the owner. On receiving any suspicious data-piece, the owner may perform a verification process using the same private key K by extracting and comparing the embedded watermark (if present) with the original watermark information W.

There are large number of watermarking approaches exist in the literature, targeting a variety of digital contents such as databases, images, audio, video, text data, etc. [4, 5, 7], however they are found unfit to adopt to the case of big data watermarking due to the following challenges: V's of big data, involvement of multiple owners, incremental watermarking, large cover-size and limited watermark-capacity, non-interference, etc [3]. On the other hand, four contributions [15, 8, 6, 16] in the literature exist, which target specific kind of big data (such as social network data, geographical big data, and large data algebraic graph) as a cover to watermark. However, we observed that none of them addresses the above-mentioned challenges. In recent years, blockchain technology [10] has become a source of new hope with its broad spectrum of applications in practice. Examples include supply-chain, insurance, real-estate, financial services, and many more [12, 13]. We are witnessing its footstep in digital watermarking as well. Few recently proposed watermarking solutions using blockchain technology are reported in [17, 9, 2, 1]. Although, the use of blockchain in the above-mentioned proposals fulfils a common interest to improve the verifiability of the ownership in a distributed settings, unfortunately they are not meant for big data scenarios. In [16], although the proposal keeps watermarked data sharing records in blockchain, this lacks in various aspects, such as this has not taken care of big data properties, access control, etc.

As observed, none of the techniques in the literature shows its competence to be applied directly in the realm of big data watermarking by respecting the challenges we already highlighted before. To fill this research gap, in this paper we propose a novel approach BDmark that provides a transparent immutable audit trail for data movement in big data monetizing scenarios, by exploiting both watermarking and blockchain technologies power. To the best of our knowledge, this is the first proposal which deals with watermarking issues in the context of big data. To summarize, our main contributions in this paper are:

– We consider a complex big data monetization scenario where multiple actors are involved in data collection, aggregation, storage, selling, and purchasing.
– We propose a novel approach which provides a transparent immutable audit trail for data movement, by exploiting the power of both watermarking and blockchain technologies.
– We devise an access control mechanism to ensure the legitimate buyers, on receiving data-seller's permission, can get access to the target data-piece.
– We show in detail how the approach overcomes the present challenging factors in big data watermarking settings.
– Finally, as a proof of concept, we present a prototype implementation of the proposed system using Java and Solidity on Ethereum platform and report experimental results.

The structure of the paper is organized as follows: Section 2 describes the related works in the literature. In Section 3, we identify the challenging factors involved in big data watermarking. The detail description of our proposed approach is presented in Section 4. The attack analysis is performed in section 5. Section 6 presents proof of concept and experimental results. Finally, Section 7 concludes our work.

2 Related Works

The authors in [6] proposed a reversible watermarking of social network data. The approach, before watermark encoding, adopted a data pre-processing phase, where features of numeric and non-numeric datasets are selected and encoded into the generated watermark. Moreover, the Genetic Algorithm in case of the numeric dataset and two operations - hashing and permutations - for the non-numerical dataset are applied. [15] presented the watermarking of large data algebraic graphs using a deep belief network. Aiming to solve the problem of interpretation attack, a zero-watermarking scheme for vector map, a type of geographical big data, is proposed in [8] based on the feature points of vector maps. In [17], the authors introduced BMCProtector, a prototype implementation based on blockchain and smart contracts, to protect music copyright issues and to ensure income rights/incentives to original holders or owners. The deployed smart contract is responsible to share the copyright parameters of the music owners and to transfer cryptocurrency to their wallet when purchasing events take place. A new design approach for copyright management of image files based on digital watermarking and blockchain is proposed in

[9]. The approach stores owner's digital signatures along with images' cryptographic hashes in the blockchain and the corresponding blocks information along with watermarked images in an interplanetary file system. In [2], the authors designed a proof of concept prototype which provides an online verification platform to verify the integrity of the recorded video. With the help of blockchain technology, the approach stores cryptographic hashes of video contents in a chronological chained link to establish an irrefutable database. A blockchain-based multimedia watermarking framework is proposed in [1]. This allows the retrieving of either the transaction trails or the modification histories of an image and preserves retrievable original media content identifying the edited/tampered regions. The framework proposed in [16] facilitates the sharing of watermarked data, keeping records in the underlying blockchain, which is limited to numerical data only. A Comparative Summary in this direction, as compared to BDmark, is depicted in table 1 where BC stands for Blockchain.

Table 1: A Comparative Summary w.r.t. Literature

Metric / Proposals	Is BC-based ?	Cover Type	Blockchain Type	Storage Type	Support Access Control ?	Dealing BigData?
[6]	N	Social network dataset	NA	NA	N	Y
[15]	N	Algebric graph	NA	NA	N	Y
[8]	N	Vector map	NA	NA	N	Y
[2]	Y	Video	Ethereum	Android devices + BC	N	N
[9]	Y	Image	NA	IPFS + BC	N	N
[17]	Y	Music	Ethereum	IPFS + BC	Y	N
[1]	Y	Multimedia	Ethereum	Media database server + BC	N	N
[16]	Y	Numerical data	Fabric	BC	N	Y
BDmark	Y	All types of data	Ethereum	IPFS + BC	Y	Y

3 Challenges in Big data Watermarking

Let us identify a number of challenging factors that lie with the big data watermarking [3, 4]:

1. **Capacity.** The most important challenge in big data watermarking is *capacity*. It determines the optimum amount of data that can be embedded in a cover and the optimum way to embed and extract this information. Due to "*volume*" property of big data, this needs to be ensured that the embedded watermark is spread over all parts of the data. Moreover, if multiple marks are inserted into such large cover of big data, they should not interfere with each other.
2. **Incremental Watermarking.** The second property "*velocity*" of big data gives rise to another challenge: the generation of streaming data requires an adoption of incremental watermarking approach to ensure that the watermarking algorithm should consider only the newly added or modified data for the watermark, keeping the unaltered watermarked-data untouched.
3. **Usability.** This is quite natural to assume that more than one actors participate in the process of big data collection. In addition to the original data-owners (who collects the data from the environment at individual levels), a number of data-collectors are also involved. The sole responsibility of data-collectors is to collect and aggregate data coming from either original data-owners or other data-collectors. Notably, in a more generic practical situation, data-collectors may also claim the ownership of the collected data, in

addition to original data-owners. In such complex situation, the embedding of large number of ownership signatures, each corresponds to individual data-owner or -collector, may degrade the usability of big data.

4. **Security.** The involvement of large volume of data and many ownerships make it a natural choice to embed multiple signatures into the big data. Such setting pushes watermarked-data towards a number of attacks, including subset attack, superset attack, collusion attack, etc.
5. **Public Verifiability.** The major drawback in private watermarking scheme is single-time verifiability, due to the possibility of revealing private parameters during the verification process to take place for the first time. In contrary, watermarking scheme based on public parameters overcome such limitations and anyone can verify the ownership at any time publicly. Big data watermarking requires a public treatment because of huge cover-size and multiple ownership involved, especially when big data can be split, aggregated and shared in pieces among many actors.
6. **Trust.** Big data collection, storage, sharing, etc., requires the support of cloud-based infrastructure. In such case, trust is a concern when we are providing watermark solution to big data and at the same time relying on semi-trusted or even untrusted third party cloud service provider.
7. **Traceability.** Traditional watermarking approaches do not support traceability. Although this may not be required in case of simple and restrictive scenarios involving single owner and limited sized data, this of course carries an important and impactful role in a complex scenarios like our case.

4 BDmark: A Blockchain-driven Big Data Watermarking

The proposed system involves the following actors: (1) Data-owners, (2) Data-collectors, and (3) Data-buyers. Data-owners are those persons or organizations who are responsible to generate data from the environment. The task of data-collectors is to collect data either from data-owners or from other data-collectors and to transfer them further to another set of data-collectors. Since we are considering data as purchasing and selling items, there exist many data-buyers in the system who need data to buy either from the original owner or from the data-collectors. All these actors are connected in a peer-to-peer network which maintains a blockchain and records all kinds of interactions among the actors. The overall system components are shown pictorially in Figure 1. The smart contracts deployed in the blockchain are responsible to perform crucial functionalities, such as registration, traceability, access control, etc. Our proposed approach BDmark comprises of the following phases: Joining to Blockchain Network, Registration, Data Collection and

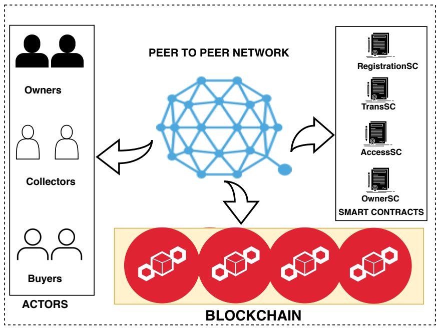

Fig. 1: A pictorial representation of the overall system components

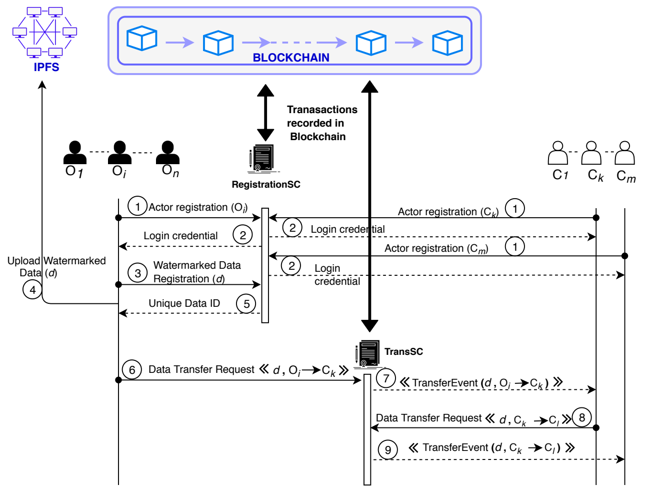
IPFS
BLOCKCHAIN
Tranasactions recorded in Blockchain
O_1
O_i
O_n
RegistrationSC
C_1
C_k
C_m
1 Actor registration (O_i)
Actor registration (C_k) 1
2 Login credential
Login credential 2
Actor registration (C_m) 1
2 Login credential
Upload Watermarked Data (d)
4
3 Watermarked Data Registration (d)
Unique Data ID 5
TransSC
6 Data Transfer Request ≪ d, $O_i \rightarrow C_k$ ≫
7 ≪ TransferEvent (d, $O_i \rightarrow C_k$) ≫
Data Transfer Request ≪ d, $C_k \rightarrow C_l$ ≫ 8
9 ≪ TransferEvent (d, $C_k \rightarrow C_l$) ≫

Fig. 2: Interaction-diagram among owners, collectors and smart contracts

Storage and Data Monetization. Figure 2 depicts the interactions among data-owners, data-collectors and smart contracts which take place during the phases 2 and 3, whereas Figure 4 depicts the same among data-buyers and smart contracts which take place during phase 4. Let us now describe below each of the phases in detail.

4.1 Joining to Blockchain Network

The first and foremost step for an actor to participate in the system is to join the blockchain network through specially designated nodes, known as seed nodes. This simply starts by exchanging joining messages between new node and any one seed node. As a result, the seed node returns addresses of its neighbouring nodes with whom the new node can establish peer to peer connections. This process is repeated until a satisfactory number of peers is found.

4.2 Registration

This phase is responsible to provide a unique credential to all actors who participate in the system. This is crucial to ensure that only legitimate actors are performing only those transactions for which they are allowed. Observe that additional use of biometric-based authentication through national identity database may improve the security of the registration process. The smart contract RegistrationSC is responsible for the registration process. The interactions among data-owners O_i, data-collectors C_k and data-buyers B_j with RegistrationSC are indicated by the steps $\textcircled{1}$ and $\textcircled{2}$ in Figures 2 and 4 respectively.

4.3 Data Collection and Storage

This is the core phase in our approach to ensure the copyright protection of big data. We assume the presence of multiple actors (owners and collectors) in the process of big data collection, aggregation and storage. The data-collection from the environment would be performed by data-owners who then transfer the collected data-pieces to another collectors. After aggregating the incoming data-pieces, a collector may send it to another set of collectors as well. In contrast to the data-aggregation operation, the approach also supports data-splitting when a collector needs to transfer different segments of the collected-data to different collectors. Let us now describe all the activities separately which are involved in this phase:

Watermarking of Owner's Data. After collecting data from the environment, an owner must embed his/her signature (as watermark information) into the data. This enables the owner to claim the ownership of his/her data anytime later. Assuming the size of the collected data-pieces at individual level is limited, one can apply any suitable existing watermarking technique [4] for this purpose.

Registering Watermarked-Data. After successful watermark embedding, data-owners register their watermarked-data by invoking `RegistrationSC` and by uploading it on the IPFS. Like actor's registration, this step also assigns an unique identity to the watermarked-data in the form of two-tuple ⟨*uniqueID*, *data-hash* ⟩. These interactions are shown by the steps (3), (4) and (5) in Figure 2. Please note that, from now onwards, we use the terms 'data' and 'watermarked-data' interchangeably when there is no ambiguity in the context.

Data Transfer. This phase allows the transfer of data either from an owner to a collector, or from a collector to another collector. The interactions are shown by the steps (6) - (9) in Figure 2. The smart contract `TransSC` is responsible for these tasks. To improve the scalability of the system, the operations on data, such as aggregation and splitting, are achieved with the help of token creation. Figure 3 depicts a scenario involving various operations on data through token generation. For example, $Token_1 : \langle d_1, d_2, d_3 \rangle$ represents a new token (named as $Token_1$) which consists of three data units d_1, d_2 and d_3. On the other hand, $Token_3$ consists of d_1, d_2 and d_5 where d_1 and d_2 are obtained from $Token_1$. Formally, we denote by '⟨ ⟩' as data-aggregation operation, whereas '✂{ }' as data-splitting operation. The symbol ':' denotes 'named as' operation. Observe that a token may contain another token as well. The overall algorithm to perform data-transfer and data-traceability is depicted in Algorithm 1.

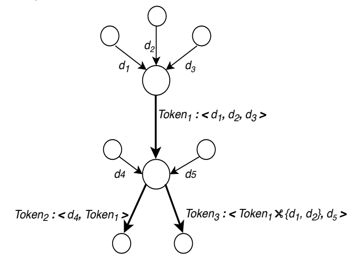
d_1
d_2
d_3
$Token_1 : < d_1, d_2, d_3 >$
d4
d5
$Token_2 : < d_4, Token_1 >$
$Token_3 : < Token_1$ ✂$\{d_1, d_2\}, d_5 >$

Fig. 3: Data transfer scenarios involving data-collection, aggregation and splitting through token generation

Let us now briefly describe the algorithmic steps. The algorithm takes as inputs the unique identifiers of sender and receiver (which are generated in the registration phase) and the collection D of data for which ownership needs to be transferred.

Algorithm 1: TransferOwnership

```
Input  : Sender $A_i$, Receiver $A_j$, Data-collection $D$
Output: Transfer id, Token id

1  flag=true;
2  $S=\emptyset$;
3  for all basic data block $d_i \in D$ do
4      $S=S \cup d_i$;
5  for all token $t_i \in D$ do
6      $X$=getBasicDataBlocks($t_i$);
7      $S=S \cup X$;
8  for all splittedtoken $d_j \in t$✄$\phi$ do
9      $X$=getBasicDataBlocks($\phi$);
10     $S=S \cup X$;
11 for all basic data block $d_k \in S$ do
12     if $A_i \notin$ getCurrentOwner($d_k$) then
13         flag=false;
14         exit();
15 if $flag==true$ then
16     for all basic data block $d_k \in S$ do
17         replace $A_i$ by $A_j$ in the current owner
             getCurrentOwner($d_k$);
18         Append $A_j$ to the list InheritOwners($d_k$);
19 Create a new token $l$ to uniquely identify $D$;
20 Store the token id $t$ corresponding to $D$ denoted $t$:$D$ with
     ownership $A_j$ and the transfer id $h$;
21 Create a new Transfer id $h$;
22 Store the token id $t$ corresponding to $D$ denoted $t$:$D$ with
     ownership $A_j$ and the transfer id $h$;
23 Return $h$ and $t$;
```

Algorithm 2: GetWatermarkedData

```
Inputs : Buyer's identity $B_i$, Data Collection $D$
Output: Hash links to download $D$

1  $B_i$ requests to the smart contract AccessSC to purchase data $D$;
2  AccessSC interacts with the smart contracts RegistrationSC and TransSC to
     check whether $B_i$ and $D$ are registered, and gets the list of $D$'s owners
     $L_{owner}$ and their personal smart contract addresses and names $L_{SC}$.
3  if $B_i$ is valid buyer then
4      if $D$ is registered data then
5          Ask $B_i$ to select one owner $O_j$ from the list $L_{owner}$ of eligible
             owners;
6          $Addr$:= getOwnerSCaddress($O_j$, $L_{SC}$);
7          $Name$:= getOwnerSCname($O_j$, $L_{SC}$);
8          $flag$:=true;
9          Generate an unique interaction id $I$;
10         $\tau$=getCurrentTime();
11         Store a new tuple $\langle I, B_i, D, O_j, \tau, Addr, Name, flag \rangle$ into
             LookTab;
12         Return $Addr$ to the buyer;
13 $B_i$ invokes OwnerSC, deployed to the address $Addr$, with inputs $\langle B_i, D, O_j \rangle$;
14 OwnerSC contacts AccessSC by passing $\langle B_i, D, O_j \rangle$;
15 AccessSC checks the look-up table LookTab;
16 if $\exists \langle B_i, D, O_j, \tau, Addr, Name \rangle \in$ LookTab then
17     $\tau_1$=getCurrentTime();
18     Return interaction id $I$ to OwnerSC;
19 OwnerSC sends $I$ to the buyer $B_i$;
20 $B_i$ requests AccessSC for $D$ supplying $\langle I, B_i, D, O_j \rangle$;
21 $\tau_2$=getCurrentTime();
22 if $\tau_2 - \tau_1 \leqslant \delta$ then
23     if Verify($\langle I, B_i, D, O_j \rangle$, LookTab)==$success$ then
24         if flag$\neq$ $false$ then
25             return IPFS hash link of $D$ to $B_i$;
26             Set $flag$:= false;
```

Observe that the data-collection may contain basic data blocks or previously generated tokens or splitted tokens representing other data-collections. Steps 3 to 7 identify a set S of all basic data blocks belonging to D by enumerating it recursively, whereas steps 11 to 12 verify whether the transfer request is issued by legitimate owner A_i by checking the current ownership of all data blocks in S as per the record in blockchain smart contract state variables. On successful verification, *flag* variable remains *true*, and steps 15 to 18 updates the ownership. Observe that, in addition to the current ownership change at step 17, the algorithm maintains a trace of ownership changes by appending all new owners to a list. Finally, a token identifier representing this data-collection and a transaction identifier are generated and stored.

4.4 Data Monetization and Access Control

This phase is designed to enable data-owners or data-collectors to sell their data to registered data-buyers. The interaction diagram is depicted in Figure 4. As usual, to participate in the system, data-buyers first register themselves by invoking RegistrationSC, depicted by steps ① and ②. Whenever a data-buyer wants to buy a chunk of data, the whole process must pass through an access control mechanism involving the steps ③ - ⑮ among data-buyers and two smart contracts AccessSC and OwnerSC. A look-up table LookTab is maintained by AccessSC to record some crucial information, like unique identifier for a particular data request, the time-stamp when the request is made, etc. The overall algorithm is depicted in Algorithm 2. Let us now explain the algorithmic steps and the corresponding interactions in detail. When a legitimate data-buyer B_i requests for registered data D to AccessSC, on successful verification in steps 3 and 4 through invocation of RegistrationSC and TransSC, an entry containing a unique interaction id, buyer-seller information, time-stamp of the request, and details of data-owner's smart contract are stored in LookTab in step 11. These are shown by the steps ③ and ④ in Figure 4. Observe that the *flag* value indicates whether the current request is already fulfilled or yet to fulfill. On receiving the address of owner's smart contract OwnerSC, the data-buyer invokes it by passing the request information $\langle B_i, D, O_i \rangle$ in step 13. In order to check weather to give permission or not, OwnerSC will contact AccessSC to check the validity of the request in step 14. These are shown by the steps ⑤, ⑥ and ⑦ respectively in Figure 4. If an entry corresponding to this request is found in LookTab (step 16), the validity is ensured and therefore AccessSC returns the corresponding I to OwnerSC who then passes it to the buyer as an indication of "data access permission". These are shown in algorithmic steps 17 to 19 and by steps ⑧, ⑨, ⑩ and ⑪ in Figure 4. Finally, on producing the same interaction id by the buyer in step 20, AccessSC determines that owner has given the permission and the hash links to download D from IPFS are therefore provided. Accordingly, the value of *flag* is set to *false*, in order to prevent the repeated download of the same data against the same I. These interactions are shown by the steps ⑫ to ⑯ in Figure 4. Observe that δ represents the maximum time gap, starting from the receiving of I to the production of the same to AccessSC by the data-buyer.

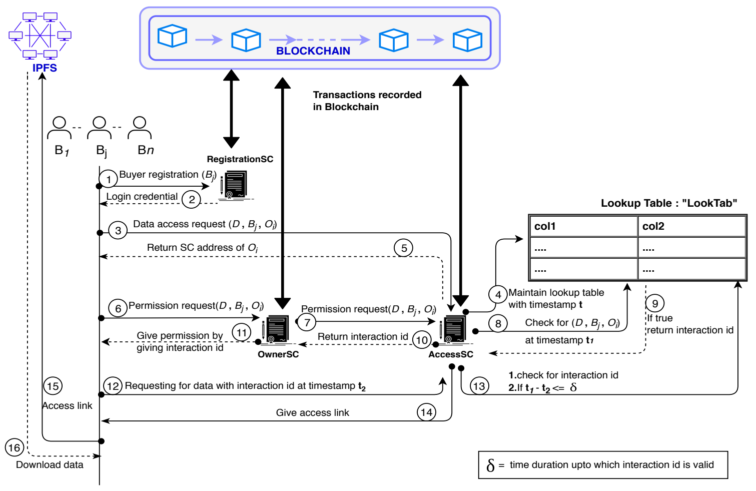
IPFS
BLOCKCHAIN
Transactions recorded in Blockchain
B1
Bj
Bn
RegistrationSC
1 Buyer registration (Bj)
Login credential 2
3 Data access request (D, Bj, Oi)
Return SC address of Oi 5
Lookup Table : "LookTab"
col1
col2
....
....
....
....
4 Maintain lookup table with timestamp t
8 Check for (D, Bj, Oi) at timestamp t1
9 If true return interaction id
6 Permission request(D, Bj, Oi)
7 Permission request(D, Bj, Oi)
Give permission by giving interaction id 11
Return interaction id 10
OwnerSC
AccessSC
15 Access link
12 Requesting for data with interaction id at timestamp t2
13 1.check for interaction id 2.If t1 - t2 <= δ
Give access link 14
16 Download data
δ = time duration upto which interaction id is valid

Fig. 4: Interaction-diagram between data-buyers and smart contracts.

5 Attack Analysis

In this part, we discuss few relevant attacks and their preventive-measures addressed by the proposed approach.

- **Denial-of-Service (DoS) attack.** DoS attacks may take place in the system aiming to prevent data access services to other legitimate buyers. A flood of data request messages may intentionally be sent to the corresponding smart contract AccessSC keeping it throughout busy. The thawrt of DOS attacks is achieved by discouraging the attackers as Gas price is involved in every execution of AccessSC. Where gas price is the total cost of a transaction.
- **Man in the middle attack.** Man in the middle attack is possible when an attacker intercepts the communication between buyer and owner, and get access to the interaction id which she may present later to the smart contract AccessSC in order to get data-access. This is impossible in the proposed framework because of the following factors: (1) The login mechanism using pre-approved credentials, (2) The sharing of interaction id through registered media, *e.g.* mobile, (3) One time use of generated interaction id for a given owner-buyer combination at a specific time-stamp, (4) The access restriction within a predefined time interval.
- **Attacks on Watermark:** Big data is highly susceptible to the following attacks: (a) Subset Attack, (b) Superset Attack, (c) Collusion Attack, (d) Value Modification Attack, etc. The proposed system takes care of big data ownership by embedding watermarks in basic data blocks at fine-grained granularity level after collected by data-owners at individual levels. We rely on the security strength of the existing watermarking techniques, to ensure the success of ownership claim in future. Moreover, in addition to this, the association of hash value of basic data blocks during registration also helps to perform tamper detection. Since the blockchain network is pseudonymous, this is an advantageous step to weaken the collusion attacks.

6 Proof of Concept and Experimental Results

We have implemented a prototype to analyze the feasibility of the proposal. The programming languages we used in the implementation are Java and Solidity. We have created a simple Ethereum DApp using Web3.js and Truffle and set up monitoring of the API transactions sent to the blockchain. We have used Truffle framework, as it provides a set of tools and boilerplate code for scaffolding DApps for Ethereum. In the current implementation, we have provided the complete functionalities in the form of two Smart Contracts: (i) RegistrationTransSC which implements the functions of both the Smart Contracts RegistrationSC and TransSc, that includes Actors Registration, Data Registration, and Ownership Transfer. (ii) AccessOwnerSC which implements the functions of AccessSC and OwnerSC, that includes Data Request, Access Policy, Buyer's Validation, and Permission. We have designed a web interface to interact with the contracts that assume three roles: Actors Registration, Data Registration, and

Table 2: Transaction gas costs for remaining smart contract functionalities

Sl. No.	Functions	Gas Used	Actual Cost (*Ether*)	*USD* ($)
1	*RegistrationTransSC*	4724439	0.009448878	2.92
2	*AccessOwnerSC*	568945	0.001137890	0.35
3	*createUser*	114041	0.000228082	0.07
4	*newData*	190189	0.000380378	0.11
5	*get_data_owners*	22888	0.000045776	0.014
6	*getdata_tracking_ids*	22954	0.000045780	0.014
7	*getdata_trackindes*	25435	0.000050870	0.015
8	*userLogin*	27857	0.000055714	0.017
9	*getdetails*	239230	0.000478460	0.14
10	*concatenateInfoAndHash*	29603	0.000059206	0.018

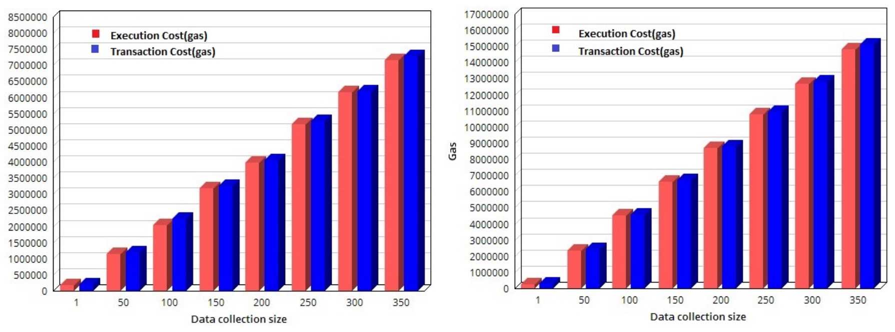

Fig. 5: Gas price analysis of TransferOwnership and GetWatermarkedData

Data Monetization. Each has its own view (page) and we keep them separated to better demonstrate how different actors could use the contracts. Figures 5(a) and 5(b) depict the experimental results which show the variation of gas cost for the execution of the modules in RegistrationTransSC and AccessOwnerSC corresponding to TransferOwnership (Algorithm 1) and GetWatermarkedData (Algorithm 2) respectively w.r.t. various size of data collection D. Observe that the gas cost complexity is linear w.r.t. the input data collection size. Table 2 depicts the transaction gas costs for the remaining functionalities where input (and hence gas cost) remains same all the time.

7 Conclusion

This paper presents a novel approach to provide transparent immutable audit trail for data movement in big data monetizing scenarios. The proposal addresses a number of common data breaches in the world of big data by strengthening the use of blockchain technology on top of existing watermarking techniques at fine-grained levels. Interestingly, the approach is immune to several common attacks and its feasibility is established through a proof of concept. To the best of our knowledge, this is the first proposal which deals with watermarking issues in the context of big data.

References

1. Bhowmik, D., Feng, T.: The multimedia blockchain: A distributed and tamper-proof

media transaction framework. In: Proceedings of 22nd International Conference on Digital Signal Processing (DSP). pp. 1–5. IEEE, London, UK (Aug 2017)

2. Billström, A., Huss, F.: Video Integrity through Blockchain Technology. KTH VETENSKAP OCH KONST, Stockholm, SWEDEN (Aug 2017)
3. Chen, M., Mao, S., Liu, Y.: Big data: A survey. Mob. Netw. Appl. **19**(2), 171–209 (2014)
4. Halder, R., Pal, S., Cortesi, A.: Watermarking techniques for relational databases: Survey, classification and comparison. Journal of Universal Computer Science **16**(21), 3164–3190 (2010)
5. Hartung, F., Girod, B.: Watermarking of uncompressed and compressed video. Signal Processing **66**(3), 283 – 301 (1998)
6. Iftikhar, S., Kamran, M., Munir, E.U., Khan, S.U.: A reversible watermarking technique for social network data sets for enabling data trust in cyber, physical, and social computing. IEEE Systems Journal **11**, 197–206 (2017)
7. Kamaruddin, N.S., Kamsin, A., Por, L.Y., Rahman, H.: A review of text watermarking: Theory, methods, and applications. IEEE Access **6**, 8011–8028 (2018)
8. Liu, Y., Yang, F., Gao, K., Dong, W., Song, J.: A zero-watermarking scheme with embedding timestamp in vector maps for big data computing. Cluster Computing **20**, 3667–3675 (2017)
9. Meng, Z., Morizumi, T., Miyata, S., Kinoshita, H.: Design scheme of copyright management system based on digital watermarking and blockchain. In: Proceedings of IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). pp. 359–364. IEEE, Tokyo, Japan (Jul 2018)
10. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. In: Online: http://bitcoin.org/bitcoin.pdf (Mar 2009)
11. Parra-Arnau, J.: Optimized, direct sale of privacy in personal data marketplaces. Information Sciences **424**, 354 – 384 (2018)
12. Pilkington, M.: Blockchain technology: Principles and applications. In: Handbook of Research on Digital Transformations. Edward Elgar (2016)
13. Sahoo, S., Fajge, A.M., Halder, R., Cortesi, A.: A hierarchical and abstraction-based blockchain model. Applied Sciences **9**(11) (2019)
14. Terzo, O., Ruiu, P., Bucci, E., Xhafa, F.: Data as a service (daas) for sharing and processing of large data collections in the cloud. In: Proceedings of Seventh International Conference on Complex, Intelligent, and Software Intensive Systems. pp. 475–480. IEEE, Taichung, Taiwan (Jul 2013)
15. Wangming, Y.: The digital watermarking algorithm based on the big data algebra graoh. In: Proceedings of International Conference on Robots & Intelligent System. pp. 342–345. IEEE, Huaian, China (Nov 2017)
16. Yang, J., Wang, H., Wang, Z., Long, J., Du, B.: BDCP: A Framework for Big Data Copyright Protection Based on Digital Watermarking (Dec 2018)
17. Zhao, S., O'Mahony, D.: Bmcprotector: A blockchain and smart contract based application for music copyright protection. In: Proceedings of International Conference on Blockchain Technology and Application. pp. 1–5. ACM, USA (Dec 2018)