BDUA: blockchain-based data usage auditing

Nesrine Kaaniche, Maryline Laurent

BDUA : Blockchain-based Data Usage Auditing

Nesrine Kaaniche
SAMOVAR, Telecom SudParis,
University Paris-Saclay, France

Maryline Laurent
SAMOVAR, Telecom SudParis,
University Paris-Saclay, France

***Abstract*—Personal data are often collected and processed in a decentralized fashion, within different contexts. For instance, with the emergence of distributed applications, several providers are used to correlate their records, to provide personalized services to their clients. As such, to protect users' privacy, different pseudonyms are generally used for different contexts. These pseudonyms have to be unlinkable to prevent identifying records to be associated to the same user. Although unlinkable, these pseudonyms have to be processed and exchanged according to their owners' consent and in a privacy-preserving fashion. In this paper, we propose $\mathcal{BDUA}$, a new Blockchain-based Data Usage Auditing system, that ensures a controlled yet privacy preserving exchange of distributed data, such that a set of authorized auditing entities are able to conduct an accurate auditing relying on registered blockchains' transactions.**

***Keywords*-pseudonymous schemes, homomorphic encryption, blockchain, privacy, auditing.**

I. Introduction

Several organizations often collect large amounts of sensitive data about their clients. This raises the question of the transparency of usage and protection of the collected data. Indeed, when data collection and processing are produced in a distributed fashion, it is commonly known that different data records associated to the same user can be easily linked. Several countries, such as US, Belgium, Denmark and Sweden rely on a unique identifier, referred to as *national social security number* for preserving linking, for example government datasets and health records.

While such unique nation-wide identifiers across the whole system allow to easily correlate records with no need for data owners' consent, they introduce several security and privacy issues. That is, a number of entities are made powerful over citizens with the capability to trace them, combine their datasets, infer some information... Even worse, any data breach reveals fully identifiable and linkable personal information. As such, several approaches, called pseudonym systems, have been proposed. In fact, they rely on the use of different context-specific pseudonyms that are unlinkable. In order to enable associating different outsourced data sets, a central entity, referred to as a *convertor*, can later be asked to associate pseudonyms to a specific user on a case per case basis.
A pseudonym system is an interesting solution that permits to guarantee auditability and ensure confidentiality of data w.r.t. involved entities. However, as the convertor is the main central entity, he can easily correlate data with their respective users, while tracing users' activities, based on data exchanges. Recently, Camenish and Lehmann [1] presented a blind-convertor system, where the conversion process, executed by the convertor, is performed in a privacy-preserving (i.e; anonymous and unlinkable) fashion, with no need for learning the pseudonyms or the identity of the concerned user. The authors extended their work by presenting in [2] a system where user-centric audits are set-up by an oblivious convertor, which is still the main central entity. This appended auditability feature is a very important property, for compliance with legal and business requirements. In 2016, the European Union (EU) adopted a new General Data Protection Regulation (GDPR) that aims at effectively ensuring the protection of users' privacy [3].

Contributions — While several systems permit to satisfy transparency and auditability requirements based on a user-centric fashion, we propose $\mathcal{BDUA}$ a new Blockchain-based Data Usage Auditing system, that ensures a controlled yet privacy preserving exchange of distributed data, such that a set of authorized auditing entities are able to conduct an accurate auditing relying on registered blockchains' transactions. Our $\mathcal{BDUA}$ scheme ensures the following properties:

- an unlinkable pseudonym system where a selected convertor obviously collaborates with each user to derive individual pseudonyms for each server. That is, the convertor cannot learn the derived pseudonyms, but is still the only entity that can link pseudonyms together without learning the particular user or pseudonym for which such translation is requested.
- an accurate privacy-preserving auditing process based on blockchain transactions and achieved thanks to transactions being registered by each of the involved participating entity, and enciphering of linking information with respect to a multi-level attribute based encryption.

Paper Organization — Section II introduces the blockchain technology and reviews the related work. Section III presents the problem statement and highlights the security requirements. Section IV provides a general overview of the proposed schemes, with the different building blocks and details $\mathcal{BDUA}$ procedures and algorithms. Section V introduces the threat model, discusses the security properties and gives a brief performances analysis, before concluding in section VI.

II. Blockchain Technology

This section reviews the blockchain technologies (cf. section II-A) and discusses related works (cf. section II-B).

A. Background

Blockchain has gained an attractive interest, since 2008, with the bankruptcy of *Lehman Brothers* in the United States. A blockchain is essentially a public ledger of transactions or events recorded and stored in chronologically connected blocks [4]. 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* [4]. Nodes are simple users' computers or mobile devices, while miners are nodes with extensive computational resources that can be used for transaction validation purposes. Two approaches, known as permissionless blockchains, have emerged to implement decentralized services. The first approach relies on the existing Bitcoin blockchain[1]. The main advantage of this approach is that the Bitcoin blockchain already exists and is adopted by many users, which makes it more secure, transparent and resilient. However, blocks are mined every 10 minutes and the scripting language is not Turing-complete [4]. The second approach is to build an alternative blockchain with all the desired features, which promises full decentralization, such as Ethereum[2]. Additionally to functions already supported by other public blockchain-based platforms, Ethereum also provides a contract functionality known as *smart contract*, ensuring a high degree of automation.
Recently, permissioned blockchains are gaining an expanding interest across multiple industries. This concept appeared as a promoting solution for business applications of 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.

B. Blockchain-based Data Auditing- 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 [5]–[8].

In [7], Zyskind et al. presented a personal data management system that combines blockchain, considered as an access control moderator, and off-blockchain storage solutions. That is, clients are aware of data collected about them by service providers and how they are used. When a client subscribes to a service provider, one transaction is created defining the set of access policies and another contains the hashes of data stored in an off-chain database. However, [7] permits to only define simple permit/deny access policies through white/blacklisting. Based on [7], Linn et al. propose an application of the data auditing framework for health scenarios [5]. In their construction, the blockchain is also considered as an access moderator to control the access to outsourced shared data.

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

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

Recently, Neisse et al. discussed design requirements of blockchain-based solutions for data provenance tracking [9] and presented an evaluation of their implementation results, in order to give a comprehensive overview of different defined approaches. Later, in [8], Kaaniche and Laurent presented a blockchain-based platform for data usage auditing while preserving users' privacy. The proposed construction relies on the use of hierarchical ID-based techniques, where a central master authority delegates the process of public/private keys' generation to the different participating entities.

III. Motivation and Security Requirements

According to the GDPR [3], the data subject's consent is given for specific purposes to which both the data controller and the data processor have to comply. In this context, three main roles are defined. The *data subject* gives his consent to a *data controller* (i.e. organization, enterprise) for processing his personal data, and optionally forwarding them to a *data processor* (i.e. organization, enterprise) that is required to process data on behalf of the data controller. Data controllers are responsible for (i) specifying to the data subject the purpose of data collection, (ii) obtaining the data subject's consent and (iii) processing personal data according to the consented purposes, and not beyond.

From a data controller or processor's perspective[3], there is a need for a trusted and transparent accountability solution that enables them to get a proof of the data owner's consent prior to processing his personal data. From a data owner's perspective, there is a need for new security mechanisms that support data accountability and provenance auditing when his personal data were forwarded to data processors. Indeed, it is important to conceive a secure and transparent solution that permits data owners to (i) check that data controllers and processors are correctly using their data with respect to the consented purposes, (ii) verify whether data were forwarded without their consent and (iii) withdraw their consent.

Several works have been proposed, mainly based on cryptographic techniques to enable service providers to inform users about the actual data processing that takes place on their personal data. In [10], Wouters et al. proposed a scheme for building a logging-chain of processes for eGovernment services. A data owner can reconstruct the trail of such a process and verify its status if he is the subject of that process. Reconstruction is based on hand-overs (ie; special types of log events) that link data stored by multiple logging servers. The proposal [10] is privacy-preserving in the sense that only data owners can link their related log entries. Afterwards, Pulls et al. presented a privacy-preserving auditing scheme [11]. The proposed mechanism supports dynamic and distributed processes' logging, such that the log entries are world-readable and distributed among several logging servers. However, authors assume that data processors need to be trusted.

[3]In the following, the remainder of the paper refers to the data subject as the data owner and to both the data controller and the data processor as the service provider.

In [12], Galindo and Verheul proposed a data exchange solution based on a convertor. This latter could derive and distribute pseudonyms, however these identifiers could be easily linked. In the same vein, Camenish and Lehman proposed a privacy preserving data exchange scheme, based on a blind convertor [1]. As such, their proposal is based on pseudonyms which are per se unlinkable but can be transformed from one server to another with the help of the convertor. However, [1] does not support auditing capabilities. Afterwards, authors extended the main construction to support user auditing [2]. The proposed scheme relies on an oblivious central entity that provides a system where user-centric audits logs are created by the convertor. [2] makes use of several new building blocks and the tag-chaining approach for ensuring user-audits.

Table I presents a comparison between $\mathcal{BDUA}$ and most closely related schemes, w.r.t. functional requirements.

TABLE I
COMPARISON BETWEEN $\mathcal{BDUA}$ AND MOST CLOSELY RELATED SCHEMES

Scheme	blind conversion	multi-convertor	user auditing	public auditing
[10]	×	×	✓	✓
[11]	×	×	✓	✓
[1]	✓	×	×	×
[2]	✓	×	✓	×
$\mathcal{BDUA}$	✓	✓	✓	✓

✓ and × indicate that the requirement is achieved or not, respectively.

IV. $\mathcal{BDUA}$: A NEW BLOCKCHAIN-BASED DATA USAGE AUDITING SCHEME

In the section, we first present an overview of $\mathcal{BDUA}$ (cf. section IV-A) and introduce the notations used in the paper in section IV-B. Then, we review the different building blocks in section IV-C. Afterwards, we detail our construction IV-D.

A. Overview

$\mathcal{BDUA}$ is a new blockchain-based data usage auditing scheme that deals with the challenge of enabling privacy preserving yet transparent and controlled data usage in a decentralized system, while considering a convertor that permits to ensure pseudonyms matching. It relies on Camenish et al. [1], [2] construction, which is adapted and mainly extended to support public auditing by authorized authorities. Indeed, each process is registered in the blockchain, via a set of transactions pushed by the participating entities. Each transaction contains the encrypted identifier of the previous transaction referring to the latest process performed on the user's data. In order to enable public authorized auditing, $\mathcal{BDUA}$ relies on a multi-level encryption scheme [13], such that deciphering entities have access to different levels of detailed information with respect to their certified attributes.

As depicted in Figure 1, $\mathcal{BDUA}$ is based on four different entities, namely user $\mathcal{U}$, convertor $\mathcal{X}$, server $\mathcal{S}$ and Blockchain BC. Conceptually, each user $\mathcal{U}$ blindly creates with the convertor $\mathcal{X}$ an individual pseudonym for each server $\mathcal{S}$, derived from his unique main identifier $id_{\mathcal{U}}$. It is to be noted that the convertor does not learn anything about the derived pseudonyms. The creation of a new user pseudonym

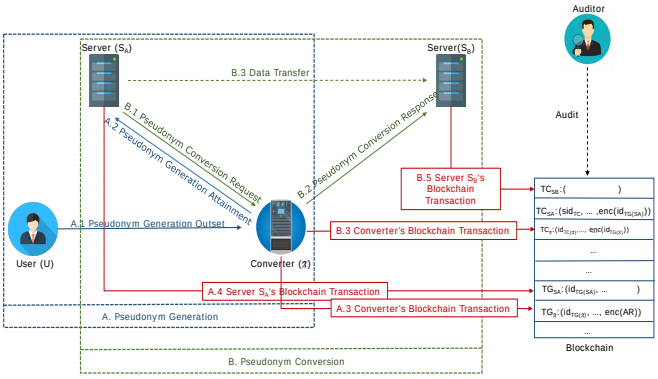

Fig. 1. $\mathcal{BDUA}$ Network Model

with a server, is then registered by $\mathcal{X}$ in the blockchain. The pseudonym creation transaction has to specify the set of usage actions allowed on shared data between the user and the server. These granted privileges as well as participating entities are then encrypted relying on the ABE scheme, while the associated ABE access policy should allow auditing authorities to access to data when needed.

The core of the pseudonym conversion protocol is similar to Camenish et al. scheme [1]. When a server $\mathcal{S}_A$ wants to convert the pseudonym $\mathsf{nym}_{\mathcal{U},\mathcal{S}_A}$ of a user $\mathcal{U}$ towards a server $\mathcal{S}_B$, it sends the pseudonym in a homomorphically encrypted form to the convertor. The convertor blindly computes $\mathsf{nym}_{\mathcal{U},\mathcal{S}_B}$. In $\mathcal{BDUA}$, the conversion request and response should be registered in the Blockchain by both servers $\mathcal{S}_A$ and $\mathcal{S}_B$ as well as the related convertor. These transactions should contain the same session identifier selected by the convertor and accompanied by a freshly generated random that is encrypted by both an ABE algorithm for auditing authorities as well as a dynamic (i.e; randomized) public key of the corresponding user (yet unknown) for user-auditability.

The encrypted audit logs are made publicly available via the Blockchain such that they can be fetched by all users. However, only authorized auditing authorities, referred to as $\mathcal{D}$ and data owners can respectively decrypt related transactions.

B. Notations

Each algorithm is denoted as: $out = \mathsf{f}_{\mathcal{E}}(ent)$, where out is the output of the algorithm $\mathsf{f}_{\mathcal{E}}$, executed by $\mathcal{E}$ based on the set of inputs ent, where $\mathcal{E}$ represents any system's entity, such that $\mathcal{E} = \{\mathcal{U}, \mathcal{X}, \mathcal{S}, \mathcal{D}\}$.

When the output of $\mathsf{f}_{\mathcal{E}}$ is exchanged between two system's entities, it is denoted by $out_{\mathcal{E}_s,\mathcal{E}_r}$, where $\mathcal{E}_s$ presents the sending entity and $\mathcal{E}_r$ the receiving entity.

If f is a two party oblivious algorithm, it is denoted by $out_{\mathcal{E}_r} = \mathsf{f}_{\mathcal{E}_r,\mathcal{E}_s}(ent)$, where $\mathcal{E}_r$ and $\mathcal{E}_s$ present the participating entities based on the set of inputs ent and $out_{\mathcal{E}_r}$ is the output of the algorithm retrieved by $\mathcal{E}_r$.

C. Building Blocks

1) Bilinear Maps: Let $\mathbb{G}$, $\tilde{\mathbb{G}}$ and $\mathbb{G}_t$ be three groups of prime order q, where g and $\tilde{g}$ are generators of $\mathbb{G}$ and $\tilde{\mathbb{G}}$

respectively. A function $\hat{e}$ is considered as a bilinear map if it satisfies the following requirements:

- bilinearity: $\hat{e}(g^a, \tilde{g}^b) = \hat{e}(g, \tilde{g})^{ab}$, for all $a, b \in \mathbb{Z}_q$;
- non-degeneracy : for all generators $g \in \mathbb{G}$ and $\tilde{g} \in \tilde{\mathbb{G}}$, $\hat{e}(g, \tilde{g})$ generates $\mathbb{G}_t$;
- efficiency: there exists an efficient algorithm that computes $\hat{e}(x, \tilde{y})$, for all $x \in \mathbb{G}$ and $y \in \tilde{\mathbb{G}}$.

2) Oblivious Pseudo Random Function: An oblivious Pseudo-Random Function (OPRF) [14] is a two party protocol between a sender $\mathcal{E}_s$ and a receiver $\mathcal{E}_r$ for securely computing a pseudorandom function $f(k, x)$ relying on the key k computed by $\mathcal{E}_s$ and on input x contributed by $\mathcal{E}_r$, such that $\mathcal{E}_r$ learns only the value $f(k, x)$ while $\mathcal{E}_s$ learns nothing from the interaction. In 2005, Dodis and Yampolskiy [15] presented an OPRF construction that is proven secure in the standard model based on the decisional Diffie-Hellman inversion assumption. $\mathcal{BDUA}$ relies on an OPRF to derive the user core identifier, such that each user $\mathcal{U}$ performs with a convertor $\mathcal{X}$ an oblivious process, with inputs $id_{\mathcal{U}}$ and $sk_{\mathcal{X}}$, respectively. As such, $\mathcal{U}$ receives his own core identifier, denoted by $c_{id} = g^{\frac{1}{sk_{\mathcal{X}} + id_{\mathcal{U}}}}$.

3) Semi-Homomorphic Encryption Scheme: An encryption scheme $(\mathsf{gen}, \mathsf{enc}, \mathsf{dec})$ is said to be *homomorphic*, if it has an efficient operation $\odot$ on ciphertexts such that, if for $(pk, sk) \xleftarrow{\$} \mathsf{gen}(1^\tau)$, $C_1 = \mathsf{enc}(pk, m_1)$ and $C_2 = \mathsf{enc}(pk, m_2)$, then $C_1 \odot C_2 = \mathsf{enc}(pk, m_1 \cdot m_2)$.
ElGamal Encryption Scheme — is homomorphic, chosen plaintext secure (CPA) under a group $(\mathbb{G}, g, p)$ such as:

- HE.$\mathsf{gen}(1^\tau)$ – it takes as input the security parameter τ, selects at random $x \in \mathbb{Z}_p$, computes $y \leftarrow g^x$ and outputs $(sk, pk) = (x, y)$.
- HE.$\mathsf{enc}(pk, m)$ – it takes as input a message $m \in \mathbb{G}$ and the public key pk, selects at random $r \in \mathbb{Z}_p$ and outputs $C = (C_1, C_2) = (pk^r, g^r m)$.
- HE.$\mathsf{dec}(sk, C)$ – it takes as input the ciphertext C and the secret key sk. It outputs $m \leftarrow C_2 \cdot {C_1}^{-1/sk}$.

4) Re-randomizable Public Keys: In [16], Waters et al. introduced the notion of *incomparable keys* for receiver anonymity. The holder of a secret key must be able to generate a large number of public encryption keys such that any message encrypted with any of these public keys can be decrypted by one secret key. Public keys must have the property that they cannot be tested for equivalence without access to the corresponding secret key(s). Waters et al. consider that derivation of several incomparable public keys based on the knowledge of a secret key, instead of a re-randomization based on the unique knowledge of the public key. Recently, Camenish et al. proposed a re-randomizable public key scheme, based on ElGamal instantiation, called *R-ElGamal*, defined as:

- $\mathsf{rkgen}(1^\tau)$ – it takes as input the security parameter τ, selects at random $x \in \mathbb{Z}_p$ and $r \in \mathbb{Z}_p$, and computes $y_1 \leftarrow g^{xr}$ and $y_2 \leftarrow g^x$. It outputs $(sk, pk) = (x, (y_1, y_2))$.
- $\mathsf{rand}(pk)$ – it takes as input the public key pk, selects at random $r' \in \mathbb{Z}_p$ and outputs $(spk' \leftarrow ({y_1}^{r'}, {y_2}^{r'})$.
- $\mathsf{renc}(pk, m)$ – it takes as input a message $m \in \mathbb{G}$ and the public key pk, selects at random $r \in \mathbb{Z}_p$ and outputs $C = (C_1, C_2) = ({y_1}^r, {y_2}^r m)$.
- $\mathsf{rdec}(sk, C)$ – it takes as input the ciphertext C and the secret key sk. It outputs $m \leftarrow C_2 \cdot {C_1}^{-1/sk}$.

5) Multi-Level Attribute based Encryption: A multi-level encryption scheme based on attribute based mechanisms ensures a selective access to data based on users' granted privileges. Practically, when a party encrypts a data file, she specifies an access structure and a certain number of security levels. Thus, a user is able to decrypt a sub-set of data blocks related to a security level k if that user's private keys satisfy the sub-set of attributes related to the k-security level [13]. Note that that several elementary functions may be used for several encryption sessions (i.e., computation of polynomials and encryption of leaves' nodes). As such, only a few number of exponentiations and multiplication is required by each user. In addition, the decryption process may be partially delegated to a semi-trusted entity, for better performances [17]

D. $\mathcal{BDUA}$ Procedures

Our $\mathcal{BDUA}$ scheme relies on five different procedures, namely: *system initialization*, *pseudonym generation*, *pseudonym conversion*, *granted privileges update*, *user and authorities' audit*, that are detailed hereafter.

1) System Initialization: During this phase, the convertor $\mathcal{X}$ and the set of related servers $\mathbb{S}$ generate their respective keys, such that each convertor $\mathcal{X}$ derives a pair of public and private keys $(pk_{\mathcal{X}}, sk_{\mathcal{X}})$, as well as a secret $x_{\mathcal{S}_i}$ associated to each server $\mathcal{S}_i$, where $i \in [1, |\mathbb{S}|]$. Similarly, each user $\mathcal{U}$ and server $\mathcal{S}_{i\{i \in [1, |\mathbb{S}|]\}}$ generate a pair of private and public keys defined as $(pk_{\mathcal{U}}, sk_{\mathcal{U}})$ and $(pk_{\mathcal{S}_i}, sk_{\mathcal{S}_i})$ respectively.

Remark 1: **consistency of exchanged messages—** $\mathcal{BDUA}$ relies on exchanging enciphered messages between the participating entities. Hence, for the sake of better security guarantees, $\mathcal{BDUA}$ may encompass non-interactive zero-knowledge proofs of knowledge (PoK) of encrypted messages [1]. That is, each encrypted message has to be followed by a PoK, such as the receiving entity may check the consistency of the received encrypted message.

2) Pseudonym Generation: The pseudonym generation process involves three entities: user $\mathcal{U}$, convertor $\mathcal{X}$ and corresponding server $\mathcal{S}$. They jointly and blindly compute a pseudonym $nym_{\mathcal{U},\mathcal{S}}$ for a user $\mathcal{U}$ holding a unique secret identifier $id_{\mathcal{U}}$. During this phase, $\mathcal{X}$ and $\mathcal{S}$ first agree on a pseudonym generation session identifier, defined as $sg_{id} = \{s\tilde{g}_{id}, \mathcal{X}, \mathcal{S}\}$, where $s\tilde{g}_{id}$ is a unique fresh random, $\mathcal{X}$ denotes the identity of the convertor and $\mathcal{S}$ represents the identity of the server.

The pseudonym generation is initiated by the user, relying on a $\text{OPRF}_{\mathcal{U},\mathcal{X}}$ function (cf. section IV-C2) based on the convertor's secret key $sk_{\mathcal{X}}$ and the user identifier $id_{\mathcal{U}}$ inputs, such that a core identifier c_{id} is generated and only known to $\mathcal{U}$, as follows: $c_{id} = \text{OPRF}_{\mathcal{U},\mathcal{X}}(id_{\mathcal{U}}, sk_{\mathcal{X}})$.

The core identifier c_{id} is blindly shared with the requesting $\mathcal{S}$, via $\mathcal{X}$. To do so, the user executes the following operations:

1) it relies on a homomorphic encryption (HE) mechanism as defined in section IV-C3, to partially encrypt, using the public key of the server $pk_{\mathcal{S}}$, the generated core identifier c_{id}, as $\tilde{nym}_{\mathcal{U},\mathcal{S}} = \text{HE.enc}_{\mathcal{U}}(pk_{\mathcal{S}}, c_{id})$.
2) a fresh generated random $t_{\mathcal{U},\mathcal{S}}$ is also generated and encrypted using the public key of the corresponding server $\mathcal{S}$ and an asymmetric $\text{enc}_{\mathcal{U}}$ algorithm, such that $\tilde{t}_{\mathcal{U},\mathcal{S}} = \text{enc}_{\mathcal{U}}(pk_{\mathcal{S}}, t_{\mathcal{U},\mathcal{S}})$.
3) $\mathcal{U}$ relies on the attribute based encryption (ABE) scheme to encipherthe permitted operations on shared data with $\mathcal{S}$, denoted by $AR_{\mathcal{U},\mathcal{S}}$, w.r.t. an access structure Γ such that only a set of auditors are allowed to decrypt the enciphered contents, such that $\tilde{AR}_{\mathcal{U},\mathcal{S}} = \text{ABE.enc}_{\mathcal{U}}(\Gamma, AR_{\mathcal{U},\mathcal{S}})$.

Afterwards, the tuple $(\tilde{nym}_{\mathcal{U},\mathcal{S}}, \tilde{t}_{\mathcal{U},\mathcal{S}}, \tilde{AR}_{\mathcal{U},\mathcal{S}})$ is sent to the convertor for further processing. That is, $\mathcal{X}$ homomorphically encrypts $1_{\mathbb{G}}$ using the public key of the corresponding server $\mathcal{S}$ and raises the resulting encrypted pseudonym $\tilde{nym'}_{\mathcal{U},\mathcal{S}}$ with a secret associated to the server $x_{\mathcal{S}}$, such that:

$$\tilde{nym'}_{\mathcal{U},\mathcal{S}} = [\tilde{nym}_{\mathcal{U},\mathcal{S}} \odot \text{HE.enc}_{\mathcal{X}}(pk_{\mathcal{S}}, 1_{\mathbb{G}})]^{x_{\mathcal{S}}}$$

Then, both $\tilde{nym'}_{\mathcal{U},\mathcal{S}}$ and $\tilde{t}_{\mathcal{U},\mathcal{S}}$ are transferred to the server $\mathcal{S}$. This latter can then decrypt $(\tilde{nym'}_{\mathcal{U},\mathcal{S}}, \tilde{t}_{\mathcal{U},\mathcal{S}})$ to retrieve the user pseudonym $nym_{\mathcal{U},\mathcal{S}}$ and the related auditing tag $t_{\mathcal{U},\mathcal{S}}$, as follows: $t_{\mathcal{U},\mathcal{S}} = \text{dec}_{\mathcal{S}}(sk_{\mathcal{S}}, \tilde{t}_{\mathcal{U},\mathcal{S}})$ and

$$nym_{\mathcal{U},\mathcal{S}} = \text{HE.dec}_{\mathcal{S}}(sk_{\mathcal{S}}, \tilde{nym'}_{\mathcal{U},\mathcal{S}}) = {c_{id}}^{x_{\mathcal{S}}}$$

Note that $\mathcal{S}$ may store a different pseudonym for each user in its database, for better security guarantees. This process is then registered in the blockchain by $\mathcal{X}$ in the form of a pseudonym generation transaction which is defined as: $TG_{\mathcal{X}} = (id_{TG_{\mathcal{X}}}, sg_{id}, \tilde{AR}_{\mathcal{U},\mathcal{S}}, \sigma_{\mathcal{X}})$, where:

- $id_{TG_{\mathcal{X}}}$ – denotes the transaction identifier associated with the pseudonym generation process.
- sg_{id} – represents the pseudonym generation session identifier, previously generated as $sg_{id} = \{s\tilde{g}_{id}, \mathcal{X}, \mathcal{S}\}$.
- $\tilde{AR}_{\mathcal{U},\mathcal{S}}$ – denotes the encryption of the access rights, relying on the ABE scheme, where only a set of auditors are allowed to decrypt permitted actions $AR_{\mathcal{U},\mathcal{S}}$ over the shared data.
- $\sigma_{\mathcal{X}}$ – is the signature of the convertor over all the transaction fields using his secret key $sk_{\mathcal{X}}$.

Note that the transaction identifier $id_{TG_{\mathcal{X}}}$ is then shared with the server $\mathcal{S}$ and the corresponding user $\mathcal{U}$.

3) Pseudonym Conversion: This phase starts when a server $\mathcal{S}_A$ requests data of user $\mathcal{U}$, from a server $\mathcal{S}_B$, via $\mathcal{X}$. It relies on two sub-phases, namely: *pseudonym conversion request* and *pseudonym conversion response*. Each conversion session is associated a conversion session identifier, referred to as sq_{id}. It is defined as $sq_{id} = \{s\tilde{q}_{id}, \mathcal{X}, \mathcal{S}_A, \mathcal{S}_B\}$, where $s\tilde{q}_{id}$ is a unique fresh random, $\mathcal{X}$ denotes the identity of the convertor and $\{\mathcal{S}_A, \mathcal{S}_B\}$ represents the identities of involved servers.

During the *pseudonym conversion request*, the requesting server $\mathcal{S}_A$ asks the convertor(s) $\mathcal{X}$ for converting the encrypted pseudonym $nym_{\mathcal{U},\mathcal{S}_A}$ towards the pseudonym $nym_{\mathcal{U},\mathcal{S}_B}$, associated with the server $\mathcal{S}_B$. Thus, $\mathcal{S}_A$ encrypts the inner pseudonym of the user using the public key of $\mathcal{S}_B$ such as:

$$\tilde{nym}_{\mathcal{U},\mathcal{S}_B} = \text{HE.enc}_{\mathcal{S}_A}(pk_{\mathcal{S}_B}, nym_{\mathcal{U},\mathcal{S}_A})$$

Then, $\mathcal{S}_A$ generates a new fresh tag ${t^{(AP)}}_{\mathcal{U},\mathcal{S}_A}$ related to the actual data processing denoted by (AP) and encrypts it using the public key of the user ${pk^{(AP)}}_{\mathcal{U}}$ known by $\mathcal{S}_A$ such as:

$${t^{(\tilde{AP})}}_{\mathcal{U},\mathcal{S}_A} = \text{renc}_{\mathcal{S}_A}({pk^{(AP)}}_{\mathcal{U}}, {t^{(AP)}}_{\mathcal{U},\mathcal{S}_A})$$

where ${pk^{(AP)}}_{\mathcal{U}} = \text{rand}_{\mathcal{S}_A}({pk^{(PP)}}_{\mathcal{U}})$ is the newly randomized public key of the user based on the previous public key ${pk^{(PP)}}_{\mathcal{U}}$ known and used by $\mathcal{S}_A$ for enciphering the relevant session tag during the previous data user processing (PP). The randomization process is presented in section IV-C4. This chaining process enables the user to check himself the consistency of his data usage (cf. section IV-D5). Then, $\mathcal{S}_A$ sends the couple $(\tilde{nym}_{\mathcal{U},\mathcal{S}_B}, {t^{(PP)}}_{\mathcal{U},\mathcal{S}_A})$ to the convertor to attain the conversion process with $\mathcal{S}_B$, where ${t^{(PP)}}_{\mathcal{U},\mathcal{S}_A}$ corresponds to the relevant session tag, generated by $\mathcal{S}_A$ during the previous data user processing (PP). The very first ${t^{(PP)}}_{\mathcal{U},\mathcal{S}_A}$ is the one provided by $\mathcal{U}$ during the pseudonym generation.

Afterwards, the *pseudonym conversion response* sub-phase starts between the convertor $\mathcal{X}$ and $\mathcal{S}_B$ to complete the conversion. First, $\mathcal{X}$ uses the homomorphic property of the encryption mechanism and encrypts $1_{\mathbb{G}}$ using the public key $pk_{\mathcal{S}_B}$, such as it raises the encrypted pseudonym to the quotient of servers' secrets $\frac{x_{\mathcal{S}_B}}{x_{\mathcal{S}_A}}$, such that:

$$\tilde{nym'}_{\mathcal{U},\mathcal{S}_B} = [\tilde{nym}_{\mathcal{U},\mathcal{S}_B} \odot \text{HE.enc}_{\mathcal{X}}(pk_{\mathcal{S}_B}, 1_{\mathbb{G}})]^{\frac{x_{\mathcal{S}_B}}{x_{\mathcal{S}_A}}}$$

Once received, $\mathcal{S}_B$ can then decrypt the received encrypted pseudonym and retrieves $nym_{\mathcal{U},\mathcal{S}_B}$ as stored in its database. If the data request is accepted by $\mathcal{S}_B$, this latter similarly sends the tag ${t^{(PP)}}_{\mathcal{U},\mathcal{S}_B}$ corresponding to the random tag, generated by $\mathcal{S}_B$, during the previous data user processing (PP).

Three transactions are then added to the blockchain by each participating entity of the conversion process. That is, the convertor adds a conversion transaction defined as $TC_{\mathcal{X}} = (id_{TC_{\mathcal{X}}}, sq_{id}, id_{DU}, ({t^{(PP)}}_{\mathcal{U},\mathcal{S}_A}, {t^{(PP)}}_{\mathcal{U},\mathcal{S}_B}), \sigma_{\mathcal{X}})$, where:

- $id_{TC_{\mathcal{X}}}$ – denotes the transaction identifier associated with the conversion process.
- sq_{id} – represents the generated conversion session identifier known as $sq_{id} = \{s\tilde{q}_{id}, \mathcal{X}, \mathcal{S}_A, \mathcal{S}_B\}$.
- id_{DU} – is the data usage identifier, specified by $\mathcal{S}_A$, when requesting the pseudonym conversion. The id_{DU} identifier may refer to data request, data transfer and so on.
- $({t^{(PP)}}_{\mathcal{U},\mathcal{S}_A}, {t^{(PP)}}_{\mathcal{U},\mathcal{S}_B})$ – denotes latest conversion processing tags on data user $\mathcal{U}$, received respectively from $\mathcal{S}_A$ and $\mathcal{S}_B$.
- $\sigma_{\mathcal{X}}$ – is the signature of the convertor, using his secret key $sk_{\mathcal{X}}$, over all the transaction fields.

Once added to the blockchain, the convertor shares the conversion transaction identifier, $id_{TC_{\mathcal{X}}}$ with both $\mathcal{S}_A$ and $\mathcal{S}_B$.

Afterwards, a conversion request and a conversion response transactions are added by $\mathcal{S}_A$ and $\mathcal{S}_B$ respectively, defined as: $TC_{\mathcal{S}_i} = (id_{TC_{\mathcal{S}_i}}, sq_{id}, t^{(PP)}{}_{\mathcal{U},\mathcal{S}_i}, t^{(\tilde{AP})}{}_{\mathcal{U},\mathcal{S}_i}, \tilde{M}_{\mathcal{S}_i}, \sigma_{\mathcal{S}_i})$, where $\mathcal{S}_i \in \{\mathcal{S}_A, \mathcal{S}_B\}$ and :

- $id_{TC_{\mathcal{S}_i}}$ – denotes the transaction identifier, added by $\mathcal{S}_i$ and associated with the conversion process.
- sq_{id} – represents the conversion session identifier where $sq_{id} = \{s\tilde{q}_{id}, \mathcal{X}, \mathcal{S}_A, \mathcal{S}_B\}$.
- $t^{(PP)}{}_{\mathcal{U},\mathcal{S}_i}$ – denotes the identifier of the previous data processing tag over the user data, executed by $\mathcal{S}_i$.
- $t^{(\tilde{AP})}{}_{\mathcal{U},\mathcal{S}_i}$ – denotes the re-encryption of the new generated identifier using a randomized key of the user, as detailed in section IV-C4.
- $\tilde{M}_{\mathcal{S}_i}$ – represents the encryption of a message $M_{\mathcal{S}_i}$, using an attribute based encryption mechanism, such as $\tilde{M}_{\mathcal{S}_i} = \mathsf{ABE.enc}_{\mathcal{S}_i}(\Gamma, M_{\mathcal{S}_i})$, where $M_{\mathcal{S}_i}$ is defined as $M_{\mathcal{S}_i} = id_{TC_{\mathcal{X}}}||id_{TG_{\mathcal{X}}}||id_{PT_{\mathcal{S}_i}}$, where $id_{TC_{\mathcal{X}}}$ is the pseudonym conversion transaction identifier, $id_{TG_{\mathcal{X}}}$ is the pseudonym generation transaction identifier and $id_{PT_{\mathcal{S}_i}}$ is the previous transaction identifier identifying the previous $id_{TC_{\mathcal{S}_i}}$ relative to the same user data and added by $\mathcal{S}_i$. Note that $id_{TG_{\mathcal{X}}}$ enables auditing authorities to retrieve the granted privileges AR and to check the compliance of server $\mathcal{S}_i$ when processing collected user data.
- $\sigma_{\mathcal{S}_i}$ – is the signature of the server over all the transaction fields using his secret key $sk_{\mathcal{S}_i}$.

Remark 2: **multi-level encryption of data usage transaction's history—** Note that servers $\mathcal{S}_A$ and $\mathcal{S}_B$ may rely on a multi-level ABE scheme (cf. section IV-C5), to encrypt their messages $M_{\mathcal{S}_A}$ and $M_{\mathcal{S}_B}$ respectively. That is, depending on their associated credentials (i.e; certified attributes), authorized authorities can access to a sub-set of the encrypted message $M_{\mathcal{S}_i}$ defined as $M_{\mathcal{S}_i} = id_{TC_{\mathcal{X}}}||id_{TG_{\mathcal{X}}}||id_{PT_{\mathcal{S}_i}}$. For instance, the authorized auditor $\mathcal{E}$ may be allowed to only check the consistency of the actual transaction. Hence, $\mathcal{E}$ deciphers the subset information $\{id_{TC_{\mathcal{X}}}, id_{TG_{\mathcal{X}}}\}$, such that he can verify the transaction by $\mathcal{S}$ making use of user data comply with the permissions AR to transfer or request user data, with no need to verify the whole history of operations.

4) Granted Privileges' Update: When $\mathcal{U}$ wants to change the granted privileges associated to a server $\mathcal{S}$, he has to send the updated $\tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}}$ such as $\tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}}$ $=\mathsf{ABE.enc}_{\mathcal{U}}(AR^{(up)}{}_{\mathcal{U},\mathcal{S}}||id_{TG_{\mathcal{X}}})$ to the convertor $\mathcal{X}$. This latter has then to inform the associated server $\mathcal{S}$ and a new granted privilege transaction is added to the blockchain, as follows: $TU_{\mathcal{X}} = (id_{TU_{\mathcal{X}}}, su_{id}, \tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}}, \sigma_{\mathcal{X}})$, where $id_{TU_{\mathcal{X}}}$ is the transaction update identifier, su_{id} is the session identifier, $\tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}}$ is the encryption of updated access privileges $AR^{(up)}{}_{\mathcal{U},\mathcal{S}}$ associated with the pseudonym generation transaction identifier and $\sigma_{\mathcal{X}}$ is the signature of the convertor.

When a server $\mathcal{S}_A$ requests a pseudonym conversion (respectively responds), it similarly adds a pseudonym conversion transaction, where

$$TC_{\mathcal{S}_A} = (id_{TC_{\mathcal{S}_A}}, su_{id}, t^{(PP)}{}_{\mathcal{U},\mathcal{S}_A}, t^{(\tilde{AP})}{}_{\mathcal{U},\mathcal{S}_A}, \tilde{M}_{\mathcal{S}_A}, \sigma_{\mathcal{S}_A})$$

where $\tilde{M}_{\mathcal{S}_A}$ is the encryption based on an ABE scheme of the message $M_{\mathcal{S}_A}$ defined as $M_{\mathcal{S}_A} = id_{TC_{\mathcal{X}}}||id_{TU_{\mathcal{X}}}||id_{PT_{\mathcal{S}_A}}$.

5) User and Authorities Audit: The auditing process can be performed either by the data user $\mathcal{U}$ or by an authorized auditing entities $\mathcal{D}$, with respect to their associated credentials.

The user audit process permits $\mathcal{U}$ to learn all the conversion requests and responses related to his pseudonym, as well as associated data usage. As such, the user starts from the shared tag $t_{\mathcal{U},\mathcal{S}_A}$ and $id_{TG_{\mathcal{X}}}$ shared with the server $\mathcal{S}_A$, during the pseudonym generation phase. Hence, $\mathcal{U}$ can iteratively retrieve the chain of new tags that were generated by $\mathcal{S}_A$ for every pseudonym conversion or response, as shown in Table II. The new tags are encrypted, relying on a re-randomizable key encryption scheme (cf. section IV-C4). For the sake of efficiency, the user may avoid checking the history of his pseudonym usage (based on the first shared tag with $\mathcal{S}_A$). Instead, $\mathcal{U}$ may periodically verify the latest added transactions and finally save the latest deciphered tag, for further auditing.

TABLE II
CHAINING ASSOCIATED TO THE DATA PROCESSING OF $\mathcal{U}$ BY $\mathcal{S}_A$

$t_{\mathcal{U},\mathcal{S}_A}$	$\tilde{t'}_{\mathcal{U},\mathcal{S}_A} = \mathsf{renc}_{\mathcal{S}_A}(pk'_{\mathcal{U}}, t'_{\mathcal{U},\mathcal{S}_A})$	where $pk'_{\mathcal{U}} = \mathsf{rand}_{\mathcal{S}_A}(pk_{\mathcal{U}})$
$t'_{\mathcal{U},\mathcal{S}_A}$	$t''_{\mathcal{U},\mathcal{S}_A} = \mathsf{renc}_{\mathcal{S}_A}(pk''_{\mathcal{U}}, t''_{\mathcal{U},\mathcal{S}_A})$	where $pk''_{\mathcal{U}} = \mathsf{rand}_{\mathcal{S}_A}(pk'_{\mathcal{U}})$
$\vdots$	$\vdots$	$\vdots$
$t^{(n-1)}{}_{\mathcal{U},\mathcal{S}_A}$	$t^{(n)}{}_{\mathcal{U},\mathcal{S}_A} = \mathsf{renc}_{\mathcal{S}_A}(pk^{(n)}{}_{\mathcal{U}}, t^{(n)}{}_{\mathcal{U},\mathcal{S}_A})$	where $pk^{(n)}{}_{\mathcal{U}} = \mathsf{rand}_{\mathcal{S}_A}(pk^{(n-1)}{}_{\mathcal{U}})$

The auditing process may be performed by a dedicated authority $\mathcal{D}$ w.r.t. its certified credentials, relying on a multi-level access scheme. That is, $\mathcal{D}$ has to verify the consistency of the transactions containing the audited server identity $\mathcal{S}_A$:

- server's activities auditing — $\mathcal{D}$ is able to verify the consistency of the information included in the transactions added by $\mathcal{S}_A$, while comparing for each conversion session, information included in the transactions added by the convertor and the corresponding server, namely, session identifiers sq_{id}, data usage identifier id_{DU}, $\cdots$ Thus, in this sub-case, $\mathcal{D}$ checks the server's activities independently, relying on the decryption of $\tilde{M}_{\mathcal{S}_A}$ defined as $M_{\mathcal{S}_A} = id_{TC_{\mathcal{X}}}||id_{TG_{\mathcal{X}}}||id_{PT_{\mathcal{S}_A}}$. As such, $\mathcal{D}$ can iteratively chain all the previous transactions based on $id_{PT_{\mathcal{S}_A}}$ and verify the permissionned operations w.r.t. $id_{TG_{\mathcal{X}}}$. Designed as a multi-level ABE algorithm, depending on his certified credentials, the auditor may be able to only decipher $id_{TC_{\mathcal{X}}}||id_{TG_{\mathcal{X}}}$. As such, he cannot retrieve the chain of activities w.r.t. each (yet anonymous) user.
- server's activities with respect to a specific data user auditing — similarly, the dedicated auditor $\mathcal{D}$ verifies the consistency of the information included in the transactions added by $\mathcal{S}_A$, while comparing for each conversion session, the information included in the transactions added by $\mathcal{X}$ and $\mathcal{S}_A$. Thus, in this auditing sub-case, $\mathcal{D}$ can check the server's activities with respect to each (yet anonymous) user, relying on the decryption of $\tilde{M}_{\mathcal{S}_A}$

defined as $M_{\mathcal{S}_A} = id_{TC_{\mathcal{X}}}||id_{TG_{\mathcal{X}}}||id_{PT_{\mathcal{S}_A}}$.
If granted privileges associated with $\mathcal{S}_A$ have been updated by $\mathcal{U}$, then $\mathcal{D}$ checks the server's activities, based on the decryption of $\tilde{M}_{\mathcal{S}_A}$ defined as $M_{\mathcal{S}_A} = id_{TC_{\mathcal{X}}}||id_{TU_{\mathcal{X}}}||id_{PT_{\mathcal{S}_A}}$, where $id_{TU_{\mathcal{X}}}$ is the identifier of an updated transaction defined as $TU_{\mathcal{X}} = (id_{TU_{\mathcal{X}}}, su_{id}, \tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}}, \sigma_{\mathcal{X}})$, such that $\tilde{AR}^{(up)}_{\mathcal{U},\mathcal{S}} = \mathsf{ABE.enc}_{\mathcal{U}}(AR^{(up)}{}_{\mathcal{U},\mathcal{S}}||id_{TG_{\mathcal{X}}})$, thus referring to the pseudonym generation transaction. As such, $\mathcal{D}$ can iteratively chain all the previous transactions based on $id_{PT_{\mathcal{S}_A}}$ and verify whether operations are permitted w.r.t. $id_{TG_{\mathcal{X}}}$ and $id_{TU_{\mathcal{X}}}$ Designed as a multi-level ABE algorithm, depending on his certified credentials, $\mathcal{D}$ is able to decipher the whole message. As such, he can iteratively retrieve the chain of activities w.r.t. each user.

Table III shows an example for the $\mathcal{BDUA}$ chaining process. That is, we consider three pseudonym' generation processes, referred to as "AA", "BB" and "CC" associated with $\mathcal{S}_C$, $\mathcal{S}_B$ and $\mathcal{S}_A$, w.r.t. "03", "02" and "01" sessions' identifiers and "$0A$", "$0B$" and "$0C$" shared user tags, respectively. In addition, Table III presents two pseudonyms' conversion processes between $(\mathcal{S}_B, \mathcal{S}_A)$ and $(\mathcal{S}_C, \mathcal{S}_A)$ associated with "10" and "30" sessions' identifiers respectively.

TABLE III
EXAMPLE OF CHAINING PROCESS FOR DATA PROCESSING AUDITING

$TC_{\mathcal{S}_C}$	"XX",	"30",	"$0A$",	$\mathsf{renc}_{\mathcal{S}_C}(pk'_{\mathcal{U}}, "1A")$,	$\mathsf{ABE.enc}_{\mathcal{S}_C}("XY"\|\|"AA"\|\|"OO")$
$TC_{\mathcal{S}_A}$	"YY",	"30",	"$1C$",	$\mathsf{renc}_{\mathcal{S}_A}(pk'_{\mathcal{U}}, "2C")$,	$\mathsf{ABE.enc}_{\mathcal{S}_A}("XY"\|\|"CC"\|\|"FF")$
$TC_{\mathcal{X}}$	"XY",	"30",	id_{DU},	("$1C$","$0A$")	
⋮	⋮		⋮		
$TC_{\mathcal{S}_B}$	"EE",	"10",	"$0B$",	$\mathsf{renc}_{\mathcal{S}_B}(pk'_{\mathcal{U}}, "1B")$,	$\mathsf{ABE.enc}_{\mathcal{S}_B}("EF"\|\|"BB"\|\|"JJ")$
$TC_{\mathcal{S}_A}$	"FF",	"10",	"$0C$",	$\mathsf{renc}_{\mathcal{S}_A}(pk'_{\mathcal{U}}, "1C")$,	$\mathsf{ABE.enc}_{\mathcal{S}_A}("EF"\|\|"CC"\|\|"DD")$
$TC_{\mathcal{X}}$	"EF",	"10",	id_{DU},	("$0C$","$0B$")	
⋮	⋮		⋮		
$TG_{\mathcal{X}}$	"AA",	"03",	$AR_{\mathcal{U},\mathcal{S}_C}$		
$TG_{\mathcal{X}}$	"BB",	"02",	$AR_{\mathcal{U},\mathcal{S}_B}$		
$TG_{\mathcal{X}}$	"CC",	"01",	$AR_{\mathcal{U},\mathcal{S}_A}$		

where $pk'_{\mathcal{U}}$ is a different randomized user key (cf. Table II) such as $pk'_{\mathcal{U}} = \mathsf{rand}_{\mathcal{S}_i}(pk_{\mathcal{U}})$ and $\mathcal{S}_i \in \{\mathcal{S}_A, \mathcal{S}_B, \mathcal{S}_C\}$

V. SECURITY ANALYSIS

This section presents the threat model and discusses the resistance of $\mathcal{BDUA}$ against unlinkability, auditability and availability attacks. Then, it gives a brief performances analysis.

A. Security Model

To design $\mathcal{BDUA}$, we consider two different adversaries: the convertor can be corrupted in a passive fashion, thus called *honest but curious*. That is, when corrupted, the adversary sees all the convertor's inputs and outputs and internal state, but the convertor behaviour remains honest. For servers and users, we consider active corruptions by the adversary, referred to as *malicious servers or users*, i.e; upon corruption the active adversary is in full control over the servers' or users' behaviour. Consequently, both attackers are able to read transactions addressed to BC, but only malicious adversaries try to include false information to registered transactions and drop correct ones. In the sequel, we consider a mixed adversarial setting that can actively corrupt servers and users but only gain passive control over the convertor, such that:
(i) by exploiting protocol messages, namely enciphered information, the convertor's secret key $sk_{\mathcal{X}}$, secret keys associated with servers, a curious $\mathcal{X}$ tries to trace or identify users. Similarly, malicious servers may rely on BC's transactions to identify the processing associated to specific data user.
(ii) based on previous data usage sessions, as well as provided BC data, an attacker tries to impersonate a user to afford some extra rights. Thus, a malicious server could use arbitrary combination of pseudonyms, public keys and tags via $\mathcal{X}$.
(iii) a malicious server attempts to prevent the publication of a legitimate transaction in BC. For example, 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.

B. Security Discussion

In this section, we discuss the resistance of $\mathcal{BDUA}$ against the considered threat model, as detailed above.

Unlinkability — Beyond data secrecy guarantees, strongly related to the security of selected encryption schemes, the unlinkability property is ensured in $\mathcal{BDUA}$ thanks to:

- a randomized user key is generated for each conversion procedure – to address the trade-off between transparency and auditability features, $\mathcal{BDUA}$ relies on an iterative tag encrypted chaining process, as detailed in section IV-D5. Clearly, if the public key of the user $pk_{\mathcal{U}}$ is static, this would immediately destroy our privacy features. As such, $\mathcal{BDUA}$ considers that each new generated tag have to be encrypted with a different user public key $pk_{\mathcal{U}}$ based on ElGamal randomizable public key instantiation.
- different sessions' identifiers – session identifiers include the identities of involved parties as well as a unique fresh random. They have to be added to the set of transactions registered by each participating entity, namely servers and the convertor, thus creating an unlinkable yet controlled means for each pseudonym conversion procedure. As such, different sessions related to the same user cannot be linked. However, consistency of registered transactions may be detected by external readers of BC-transactions.

Auditability — $\mathcal{BDUA}$ ensures the auditibility feature, w.r.t. the integrity and access control requirements, as follows:

- tamper-proof architecture – all the blockchain-specific operations, like transaction anchoring activities, are considered as secure and non-corruptible, thus ensuring non-tamper proofs of data processing and managing events.
- transparent usage – $\mathcal{BDUA}$ relies on a BC infrastructure, that permits public access (i.e; read privilege) to registered transactions. Thus, it provides a transparent view over how data are processed.
- signed transactions – $\mathcal{BDUA}$ relies on signed transactions. Both pseudonyms generation and conversion procedures' transactions have to be signed by the involved parties. Signed transactions ensure that each activity is

accounted by the holder of the used private key. As such, the resistance of the chosen encryption and signature schemes against forgery attacks has a direct impact on the fulfillment of the auditability requirement.
- authorized access to enciphered auditing information – the auditing process relies on the security of the selected attribute based encryption scheme. As such, only entities satisfying the access policy set by the data owner may access to auditing information, based on transactions' identifiers chaining process, as detailed in section IV-D5.

Availability — As a highly decentralized infrastructure, the blockchain technology helps in terms of availability. It becomes possible to provide liveness guarantees of data usage. Thus, $\mathcal{BDUA}$ inherits the availability property from the Blockchain intrinsic features, relying on the replication of the whole chain on different independent nodes.

C. Performance Discussion

Based on the instantiations detailed in section IV-C and the CP-ABE scheme introduced by Bethencourt et al. [18], Table IV details the amount of exponentiation, multiplication and pairing functions in the respective groups.

For more realistic performance results of our scheme, we

TABLE IV
PROCESSING COMPLEXITY $\mathcal{BDUA}$ PROCEDURES

Generation	$\mathcal{U}$	$\mathcal{X}$	$\mathcal{S}_A$
	$\gamma_O + 2\gamma_M + 2(2 + Y_\Gamma)\gamma_E$	$4\gamma_E + 3\gamma_M$	$2\gamma_E + 2\gamma_M$
Conversion	$\mathcal{S}_A$	$\mathcal{X}$	$\mathcal{S}_B$
	$3\gamma_M + 2(4 + Y_\Gamma)\gamma_E$	$2\gamma_E$	$3\gamma_M + 2(3 + Y_\Gamma)\gamma_E$
Audit	$\mathcal{U}$	$\mathcal{D}$	while considering
	$c\gamma_E + c\gamma_M$	$c(2\gamma_P + \gamma_E + \gamma_M)Y_\Gamma$	c BC transactions

where γ_O denotes the cost of the OPRF function execution, γ_E, γ_M and γ_P denote the cost of the exponentiation, multiplication and pairing functions' processing cost and Y_Γ is the number of attributes in the access policy Γ.

conducted some experiments, for several mathematical operations (i.e; exponentiation, multiplication and pairing functions) on an Intel *E5-1650-v3* 6 cores. Our measurements show that the computation of a symmetric pairing function requires approximately 6 ms, while exponentiations and multiplications take about 1.2 ms and 0.5 ms, respectively.

Referring to the cpabe toolkit[4] proposed in [18], the computation costs of the key generation, encryption and decryption algorithms are mainly depending on the number of attributes. The cpabe toolkit provides a set of programs implementing CP-ABE schemes, using the PBC library[5] for the algebraic operations. The code is split into two packages, libbswabe (i.e; a library implementing the core cryptographic operations) and cpabe (i.e; higher level functions and user interface). In addition, as stated above, based on the cpabe toolkit, the encryption algorithm takes about 1.5 second based on an access tree containing around 60 attributes [18].

VI. CONCLUSION

Personal data are highly exposed to data leakage and misuse by third parties. As such, there is strong interest for users to get full control on their personal data usage without compromising their privacy or limiting authorities' ability for auditing activities. This paper proposes a new blockchain-based solution for data usage auditing that does not require a fully trusted auditor or convertor, thus answering an open question from [2]. The convertor blindly derives and converts unlinkable pseudonyms approved by involved parties for each $\mathcal{BDUA}$ procedure. Based on a blockchain infrastructure, $\mathcal{BDUA}$ enables the user to grant consent to service providers, specify their data access policy and track data usage flows in a trusted and privacy-preserving manner. Second, it provides a regulatory framework to properly enforce the legislation. Third, the deployment of a blockchain infrastructure helps in resolving availability concerns as blockchain transactions are replicated a large number of times on independent nodes.

[4]http://acsc.cs.utexas.edu/cpabe/index.html

[5]https://crypto.stanford.edu/pbc/

REFERENCES

[1] J. Camenisch and A. Lehmann, "(un) linkable pseudonyms for governmental databases," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015.

[2] ——, "Privacy-preserving user-auditable pseudonym systems," in *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2017.

[3] Regulation(EU), "2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec," 2016.

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

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

[6] 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.

[7] G. Zyskind, O. Nathan, and A. Pantland, "Decentralizing privacy: Using blockchain to protect personal data," in *IEEE Security and Privacy Workshops (SPW)*. IEEE, 2015, pp. 180–184.

[8] N. Kaaniche and M. Laurent, "A blockchain-based data usage auditing architecture with enhanced privacy and availability," in *NCA 2017: 16th IEEE International Symposium on Network Computing and Applications*. IEEE Computer Society, 2017, pp. 1–5.

[9] R. Neisse, G. Steri, and I. Nai-Fovino, "A blockchain-based approach for data accountability and provenance tracking," *arXiv preprint arXiv:1706.04507*, 2017.

[10] K. Wouters, K. Simoens, D. Lathouwers, and B. Preneel, "Secure and privacy-friendly logging for egovernment services," in *Availability, Reliability and Security, 2008. ARES 08. Third International Conference on*. IEEE, 2008, pp. 1091–1096.

[11] T. Pulls, R. Peeters, and K. Wouters, "Distributed privacy-preserving transparency logging," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 83–94.

[12] D. Galindo and E. R. Verheul, "Microdata sharing via pseudonymization," *Joint UNECE/Eurostat work session on statistical data confidentiality*, 2007.

[13] N. Kaaniche and M. Laurent, "Attribute based encryption for multi-level access control policies," in *SECRYPT 2017: 14th International Conference on Security and Cryptography*. Scitepress, 2017.

[14] S. Jarecki and X. Liu, "Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection." in *TCC*, vol. 5444. Springer, 2009, pp. 577–594.

[15] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys." in *Public Key Cryptography*. Springer, 2005.

[16] B. R. Waters, E. W. Felten, and A. Sahai, "Receiver anonymity via incomparable public keys," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 112–121.

[17] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Phoabe: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted iot," *Computer Networks*, 2018.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *IEEE Symposium on Security and Privacy*, 2007.