A blockchain-based data usage auditing architecture with enhanced privacy and availability

Nesrine Kaaniche, Maryline Laurent

To cite this version:

Nesrine Kaaniche, Maryline Laurent. A blockchain-based data usage auditing architecture with enhanced privacy and availability. NCA 2017 : 16th IEEE International Symposium on Network Computing and Applications, Oct 2017, Cambridge, United States. IEEE Computer Society, Proceedings NCA 2017 : 16th IEEE International Symposium on Network Computing and Applications, pp.1 - 5, 2017, <10.1109/NCA.2017.8171384>. <hal-01682227>

HAL Id: hal-01682227
https://hal.archives-ouvertes.fr/hal-01682227

Submitted on 12 Jan 2018

A Blockchain-based Data Usage Auditing Architecture with Enhanced Privacy and Availability

Nesrine Kaaniche and Maryline Laurent

SAMOVAR, Télécom SudParis, CNRS, Université Paris-Saclay, France
Member of the Chair Values and Policies of Personal Information

Abstract—**Recent years have witnessed the trend of increasingly relying on distributed infrastructures. This increased the number of reported incidents of security breaches compromising users' privacy, where third parties massively collect, process and manage users' personal data. Towards these security and privacy challenges, we combine hierarchical identity based cryptographic mechanisms with emerging blockchain infrastructures and propose a blockchain-based data usage auditing architecture ensuring availability and accountability in a privacy-preserving fashion. Our approach relies on the use of auditable contracts deployed in blockchain infrastructures. Thus, it offers transparent and controlled data access, sharing and processing, so that unauthorized users or untrusted servers cannot process data without client's authorization. Moreover, based on cryptographic mechanisms, our solution preserves privacy of data owners and ensures secrecy for shared data with multiple service providers. It also provides auditing authorities with tamper-proof evidences for data usage compliance.**

I. Introduction

With a data driven society, there is a growing public concern about users' privacy and data secrecy preservation. Centralized public and private companies collect large amounts of personal and sensitive information. However, users have little or no control over the data collected and stored about them and how they are used. Several approaches have been introduced in order to address data secrecy and privacy issues, from both legislative and technological perspectives. Indeed, strong authentication and authorization mechanisms based on centralized trusted authorities, emerged for protecting identifying information and ensuring privacy' preserving access to services.

In 2016, the European Union (EU) adopted a new General Data Protection Regulation (GDPR) that compels new obligations on service providers [3], reinforcing the protection of personal data. Several key obligations have to be considered under GDPR, such as, (i) it extends the liability and accountability requirements for organizations and (ii) it requires the consent from data owners.

Recently, various accountable systems appeared, such as *Bitcoin* which enables users to transfer cryptocurrencies (i.e; bitcoins) securely with no need for a centralized authority, using a publicly verifiable open ledger, referred to as *blockchain*. Consequently, blockchain technologies are widely adopted for data accounting and auditing features, thanks to their main intrinsic properties, namely providing tamper-proof evidences.

In this paper, we propose a new blockchain-based platform for data usage auditing while preserving users' privacy and ensuring continuous data availability. Our proposal relies on the use of the hierarchical ID-based cryptographic technique, where a central master authority delegates the process of public/private keys' generation to the different participating entities, based on authentic public elements.

Our solution has several advantages. First, relying on a blockchain infrastructure, we provide a trusted and transparent environment that permits service providers to have tamper-proof evidence of receiving the users' consent before processing their personal data. Second, our proposal ensures better confidentiality. That is, every data owner acts as a delegated PKG by computing an ID-based pair of keys to encrypt/sign the data that he intends to share with a service provider. As such, the data access is managed by the data owner. Third, by using a per smart contract ID-based key, we provide a flexible and privacy preserving sharing approach. Indeed, the distribution of decrypting keys between the client and the authorized service providers, does not reveal any information about the data owner's private information.

Paper organization – Section II reviews blockchain-based technology and discusses related work. Section III highlights security and functional concerns. Section IV details our proposed solution. Section V gives a security analysis of our proposal and section VI concludes the paper.

II. Blockchain-based Technology

A. Background

Bitcoin [1] appeared as an innovative technology enabling users to directly transfer cryptocurrencies in between with no intermediaries. It is considered as the first decentralized cryptocurrency transfer system. It relies on cryptographic proofs of work, digital signatures, and peer-to-peer networking to provide a distributed ledger containing transactions, and referred to as a *blockchain*. A blockchain is essentially a public ledger of transactions or events recorded and stored in chronologically connected blocks [2], [10]. The philosophy underlying the blockchain technology is that records are *shared by all network nodes, updated by miners, monitored by everyone and owned and controlled by no one*. Nodes are simple users' computers, while miners are nodes with exten-

[1] https://bitcoin.org/en/

sive computational resources that can be used for transaction validation purposes.

Two approaches, known as permissionless blockchains, have emerged to implement decentralized services and applications, namely Bitcoin-blockchain and Ethereum [2]. Additionally to functions already supported by the Bitcoin-blockchain, e.g. mining of the digital currencies and transaction management, Ethereum also provides a contract functionality known as *smart contract*. Very good expressive language provides smart contracts with a high degree of automation.

Transactions submitted to the Ethereum environment are organized into blocks and chained to each other based on a cryptographic hash function, initially relying on a pre-computed genesis block. Once a block is added to the blockchain, it cannot be modified or removed for two reasons: first, a block modification would lead to wrong verification of the chain of hash values, and second, the block modification would require intensive efforts to change every replicate of the blockchain supposed to be hosted on a large number of independent nodes.

Recently, permissioned blockchains are gaining an expanding interest across multiple industries. This concept appeared as a promoting solution for business applications of blockchain and distributed ledgers, in which participants do not necessarily have full trust on each, yet requiring some means of identification. Unlike permissionless blockchains, there exists a central entity that decides and grants the right to individual peers to participate in the read/write operations. The Hyperledger [3] project is a prominent initiative dedicated to bringing blockchain technologies to business. It provides a modular consensus protocol, such as Byzantine Fault Tolerance (DFT) algorithm, that ensures efficient scalability and performance complexities with thousands of transactions per second [11].

B. Related Work

The nature of the blockchain is particularly suitable for data accounting and auditing features. It has attracted interest of the research community due to its shared and fault-tolerance database. Indeed, several constructions have been introduced to ensure provenance tracking [8], [4], [7], [9].

Liang et al. [7] proposed a blockchain-based data provenance architecture for cloud applications. Their construction records operations' history as provenance data which are hashed into Merkle tree nodes. A list of hashes of provenance data forms a Merkle tree which are attached as a blockchain transaction. As such, it is possible to immutably prove the provenance of data exchanges. Although the proposed approach is novel, it does not cover the definition of advanced policies or contracts regulating the usage of collected data.

In [12], Zyskind et al. presented a personal data management system that combines blockchain, considered as an access control moderator, and off-blockchain storage solution. Designed as unique owners of their personal data, clients are aware of data collected about them by service providers and how they are used. However, the [12] proposal permits to only define simple permit/deny access policies through white/blacklisting. Afterwards, Linn et al. propose an application of the data auditing framework for health scenarios [8]. In their construction, the blockchain is also considered as an access moderator to control the access to outsourced shared data.

Afterwards, Ouaddah et al. proposed, in [9], a blockchain based access control framework for IoT applications, referred to as FairAccess. Their proposal relies on the blockchain-based bitcoin technology as an access moderator that permits to distribute authorization tokens, where each authorization token represents the data owner signature of the granted access right.

In [4], Anmin et al. introduced a privacy-aware blockchain-based auditing system for shared data in cloud applications. In order to mitigate the power abuse of single tracing authorities, [4] presents a threshold approach, where at least t entities have to collaborate to recover the identity of a malicious user, ensuring thus the non-frameability of users.

III. Security and Functional Requirements

To propose an efficient blockchain based data usage auditing solution, we have to take into consideration (i) the contract lifecycle, (ii) accountability granularity, (iii) required information to be stored in the contract, (iv) access patterns and (v) blockchain architecture w.r.t. different models of deployments. A detailed discussion about blockchain design directions is available in http://www-public.tem-tsp.eu/ lauren_m/Blockchain/BlockchainDesign.pdf.

Our blockchain-based data usage auditing solution has to consider the following security and functional properties.
– privacy preservation : preserving users' privacy is twofold. First, it is useful in a context where anonymity should be enforced to forbid any user's personal information leakages. Second, the privacy property is closely related to the unlinkability requirement, such that the proposed solution has to ensure unlinkability of the data owner identity between different service providers.
– confidentiality: our solution has to prevent unauthorized disclosure of both personal identifying information and shared data between data owners and service providers.
– auditability: it is important to enable auditing authorities to lead an investigation and obtain consistent proofs in case of non-compliant activities.
– data transparency : each data owner should have a transparent view over how data are collected, accessed and processed.

IV. A Blockchain-based Auditing Architecture

A. Overview

Our blockchain-based data usage auditing solution relies on three different entities, defined as follows: a data owner ($\mathcal{O}$), a service provider ($\mathcal{S}$) (acting as a data controller) and a service provider ($\mathcal{P}$) (acting as a data processor) [3].
$\mathcal{O}$ subscribing to a service provider $\mathcal{S}$, has to create a data usage contract. In this contract, the data owner specifies the usage policy for both data transferred by $\mathcal{O}$ to $\mathcal{S}$, as well as any other collected data by $\mathcal{S}$ during the interactions with $\mathcal{O}$. When

[2] https://www.ethereum.org/

[3] https://www.hyperledger.org/

processing $\mathcal{O}$'s data, $\mathcal{S}$ has to comply to the granted usage policy, as registered in the smart contract in the blockchain. In addition, $\mathcal{O}$ needs to identify the list of possible actions that might be performed on behalf of the smart contract, including the smart contract approval (cf. section IV-D). If afforded by the data usage policy in the smart contract, $\mathcal{C}$ might forward $\mathcal{O}$'s data to service provider $\mathcal{P}$, by pushing the forwarding information to the blockchain. $\mathcal{O}$ being notified by the event, then proceeds to the creation of a new smart contract with $\mathcal{P}$. Our solution relies on a consortium-blockchain infrastructure, such that anyone is able to access (i.e; read) the contract as well as associated transactions. As such, we propose to apply some cryptographic mechanisms in order to provide secrecy of exchanged data and preserve data owner's privacy.

For this purpose, we rely on the usage of hierarchical ID-based encryption (HIBE) and signature (HIBS) schemes [5], [1]. That is, our hierarchical IBC approach presents several advantages. First, each $\mathcal{O}$, considered as a blockchain entity, acts as a delegated Private Key Generator (PKG). $\mathcal{O}$ is then able to derive an ID-based pair of keys to encrypt/sign the data that he intends to share with $\mathcal{S}$ or $\mathcal{P}$. Note that data access is managed by $\mathcal{O}$ himself. Second, the usage of hierarchical ID-based schemes permits to benefit from the existence of a root PKG entity (i.e; a trusted central entity). Indeed, this root PKG is responsible for generating certified public system parameters IBC–PE, for the whole system's entities ($\mathcal{O}$, $\mathcal{P}$, $\mathcal{S}$) and to guarantee the authenticity of the provided entities' identities. In addition, contrary to the client-PKG classical approach, the hierarchical identity application makes the public system parameters common to any entities and not specific to the public identity of the data owner $\mathcal{O}$. This brings two advantages, as it enables $\mathcal{O}$ to get rid of the cumbersome tasks of generating and publishing its own public parameters, and it supports indistinguishability among entities, mostly during the signature verification processes. Third, by using a per smart-contract ID-based key, our proposal ensures indistinguishability among smart contracts at the blockchain level, and unlinkability of smart contracts to data owners, thus preventing any re-identification of data owners based on search operations over the blockchain.

B. Security and Design Assumptions

1) Security Assumptions: Our blockchain-based solution considers the following security assumptions:

– an off-blockchain secure communication : there is an established secure channel between $\mathcal{O}$ and the service provider $\mathcal{P}$ or $\mathcal{S}$. It permits $\mathcal{O}$ to securely send its personal information along with data that he intends to share with the service provider. This secure channel supports mutual authentication and data confidentiality and integrity. For strong privacy preserving guarantees, the authentication procedure may rely on attribute based credential mechanisms, such as Idemix [4] and UProve [5].

– a robust blockchain and smart contract implementation: we assume that all blockchain-specific operations, such as mining processes and transaction anchoring activities are secure and not corruptible, and blockchain provides non-tamper proofs of data processing and managing events.

– a trusted PKG entity: the PKG has a fundamental role, as it supports the enrollment of the different entities $(\mathcal{O}, \mathcal{S}, \mathcal{P})$ into the system, the derivation of their respective private keys and the generation and publication of security ID-based public elements for the full ID-based system.

2) Smart Contract Design Assumptions: For our solution, each data contract involves two parts: *data structure*, denoted by p_1 and *data usage policy*, denoted by p_2. The *data structure* part, contains types and instances of exchanged data between the data owner $\mathcal{O}$ and the service provider $\mathcal{S}$. The *data usage policy* part specifies all authorized and non-authorized usage actions for both transferred and implicitly collected data. Data usage actions may include data storage duration afforded to the service provider, type of processing granted over the owner's data (i.e; execution of any purpose specific activity), authorization to transfer data to other service providers, generation of derived data based on exchanged personal data. In addition, the data owner needs to specify different actions that are authorized by the smart contract from both $\mathcal{O}$ and service providers ($\mathcal{S}$ and $\mathcal{P}$). Data owner's actions include deletion of the smart contract, in-activation of the contract and modification of new data usage policies, while service providers' actions include approval of the smart contract content.

C. System Initialization

We consider that a trusted root PKG entity is generating and publishing the ID-based public elements increased with the ID-based Signature and Encryption scheme (cf. section IV-B1). Moreover, each entity $\mathcal{E}$ holds a pair of public and private keys $(sk_{\mathcal{E}}, pk_{\mathcal{E}})$, where $\mathcal{E} \in \{\mathcal{O}, \mathcal{S}, \mathcal{P}\}$. The pair of private and public keys $(sk_{\mathcal{E}},\ pk_{\mathcal{E}})$ is mathematically generated by the root PKG according to the selected ID-based cryptographic schemes. The private key $sk_{\mathcal{E}}$ is then securely transferred by the root PKG to the entity $\mathcal{E}$, via a secure channel. That is, $sk_{\mathcal{E}}$ is a secret of $\mathcal{E}$ associated to the public key $pk_{\mathcal{E}}$ which is derived from the unique entity identity of $\mathcal{E}$ (i.e; one of $\mathcal{E}$'s blockchain addresses), based on the PKG's own secret and the public elements IBC–PE. As such, the pair $(pk_{\mathcal{E}},\ sk_{\mathcal{E}})$ is somewhat certified by the root PKG, as signature generation and data decryption operations require an entity to be equipped with his own secret $sk_{\mathcal{E}}$.

D. Smart Contract Creation

First, $\mathcal{O}$ establishes a direct secure channel with $\mathcal{S}$ where $\mathcal{S}$ and $\mathcal{O}$ authenticate each other thanks to their respective $(sk_{\mathcal{E}}$ and $pk_{\mathcal{E}})$ pair of keys. The transaction is identified with an T_{id} parameter. Through this channel, $\mathcal{S}$ might issue a data usage request to $\mathcal{O}$, thus leading $\mathcal{O}$ to derive some parameters for the smart contract $\mathcal{C}$ before creating the smart contract into the blockchain.

Indeed, after the initialization phase, $\mathcal{O}$ is provided with public

[4] https://www.zurich.ibm.com/identity_mixer/

[5] https://www.microsoft.com/en-us/research/project/u-prove/

and private keys $(sk_{\mathcal{O}}, pk_{\mathcal{O}})$ and is then able to derive a per smart-contract $ID_{\mathcal{C}}$, and its corresponding pair of public and private keys $(sk_{\mathcal{C}}, pk_{\mathcal{C}})$. Based on the general public elements IBC–PE, $\mathcal{O}$ first creates an identifier $ID_{\mathcal{C}}$, that forms the hierarchical smart contract identity that depends on both the data owner identity $ID_{\mathcal{O}}$ as well as the smart contract identifier $ID_{\mathcal{C}}$. Then, $\mathcal{O}$ relies on the keygen to derive a per smart contract ID-based public key $pk_{\mathcal{C}}$ which is generated from the concatenation of the data owner's address $ID_{\mathcal{O}}$, the service provider address $ID_{\mathcal{S}}$ and the smart contract identifier $ID_{\mathcal{C}}$, and based on the general public elements IBC–PE, such that $pk_{\mathcal{C}} = Hash_{pub}(ID_{\mathcal{S}}||ID_{\mathcal{C}})$.

Note that the secret key $sk_{\mathcal{C}}$ associated with the public key $pk_{\mathcal{C}}$ is only known by the data owner, at the moment of the smart contract creation.

After approving the data usage request, $\mathcal{O}$ builds the data section part ($\mathtt{p}_2$) of the contract. To avoid publishing identifying personal information, we obfuscate data values in $\mathtt{p}_1$ by using a public hash function. To strengthen the privacy and prevent linkability attacks, data values are concatenated with the smart contract secret key $sk_{\mathcal{C}}$ before applying the hash function.

Then, $\mathcal{O}$ generates a smart contract creation transaction, which will be anchored in the blockchain. As such, we define a smart contract creation transaction T as the tuple $T = (T_{id}, ID_{\mathcal{C}}, U_{\mathcal{C}}, enc_{pk_{\mathcal{S}}}(sk_{\mathcal{C}}), \sigma_{\mathcal{C}})$, where T_{id} is the transaction identifier, $ID_{\mathcal{C}}$ is the smart contract identifier, $U_{\mathcal{C}}$ is a smart contract approval request identifier towards $\mathcal{S}$, $enc_{pk_{\mathcal{S}}}(sk_{\mathcal{C}})$ is the encrypted private key $sk_{\mathcal{C}}$, based on the public key of the service provider $\mathcal{S}$ and $\sigma_{\mathcal{C}}$ is the signature of $\mathcal{O}$ of all the previous fields, using $sk_{\mathcal{C}}$, actually only known to $\mathcal{O}$.

Upon receiving the smart contract creation request notification, the service provider proceeds in a classical fashion. Based on the public system parameters IBC–PE, $\mathcal{S}$ starts decryption the enciphered private key $sk_{\mathcal{C}}$, relying on his own private key $sk_{\mathcal{S}}$. Then, $\mathcal{S}$ checks the corresponding data owner signature, using the public key $pk_{\mathcal{C}}$. If both operations are successful, then $\mathcal{S}$ approves the transaction associated with the smart contract creation thanks to its own key $sk_{\mathcal{S}}$.

After approval, $\mathcal{O}$ sends his personal data via a secure channel to $\mathcal{S}$, with respect to the hashed values embedded in $\mathtt{p}_1$ thus certifying his consent agreement. As such, using the deciphered secret key $sk_{\mathcal{C}}$ and the received data values, $\mathcal{S}$ may check the integrity of acknowledged owner's data. Note that additionally to the approval of the smart contract by $\mathcal{S}$, other actions have to be included into the smart contract, including: (i) deleting the contract from the blockchain by $\mathcal{O}$ as $\mathcal{O}$ must be able to withdraw his consent. While making inactive the contract, subsequent data usage should not be possible. However, the contract history should be always registered in the blockchain, and (ii) changing data usage where $\mathcal{O}$ must be able to add some restrictions to his data usage.

E. Data Usage Transfer

In case a forward action is authorized on some $\mathcal{O}$'s data by the data usage policy specified in the smart contract with $\mathcal{S}$, then data might be transferred by the service provider $\mathcal{S}$ to the corresponding data processor $\mathcal{P}$ and a transaction has to be pushed to the blockchain. For this purpose, $\mathcal{S}$ has to generate a data usage transaction. As such, the data owner is notified about the service provider data transfer activity. We define a data usage transaction T corresponding to a data transfer activity as the tuple $T = (T_{id}, ID_{\mathcal{C}}, ID_{\mathcal{S}}, U_{\mathcal{C}}, enc_{pk_{\mathcal{O}}}(ID_{\mathcal{P}}), \sigma_{\mathcal{S}})$, where T_{id} is the transaction identifier, $ID_{\mathcal{C}}$ is the smart contract identifier, $ID_{\mathcal{S}}$ is the service provider identity, $U_{\mathcal{C}}$ is a transfer data event identifier, $enc_{pk_{\mathcal{O}}}(ID_{\mathcal{P}})$ is the encrypted identity of $\mathcal{P}$, based on the public key of $\mathcal{O}$ and $\sigma_{\mathcal{S}}$ is the signature of $\mathcal{S}$ over all the previous fields.

Upon receiving the data transfer activity notification, the data owner $\mathcal{O}$ creates a new contract identifier with $\mathcal{P}$, following the same approach presented in section IV-D. Note that $\mathcal{P}$ does not know the new contract identifier between $\mathcal{O}$ and $\mathcal{S}$.

F. Auditing

The auditing process is performed either in a private fashion by $\mathcal{O}$ or by a dedicated auditing authority; or publicly by anyone. First, public auditing relies on blockchain transactions' information. Considered as a tamper-proof architecture, a public verifier may detect non-compliant activities associated to a specific smart contract data usage policy. That is, each anchored transaction carried with respect to a selected smart contract has to include a contract identifier and a data usage activity identifier which is signed by the service provider. Second, private auditing relies on public blockchain information, as well as private data, provided by the claimer. As such, when $\mathcal{O}$ requests a private audit for a misuse of his personal data, he shares the private and public keys associated to concerned smart contracts $(pk_{\mathcal{C}_i}, sk_{\mathcal{C}_i})_{\{i \in [1,N]\}}$, where N is the number of smart contracts. As such, the auditing authority is able to lead an investigation, while crawling blockchain transactions corresponding to the provided smart contracts' identifiers. We note that private auditing has to be paid by the audited service provider, if non-compliant activities are reported.

V. Security Analysis

For designing a secure blokchain-based auditing scheme, we consider realistic threat models. That is, an attacker is able to read, send and drop a transaction addressed to the blockchain. The attacker targets data owners, service providers as well as the blockchain, as follows:

– based on previous data usage sessions, as well as provided blockchain data, an attacker tries to impersonate a data owner to afford a honest service provider some rights to be logged into the blockchain without the legal data owner's consent.

– an attacker tries an attack against the privacy property w.r.t. data owners, while trying to link smart contract identifiers or a smart contract to a specific owner.

– an attacker attempts to prevent the publication of a legitimate transaction. For example, in order to prevent data transfer notification to the data owner, an attacker may try a DoS attack against a data usage event, or attempt a flooding attack on the blockchain with invalid data usage information.

Confidentiality — Our proposed solution is resistant against data secrecy attacks for several reasons here-below listed:
– only hashed data values and enciphered information published in the smart contract: the client is in charge of hashing his personal data for fulfilling p_1 of the smart contract and enciphering a secret with $pk_{\mathcal{S}}$ addressed to server $\mathcal{S}$.
– the enciphering key only known by a pair of owner and service provider : the owner acting as a delegated PKG, is the only entity knowing its secret $sk_{\mathcal{C}}$, and as such is the only entity able to generate the pair of public and private keys $(pk_{\mathcal{C}}, sk_{\mathcal{C}})$ associated with a specific smart contract. After secure transmission of it to the provider, $sk_{\mathcal{C}}$ is finally only known to the owner and the involved service provider.
– one per smart-contract enciphering key: the owner generates a specific pair of keys per smart contract. As such, even in case the private key $sk_{\mathcal{C}}$ is compromised, the attacker cannot get any information about the other per smart-contract keys. Thus, the attacker cannot learn any significant information from the blockchain, because only hashed data values or encrypted information are registered in the public ledger.

Privacy — The privacy property is ensured thanks to the following technical features. First, our approach considers one smart contract per service provider, such that the owner generates a specific identifier $ID_{\mathcal{C}}$ and a secret key $sk_{\mathcal{C}}$ for each of its smart contracts with regard to one specific service provider. The secret key is used by the owner to sign the contract creation. As such the owner can not be identified thanks to the value of $ID_{\mathcal{C}}$, and smart contracts can not be linked to the same owner based on their $ID_{\mathcal{C}}$. Second, within a smart contract, data values are concatenated with the smart contract secret key $sk_{\mathcal{C}}$ before hashing. This prevents from any search over the blockchain database for same data values in order to link smart contracts in between.

Auditability — Our approach ensures the auditibility requirement as follows:
– transparent usage : our approach is based on a consortium blockchain infrastructure, that permits public access (i.e; read privilege) the contract and its associated transactions, to anyone. Thus, it provides a transparent view over how data are collected and accessed.
– signed transactions : our approach relies on signed transactions. That is, both smart contract creation and data usage transactions have to be signed by the data owner $\mathcal{O}$ and the service provider $\mathcal{S}$ and $\mathcal{P}$ respectively. Signed transactions ensure that each activity has been efficiently performed by the holder of the used private key, which is certified by the PKG entity. As such, the resistance of the chosen HIBS scheme against forgery attacks has a direct impact on the fulfillment of the auditibility requirement.
– approval of smart contracts creation – each smart contract creation by the data owner $\mathcal{O}$ has to be approved by the service provider. That is, the smart contract approval requires holding the secret key of the service provider $sk_{\mathcal{S}}$, which is needed to decrypt the secret key of the smart contract $sk_{\mathcal{C}}$, thus certifying the ability for further decryption of shared data.

Availability — As a highly decentralized infrastructure, the blockchain technology helps in terms of availability. It becomes possible to provide liveness guarantees of data usage. In addition, we note that an attack, designed to prevent a valid transaction from being registered in the blockchain, is equivalent to the double spending problem in bitcoin architectures [6]. Indeed, the adversary has to have the control on more than a half of the blockchain nodes, which is assumed to be hard.

VI. Conclusion

In this paper, we propose a new blockchain-based solution for data usage auditing relying on the use of hierarchical ID-based mechanisms. Acting as a delegated PKG, each data owner is able to provide consent on his data usage and to control data collection and processing activities, in a privacy preserving manner based on a per smart-contract approach. In addition, our solution enables service providers to have a proof of receiving the data owner's consent before processing his personal data, as the blockchain architecture is considered to be computationally tamper-proof. Based on a consortium blockchain infrastructure, our proposal provides a regulatory framework to properly enforce laws. For instance, in case of a non-compliance activity (i.e; unauthorized data access) reported by a data owner, authorized authorities may lead an investigation referring to registered blockchain transactions with respect to concerned entities.

References

[1] O. Blazy, E. Kiltz, and J. Pan. (hierarchical) identity-based encryption from affine message authentication. In *International Cryptology Conference*. Springer, 2014.

[2] M. Crosby, P. Pattanayak, S. Verma, and V. Kalyanaraman. Blockchain technology: Beyond bitcoin. *Applied Innovation*, 2:6–10, 2016.

[3] C. Europe. Proposal for a regulation of the european parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data. In *General Data Protection Regulation*, January 2016, 2016.

[4] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang. Npp: A new privacy-aware public auditing scheme for cloud data sharing with group users. *IEEE Transactions on Big Data*, 2017.

[5] C. Gentry and S. Halevi. Hierarchical identity based encryption with polynomially many levels. In *Theory of Cryptography Conference*, pages 437–456. Springer, 2009.

[6] G. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Capkun. Misbehavior in bitcoin: A study of double-spending and accountability. *ACM Transactions on Information and System Security*, 2015.

[7] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla. Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2017.

[8] L. Linn and M. Koo. Blockchain for health data and its potential use in health it and health care related research, 2016.

[9] A. Ouaddah, A. Abou Elkalam, and A. Ait Ouahman. Fairaccess: a new blockchain-based access control framework for the internet of things. *Security and Communication Networks*, 9(18), 2016.

[10] M. Swan. *Blockchain: Blueprint for a new economy*. O'Reilly Media, Inc., 2015.

[11] M. Vukolic. Rethinking permissioned blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, BCC '17, New York, NY, USA, 2017. ACM.

[12] G. Zyskind, O. Nathan, and A. Pantland. Decentralizing privacy: Using blockchain to protect personal data. In *IEEE Security and Privacy Workshops (SPW)*, 2015.