LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

RESEARCH ARTICLE

A Decentralized Identity-Based Blockchain Solution for Privacy-Preserving Licensing of Individual-Controlled Data to Prevent Unauthorized Secondary Data Usage

Meng Kang,*† Victoria L. Lemieux‡

Abstract. This paper presents a design for a blockchain solution aimed at the prevention of unauthorized secondary use of data. This solution brings together advances from the fields of identity management, confidential computing, and advanced data usage control. In the area of identity management, the solution is aligned with emerging decentralized identity standards: decentralized identifiers (DIDs), DID communication and verifiable credentials (VCs). In respect to confidential computing, the Cheon-Kim-Kim-Song (CKKS) fully homomorphic encryption (FHE) scheme is incorporated with the system to protect the privacy of the individual's data and prevent unauthorized secondary use when being shared with potential users. In the area of advanced data usage control, the solution leverages the PRIV-DRM solution architecture to derive a novel approach to licensing of data usage to prevent unauthorized secondary usage of data held by individuals. Specifically, our design covers necessary roles in the data-sharing ecosystem: the issuer of personal data, the individual holder of the personal data (*i.e.*, the data subject), a trusted data storage manager, a trusted license distributor, and the data consumer. The proof-of-concept implementation utilizes the decentralized identity framework being developed by the Hyperledger Indy/Aries project. A genomic data licensing use case is evaluated, which shows the feasibility and scalability of the solution.

1. Introduction

The Great Hack, a 2019 documentary film about the Facebook-Cambridge Analytica data scandal, has come to symbolize the kinds of data abuse associated with many social media and sharing economy platforms.[1] On such platforms, individuals have relatively little control and privacy over their data. Their information is very likely to be used without authorization for purposes other than the one for which the individual originally shared it with a platform. Such unauthorized secondary use raises privacy and data protection concerns; for example, unauthorized access could have negative effects for individuals and is often not compliant with laws such as the General

* bc1q4vpzht2wlcsf4dp47kulvy03swyv7nht2kksg2

† M. Kang (meng.kang@ubc.ca) is a master's degree student at the School of Engineering at the University of British Columbia, Kelowna, BC, Canada.

‡ V. L. Lemieux (v.lemieux@ubc.ca) is an Associate Professor of Archival Science at the School of Information at the University of British Columbia, Vancouver, BC, Canada and Founder of and Co-Lead of Blockchain@UBC.

Data Protection Regulation (GDPR),[2] and the Health Insurance Portability and Accountability Act (HIPAA).[3] Thus, there is a need for a solution that gives individuals greater control over the usage of their personal data with trust and in a privacy-preserving manner that prevents unauthorized secondary use (or what might be called "data double-spending").

Recent research aims at developing privacy-enhancing technologies. There are a number of works that address the enforcement of data usage control policies within individual systems, and recent works extend data usage control models and implementation to decentralized environments. Gaber *et al.*, for example, proposed a privacy-preserving digital rights management system that allows consumers to license digital content without disclosing complete personal information.[4] However, the application of blockchain and distributed ledger technologies to the enforcement of data usage policies aimed at preventing unauthorized secondary use still has not been fully researched.

In parallel, blockchain-based decentralized identity has emerged as a new privacy-preserving approach to identity management. Decentralized identity management provides for individuals to have full control over the use of their own digital identity data. While current decentralized identity-based solution designs are almost exclusively used for identity or certification authentication, decentralized identity systems have the potential to enable individuals to have control over a much wider range of personal data for use beyond identity authentication or proofing certification, a capability that can be leveraged to address the issues raised above.

Recently, solutions based on decentralized identity blockchains have received the attention of some researchers: Belchior *et al.* proposed an SSI-based access control solution, which shows how DIDs and VCs can be integrated with attribute-based access control in a federated setting, minimizing data disclosure and data redundancy;[5] Liu *et al.* proposed an SSI-based solution for multimedia data management;[6] Papadopoulos *et al.* presented a privacy-preserving decentralized workflow that facilitates trusted federated learning among participants;[7] and Lemieux *et al.* proposed a self-sovereign health data management solution, where VCs were used to enable privacy-preserving sharing of personally-identifiable health data.[8] However, the existing solutions do not address the problem of preventing the unauthorized usage of the data after the identity or other data have been shared.

On the other hand, how to consume data while being able to protect data privacy and security as well as prevent leakage of sensitive information is a major challenge in data science today. Confidential computing was born to solve this problem.[9] A privacy-enhancing cryptography-based technology in the field of confidential computing, known as Fully Homomorphic Encryption (FHE), was introduced in 2009.[10] FHE allows computations on encrypted data while preserving the features and the format of the plain text. Thus, sensitive data, such as genomic and health data, could be stored and computed in the cloud in an encrypted form without losing the utility of the data.

Efforts to incorporate "Privacy by Design" call for designing privacy up front into information systems.[11] While Privacy by Design ensures that privacy (and by association, privacy-enhancing security features) are not afterthoughts in system design, there is still limited technological means to prevent services from aggregating and using personal information they receive from individuals for one purpose and using it for another purpose without their consent. Individuals have to trust services to handle their data in accordance with the terms and conditions of consent and law, rather than having the certainty that it is technologically infeasible for services to reuse

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

their data without consent. Thus, this study will address this technical gap using credential-based tokenization combined with digital rights management concepts to provide novel capabilities which could be incorporated into services designed to prevent the reproduction and reuse of user data without consent.

In this paper, we aim to combine and build upon the previously-discussed research on privacy-enhancing technologies, including data usage control, decentralized identity systems, and homomorphic encryption, to provide a novel design and implementation of a fully-decentralized and privacy-preserving usage control enforcement infrastructure that overcomes the limitations of centralized data usage control wherein (1) formulation of the data usage policies (*i.e.*, licensing) and subsequent data usage is under the control of the individual data subject, (2) the data subject receives compensation for the use of their data, and (3) secondary use of the data that has not been authorized by the data subject is prevented.

The design at the logical level is inspired by and derived from PRIV-DRM, a privacy-preserving digital rights management solution developed by Gaber *et al.*[4] While the PRIV-DRM solution sees individuals as content consumers and companies as content providers, in our design, individuals are no longer *consumers* of data, but *controllers* of data. In other words, we flip the PRIV-DRM model to give individuals greater sovereignty over their data. The solution also leverages prior work on self-sovereign health data management, which introduces a privacy preserving and secure solution for self-sovereign data holders to share their health data for research purposes.[8] Thus, we combine the flipped PRIV-DRM model with use of Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) to establish a self-sovereign oriented distributed trust architecture. We add FHE capabilities to achieve additional privacy protection and data usage control. Our solution design allows for data holders to configure a license for their data, integrates a payment system, uses VCs to provision data consumers with data access, and uses FHE to protect the privacy of data and prevent unauthorized secondary use. We illustrate the operation of our implemented architecture with the METABRIC dataset.

2. Background Literature

2.1. Blockchains, Tokenization, SSI and Identity Management—Blockchains, which are ledgers that are shared across a set of nodes and synchronized using a consensus mechanism, are designed to be tamper resistant, append-only and immutable, through confirmed and validated transactions.[12] The novel technical capabilities afforded by blockchains have generated the means to render the mutability of digital data into immutable digital assets—returning to such data some of the properties of physical documents.[13] This is achieved via tokenization.

In the context of blockchains, cryptographic tokens represent programmable assets or access rights, often managed by smart contracts and an underlying distributed ledger. They are accessible only by the person who has the private key, which is kept secret and which should only be used by that entity. As Sean *et al.* note, while increasing numbers of people are starting to create and invest in cryptographic tokens, the understanding of different token types is still limited, even among professional investors and seasoned members of the blockchain community.[14] To add to the confusion, terms like 'cryptocurrency,' 'crypto assets,' and 'tokens' are very often used synonymously. The media mostly tends to refer to these new assets as 'cryptocurrencies,' which is often used to describe a diverse range of 'crypto assets' or 'tokens' that could represent

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

virtually anything: a physical good, digital good, security, a collectible, royalty, reward, or ticket to a concert. Tokenization has giving rise to a new form of economics, 'tokenomics,' and to new services and markets based on 'verified information.'[14,15]

In common tokenized recordkeeping systems, including Bitcoin and Ethereum, tokens may be transferred, with the transfer transaction being recorded on the chain via ledger records.[16,17] When this type of system was first proposed, tokens often represented valuable physical assets. Over time, tokens have come to represent digital assets of value as well, such as tradable collectables (*e.g.*, 'cryptokitties') which are represented as non-fungible tokens (NFTs) that cannot be divided up into smaller units of exchange.

In parallel, another type of tokenized system has emerged, the identity-based distributed ledger system, with Self-Sovereign Identity (SSI) systems being a novel and growing variant of this class of systems, representing the latest evolution in identity management. The SSI is a new type of digital identity that frequently utilizes distributed ledger technology or a blockchain to implement identities in a decentralized manner. In 2016, Christoper Allen proposed the concept of SSI, which allows individuals to have full control over the use of their own digital identity.[18] Later, the principles were grouped into three categories *security* (the identity information must be protected from unintentional disclosure), *controllability* (the identity holder must be in control of who can see and access their data and for what purposes), and *portability*.[19]

Compared with other recordkeeping solutions, SSI credential-based solutions have different privacy guarantees. The design of SSI systems solves a recordkeeping problem that plagues other types of distributed ledgers (*i.e.*, personal information leakage from recording transactions on ledger). These systems, in which records are stored on a ledger, frequently contain personally identifiable information or metadata that could lead to re-identification of an individual. This can happen when clinical trial consent transactions are stored on a blockchain, putting compliance with privacy and data protection laws at risk. Since they do not require the deletion of data from a ledger, because they do not record peer exchanges on ledger, SSI systems are designed to be highly privacy-preserving and compliant with the General Data Protection Regulations (GDPR) and similar data protection legislation in other jurisdictions.

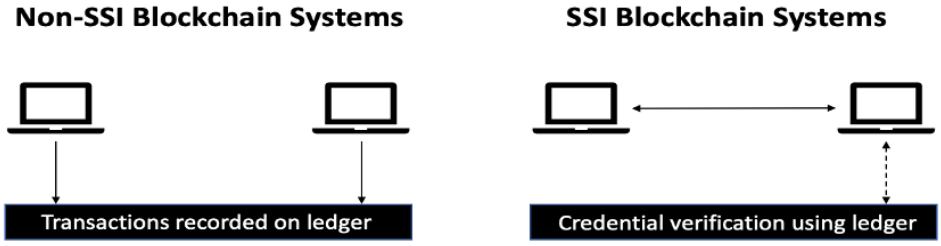


Fig. 1. Comparison of non-SSI and SSI systems showing that transactions are not recorded on ledger in SSI systems.

In SSI systems, assets, such as education certificates and health records, are converted into non-tradable tokens (*i.e.*, VCs) that can be used by one party to a transaction to make a verifiable 'truth claim' to another party to the transaction. The transaction and exchange of information is commonly not recorded in the form of a ledger record on-chain. Figure 1 shows the comparison of SSI and non-SSI systems. For instance, in the Hyperledger Indy/Aries project adopted in this

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

paper,[20] the blockchain records and keeps only public Decentralized Identifiers (DIDs), data schemas, credential definitions, and revocation registries on the blockchain.[21] Although the use of a blockchain is not required by SSI, with the privacy and security advantages, blockchain-based SSI systems are favoured by many.[22,23]

The SSI architecture was first adopted on general-purpose public blockchains like Ethereum (*e.g.*, the uPort project,[24] which uses the cryptographic address of a transaction on the ledger as the DID). Since anyone can operate nodes and take part in the consensus process in a public permissionless blockchain, every transaction made by the user could be found and possibly be correlated, this kind of solution poses additional challenges in terms of GDPR compliance.[25] In addition, it does not provide a governance framework for greater accountability. Special-purpose blockchain designed for SSI—the Sovrin SSI ecosystem utilizing public permissioned blockchains (Hyperledger Indy)—provides a higher degree of data minimization. The Sovrin framework also comprises a governance model and is operated by a consortium of trusted organizations, which is relatively more compatible with GDPR.[26]

Instead of using blockchain technology for cryptocurrency, SSI uses it for decentralized public key infrastructure (DPKI) in the field of identity management.This self-sovereign model is an improvement on the previous identity management models including the centralized model and federated models, because it removes the third-party identity provider and offers a direct connectivity between the individual and organization.[26] The most significant difference in the SSI model is that it is no longer based on accounts. Instead, it functions similarly to how identity works in the real world: it is based on a direct peer relationship between the individual and another party. Neither party 'provides,' 'controls,' or 'owns' the other's relationship.

2.2. Data Usage Control and Digital Rights Management—There is very little standing in the way of unauthorized secondary use of data. While lawmakers are enacting tighter data protection regulations,[27,28] individuals are often overwhelmed with long, complex consent forms,[28] which means they may inadvertently authorize secondary use of their data out of confusion. Even if they avoid giving inadvertent consent for the sharing of their data, they have no way of knowing if the service they are giving consent to is adhering to the terms and conditions of that consent. Regulators have begun to issue larger fines for such data abuses,[29] but a policy response is unlikely to be sufficient on its own to address the problem.

Tokenization may provide a means of achieving better usage control over digital data. One potential avenue of data usage control that could leverage tokenization is digital rights management. The purpose of digital rights management is to prevent unauthorized redistribution of digital media by controlling and regulating whether/how consumers can copy the content that they have purchased.[30] Digital rights management software aims to assure copyright protection for digital media. Typically, digital rights management solutions are designed with the view that content providers are corporations, and therefore aim to protect corporate interests; because of this, they are not optimized to protect individuals' data from unauthorized secondary use.

Related work by Pretschner *et al.* aims to enable the verifiability of data custody and a mechanism for data subjects to issue notifications about the misuse of their data.[31] From the data controller and processor point of view, the main benefit is a certified proof that can be presented to supervisory authorities showing that the data was obtained in compliance with privacy regulations. One weakness of Pretschner's approach is that the system architecture relies on a centralized trust point—the Policy Decision Point—thereby creating a single point of attack,

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

manipulation and/or fraud.

Recently, Gaber *et al.* presented a solution to preserve the privacy of digital content consumers in a multi-party digital rights system, which allows a consumer to acquire digital content with a license without disclosing complete personal information or using third parties.[4] Their design still envisions corporations as content owners; however, it can be adapted to a scenario wherein individuals have control over their data.

In this work, we extend ideas from Pretchner and Gaber to SSI blockchain systems as a decentralized trust anchor. In contrast to most digital rights management systems where the service provider (*e.g.*, a corporation) is at the center of the identity model, our model is user centric.[22] The relation between the different actors of the system can be observed in Figure 2. By attesting to certain attributes of the user, the claim issuer issues (at least part of) the identity. The user has complete control over his or her identity. Any relying party that requires the user's identity will be presented with the relevant parts of the user-controlled identity. The relying party must have a trusting relationship with the claim issuer in order to accept the identity.

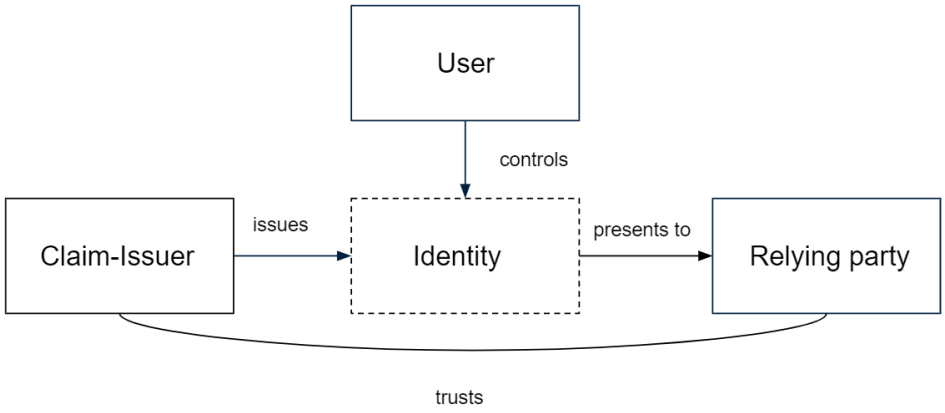


Fig. 2. Self-Sovereign Identity Actors.[22]

2.3. Confidential Computing and Fully Homomorphic Encryption—Confidential computing, or privacy computing, in a broad sense, refers to computing systems and technologies oriented toward privacy protection, covering the entire process of information processes such as data generation, storage, computation, application, and destruction, and the desired effect is to make data 'available and invisible' in all aspects.[32]

Current approaches to implementing confidential computing can be divided into several categories, such as cryptography-based approaches, federated learning, and Trusted Execution Environments (TEE). Federated learning and TEE currently suffer from weak security, and new security vulnerabilities are frequently discovered. Furthermore, the core part of federated learning frequently necessitates strong security via cryptographic techniques such as homomorphic encryption (introduced below).[33] Unlike the implementation of confidential computing at the software and algorithm levels, the TEE is based on hardware implementation.[34] The idea of this approach is to build a secure area on the CPU, which serves to provide a more secure space for data and code execution, and to perform relevant computations within this secure area. The more representative ones are Intel-SGX and ARM-TrustZone.[35,36] Therefore, TEE, as a technology

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

to protect data privacy and security, has the advantages of low cost and high efficiency, but because it is a hardware-based solution, it requires that a specific hardware vendor be trusted. For example, if the confidential computing is done through SGX, since it has to go to Intel's servers for remote authentication, it requires that all the parties involved place trust in Intel and the secure operation of the hardware. Therefore, TEE-based privacy computing may not suitable for some scenarios with higher privacy requirements.

The cryptography-based approaches refer to algorithms to achieve data protection during computing, represented by secure multi-party computation (MPC), which is a concept of collaborative computing done securely with the joint participation of multiple parties without trusted third parties.[37] The supporting technology layer provides the basic technology implementation for building MPC, which includes fully homomorphic encryption, secret sharing, and the use of a garbled circuit.[38–40] Both secret sharing and garbled circuit are interactive, *i.e.*, multiple parties have to be online at the same time to exchange information such as keys. Fully homomorphic encryption, on the other hand, does not require interaction and thus allows offline ciphertext computation. Therefore, in this study, we choose homomorphic encryption in the field of confidential computing as a method to protect personal data in the whole design.

Cryptography is a communication technique where one party encrypts a message and sends it to the other party who then decrypts it. Public-key encryption, conceived in the foundational work of Diffie and Hellman, and first developed by Rivest, Shamir, and Adleman, provides data owners with a way to encrypt a message using the data consumer's public key into a ciphertext, and for the consumer to decrypt the ciphertext to obtain the message using her secret key.[41,42] The whole message can be obtained from the perspective of this encryption scheme by possessing the secret decryption key, but the ciphertext is entirely useless without the decryption key.

This scenario poses a fascinating issue. In 1978, Rivest, Adleman and Dertouzos first mentioned the idea of a system that could indirectly manipulate the original text by performing certain calculations on the ciphertext in their paper titled "On Data Banks and Privacy Homomorphisms."[43] This idea was later recapitulated and renamed as fully homomorphic encryption.

When the concept of FHE was proposed, the entire academic community began a decades-long search, trying to find a perfect algorithm with fully homomorphic properties. Researchers tried all imaginable options, but could not find an option that satisfied all conditions of full homomorphism. A variety of proposed schemes of encryption that are either additively or multiplicatively homomorphic were proposed, which are defined as partially homomorphic encryption (PHE). This includes the El Gamal scheme and the RSA scheme.[42,44]

In 2009, Craig Gentry proposed the first fully homomorphic encryption scheme in his thesis, which has the capacity to perform a theoretically unlimited number of additions and multiplications aiding research into the practical application of FHE.[10] Since then, the academic community has spent much more effort on optimizing FHE schemes and developing libraries.

Currently, the Cheon-Kim-Kim-Song (CKKS) FHE scheme is the most effective method of performing approximate homomorphic computations over real and complex numbers.[45] The CKKS scheme has already been implemented to achieve levels of efficiency for processing for homomorphically-encrypted data in the field of cyber physical systems and machine learning.[46–51] At the same time, we have to admit that fully homomorphic encryption is still in the exploratory stage, and the existing performance is still some distance away from large-scale commercial engineering applicability due to heavy demands for computing resources. By foregoing exact

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

computations, the CKKS scheme achieves significant improvements in the ciphertext/plaintext ratio and algorithm speed. This scheme was used in the winning solution of the iDASH biomedical privacy competition in 2017 for privacy-preserving logistic regression on genomics data. Therefore, this scheme is adopted in our solution as an encryption method for preserving the privacy of personal data.

3. Preliminaries

3.1. Decentralized Identifiers—A decentralized identifier (DID) is a new type of globally unique identifier created by the World Wide Web Consortium (W3C) working group.[52] It consists of three elements, as shown below, which follows URN syntax defined in RFC 8141:

$$\underbrace{\text{did}}_{\text{DID Scheme}} : \underbrace{\text{example}}_{\text{DID Method}} : \underbrace{\text{123456789abcdefghijk}}_{\text{Method-Specific Identifier}} \tag{1}$$

where the scheme is fixed for all DIDs.[53] The method describes how DIDs work with a specific ledger.[54] The method-specific identifier is an alphanumeric string that is guaranteed to be unique within the context of the DID method. The DID method shows the way to resolve a DID to the associated DID document.

This approach has proven to be popular for linking a globally unique identifier to cryptographic keys and other interaction metadata required to prove control of the identifier. As illustrated in the following listing, where '*id*' denotes the DID, '*authentication*' is a list of protocols for verifying the control of the DID, and '*service*' is a set of service endpoints for interacting with the entity that the DID identifies.

Listing 1. Example DID Document

```
{
 "@context": "https://www.w3.org/ns/did/v1",
 "id": "did:example:123456789abcdefghi",
 "authentication": [{
 "id": "did:example:123456789abcdefghi#keys-1",
 "type": "Ed25519VerificationKey2018",
 "controller": "did:example:123456789abcdefghi",
 "publicKeyBase58" : "H3C2AVvLMv6gmMNam3
   uVAjZpfkcJCwDwnZn6z3wXmqPV"
 }],
 "service": [{
 "id":"did:example:123456789abcdefghi#vcs",
 "type": "VerifiableCredentialService",
 "serviceEndpoint": "https://example.com/vc/"
 }]
}
```

DID specifications ensure interoperability between DID schemes, allowing any storage system to interact with and resolve a DID. On the other hand, peer DID implementations are used in peer-to-peer connections that do not require a storage system and in which each peer stores and

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

maintains their own list of DID documents.[54]

3.2. DID Communication—DID Communication (DIDComm) is an asynchronous encrypted communication protocol that has been developed as part of the Hyperledger Aries project.[55] It creates a cryptographically secured way by which any two software agents (peers) can communicate directly from edge to edge or through intermediate cloud agents. In DIDComm, peers who are parties to the connection are individually responsible for the generation of their DID, the key pairs in a DID document required to establish the secure messaging between them, and the subsequent key rotation or revocation of those keys. DIDComm exchanges secure messages by utilising information from the DID document, such as the public key and its associated endpoint. It allows different entities to link with each other peer-to-peer with no middle man. The following algorithm shows a secure and private DIDComm example between Alice and Bob, where Alice first encrypts and signs a message for Bob. The signature and the cipher text are then sent through Alice's endpoint to Bob's endpoint. The authenticity of the message can be checked by resolving the DID and identifying whether it matches Alice's public key.

1 Alice has a secret key (sk_a), a DID Document for Bob which contains an endpoint ($endpoint_{bob}$), and a public key (pk_b).
2 Bob has a secret key (sk_b), a DID Document for Alice which contains Alice's public key (pk_a).
3 Alice encrypts the plain text message (m) using pk_b, creates cipher text (ct_b).
4 Alice signs ct_b using sk_a to create a signature (σ).
5 Alice sends (ct_b,σ) to $endpoint_{bob}$.
6 Bob verifies σ using pk_a.
7 **if** *(Verified)* **then**
8 | Bob decrypts ct_b using sk_b.
9 | Bob reads m.
10 **end**

Algorithm 1: DID Communication Example

3.3. Verifiable Credentials—A credential is a document detailing the qualification, ability, or authority that a third party with the relevant authority or assumed ability has given to an individual. For instance, a driver's license is used to demonstrate that a person has the ability to drive a motor vehicle, a university degree can be used to demonstrate the level of education of a person, and a passport issued by the government allows people to travel between countries. These physical credentials might consist of information related to the identifier (*e.g.*, photo, identification number), the issuing authority, specific attributes or properties being asserted by the issuing authority, and constraints on the credential. All the same information that a physical credential represents can be represented by a verifiable credential (VC), defined as a tamper-evident credential that has authorship which can be cryptographically verified.[56] The Verifiable Credential Data Model specification became a recommended standard by W3C in 2019.[56]

The following roles are introduced in this specification:

- *Credential*: A credential is a set of claims made by an issuer. A *verifiable credential* is a tamper-evident credential that has authorship that can be cryptographically verified.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

- *Issuer:* A role that an entity can perform by asserting claims about one or more subjects, creating a verifiable credential from these claims, and transmitting the verifiable credential to a holder.
- *Holder:* A role that an entity might perform by possessing one or more verifiable credentials and generating presentations from them.
- *Verifier:* The entity verifying a claim about a given subject.
- *Proof Request:* The request for one or more verifiable credentials, issued by one or more issuers, held by one holder.
- *Proof Presentation:* Data derived from one or more verifiable credentials, issued by one or more issuers, that is shared with a specific verifier.
- *Proof Verification:* The cryptographic evaluation of whether a verifiable credential or verifiable presentation is an authentic statement of the issuer or presenter, respectively.
- *Verifiable Data Registry:* An internet-accessible registry that holds all essential data and metadata that enables the VC model to operate, which includes the public keys of issuers, and the schema or ontology for all the properties that the VC may contain. The registry can be implemented using blockchain technology.

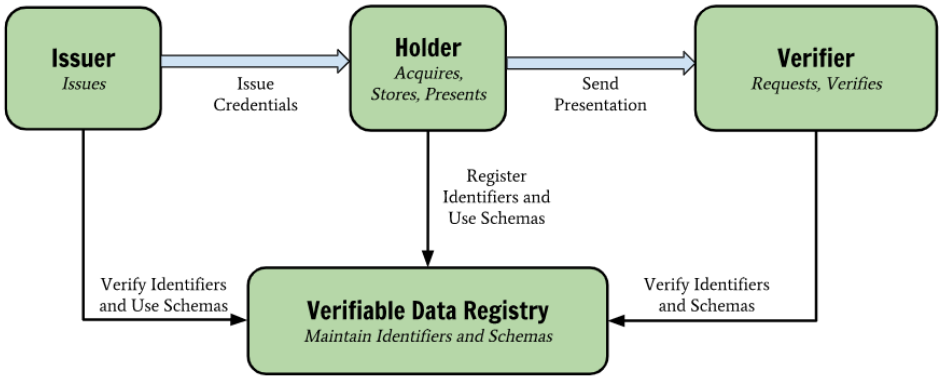


Fig. 3. The roles and information flows forming the basis for Verifiable Credential Data Model specification.

The exchange of verifiable credentials is carried out by agents using the DIDComm protocol extension, which is an area of significant activity in the Hyperledger Aries project, where specifications for the extension protocol are being released as part of the DIDComm suite. This credential exchange protocol supports zero-knowledge proof (ZKP) cryptography using the Camenisch-Lysyanskaya (CL) signature scheme, which allows credential holders to preferentially disclose claims without correlation to verifiers.[57] When issuing a credential, an issuer creates a signature on a set of attributes for a given schema using a private key associated with their public DID through a DID document.

The following process shows the lifecycle of a VC with CL signature which enables the ZKP:

- The issuer defines the credential's schema, posts the credential definition which declares the intent to publish a credential from schema X, signed using key Y, and with revocation strategy Z.
- The issuer creates a DID and associates a public-private key pair with it. Then the issuer creates a DID document, signs it, and posts it to the distributed ledger.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

- The issuer assembles all the information it wants to enclose in the credential, including the information for each attribute articulated in the schema defined previously. The issuer prepares the credential by generating a numeric representation of each field and then signs both the numeric and the text representation of each of the statements using the CL Signature.
- The credential anchors a link secret that is known only to the holder (stored in the holder's wallet), and when the holder is issued a credential, it packages up a cryptographic commitment to the link secret within another long number that the issuer uses for the credential ID. The link secret is like a stamp used to create a watermark. Thus, all content in the certificate is very difficult to forge, proving that the holder possesses the stamp and is able to generate such a watermark.
- When the holder has the VC in their digital wallet, the holder can interact with a verifier and may want to prove a set of assertions made about a subject by a particular issuer. The holder receives a request from the verifier about what type of credential it is looking for.
- The holder performs specific calculations on the VC to get it ready to share in a proof presentation. The holder generates a new, never-before-seen credential wrapped inside a proof presentation that combines and reveals whatever attributes from issued credentials are requested, plus any predicates, and hides everything else. The 'proof' block of this new VC is a mathematical demonstration that the holder indeed possesses VCs signed by the requisite issuer, containing the disclosed attributes, and having the stipulated schema. The proof also demonstrates that the issuer has not revoked the credentials and that they are bound to the holder because the holder knows the link secret that was used at issuance.
- The verifier then takes the information it receives from the holder in the form of a proof presentation and does calculations with this information. It has to verify cryptographically that the proof is valid. The verifier resolves the DID of the issuer and finds its public key. It does a validation check on the presented attributes using the issuer's public key. This presentation may contain attributes from more than one credential. For each attribute shared, the verifier looks up the credential schema it came from, along with the DID/DID document of the issuer. It uses these two pieces of information to verify the attribute presentation. Each attribute statement in the proof presentation needs to go through this process. The verifier can be assured that all of the attributes were issued to the holder of the same link secret.

In SSI systems, the exchange of VCs is realized by digital wallets/agents through the DIDComm protocol.[58] The digital wallet consists of software that enables the wallet's controller to generate, store, manage, and protect cryptographic keys of VCs, and other sensitive private data, since with the digital wallet, the holder of VCs needs a software module to manage those interactions. This software module is defined as a digital agent, which enables a person to take actions, perform communications, store information, and track the usage of the digital wallet. Figure 4 shows the conceptual architecture of a typical SSI digital wallet and agent.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

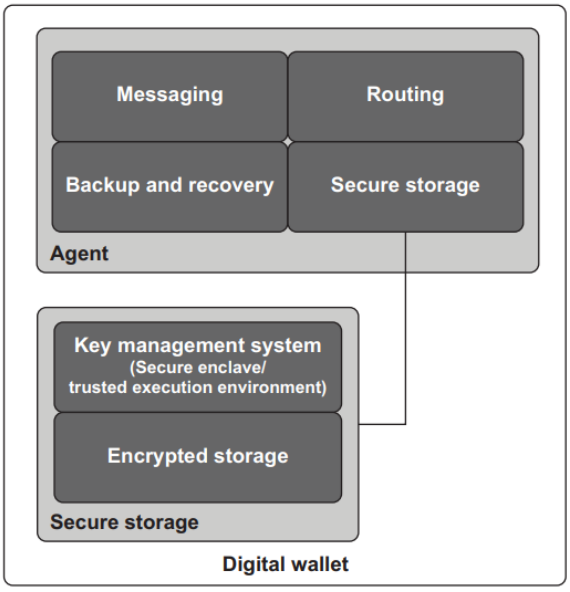


Fig. 4. Conceptual architecture of a typical SSI digital wallet and agent.[58]

3.4. Fully Homomorphic Encryption Scheme CKKS—As previously mentioned, fully homomorphic encryption is a kind of cryptography scheme that allows operations on encrypted data without decryption. The underlying principle of the CKKS scheme is to consider encryption noise to be a type of error that occurs during approximate computations.[45]

The figure below provides an overview view of the CKKS scheme. A message m which contains values for performing a certain computation is first encoded into a plaintext polynomial $p(X)$ and then encrypted using a public key. Once the message m is encrypted to ciphertext c (a couple of polynomials), CKKS scheme provides several operations that can be performed on it, including addition and multiplication. A composition of homomorphic operations is denoted as f. The decryption $c' = f(c)$ with the secret key will get the $p' = f(p)$ and then decoding to $m' = f(m)$. More technical details are described in a 2017 paper by Cheon.[45]

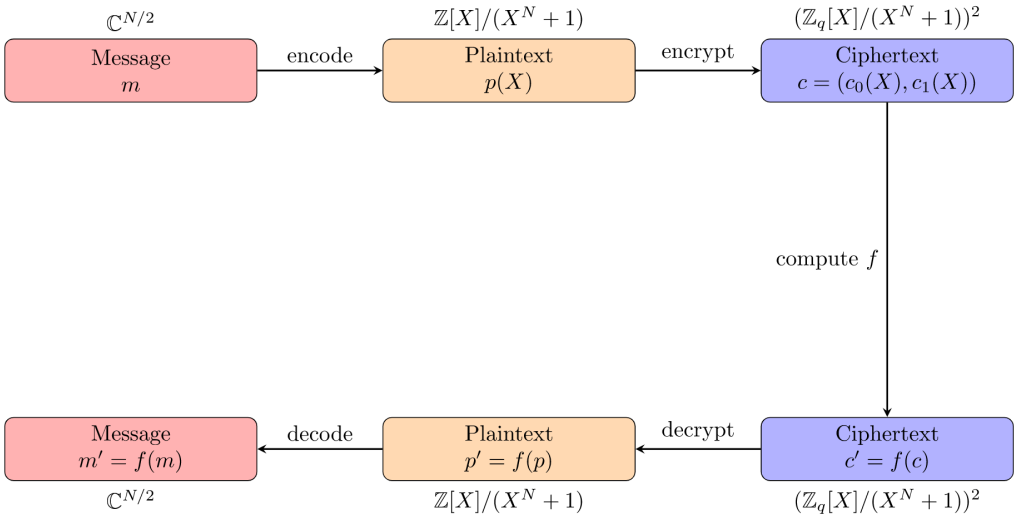


Fig. 5. Overview of CKKS Scheme.[59]

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

4. Decentralized Identity-Based Blockchain Solution

In this section, the decentralized identity-based blockchain solution for privacy-preserving licensing of individual-controlled data is presented. First, the requirements of the design are listed. Second, the roles involved in the design are introduced. Third, the systematic architecture is presented. Fourth, the process flows are analyzed.

4.1. Requirements—After surveying and reviewing data and identity protection approaches relevant to the study as discussed above, the specific technical requirements were established as below:

- Authentication: The design should be able to verify that an entity is who it claims to be, so that only legitimate users are allowed to access personal data and the service.
- Authorization: The design should be able to determine whether an authenticated user has access to particular resources and to what extent.
- Data Integrity: The design should be able to ensure the consistency of data validity over its life-cycle.
- Data Confidentiality: The design should protect personal data against disclosure, theft, and unintentional, unlawful, or unauthorized access.
- Privacy-preserving Data Processing: Personal data should be processed in a privacy-preserving environment which is either under the control of the users or ensures that no data can be leaked.
- Identity Privacy: The identity of the individual should never be leaked during the life-cycle.

4.2. Overview—Inspired and derived by the design of Gaber *et al.*,[4] five roles are incorporated into this particular solution design:

- Individual, a person who is the holder of personal data.
- Consumer, who requires limited use of individual data under authorized conditions.
- Data Issuer, the issuer of the individual's data credentials.
- Distributor, who issues license credentials to consumers.
- Data Manager, who keeps customer data securely, and authorizes the consumer to access the data.

In our decentralized solution architecture, there can be multiple instances of each role (*i.e.*, multiple participating issuers, individuals, distributors, data managers, and consumers).

The basic interactions are as follows: the data is issued to the individual by means of verifiable credentials, and the data is encrypted by AES and fully homomorphic encryption. The individual shares data by presenting a proof to the data storage manager. The storage manager stores encrypted data in the cloud, then issues a storage credential with the access token of the data. The individual sets up the data license to the distributor. The consumer purchases the data with the distributor. Figure 6 shows the overview of roles and process flows as a BPMN diagram.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

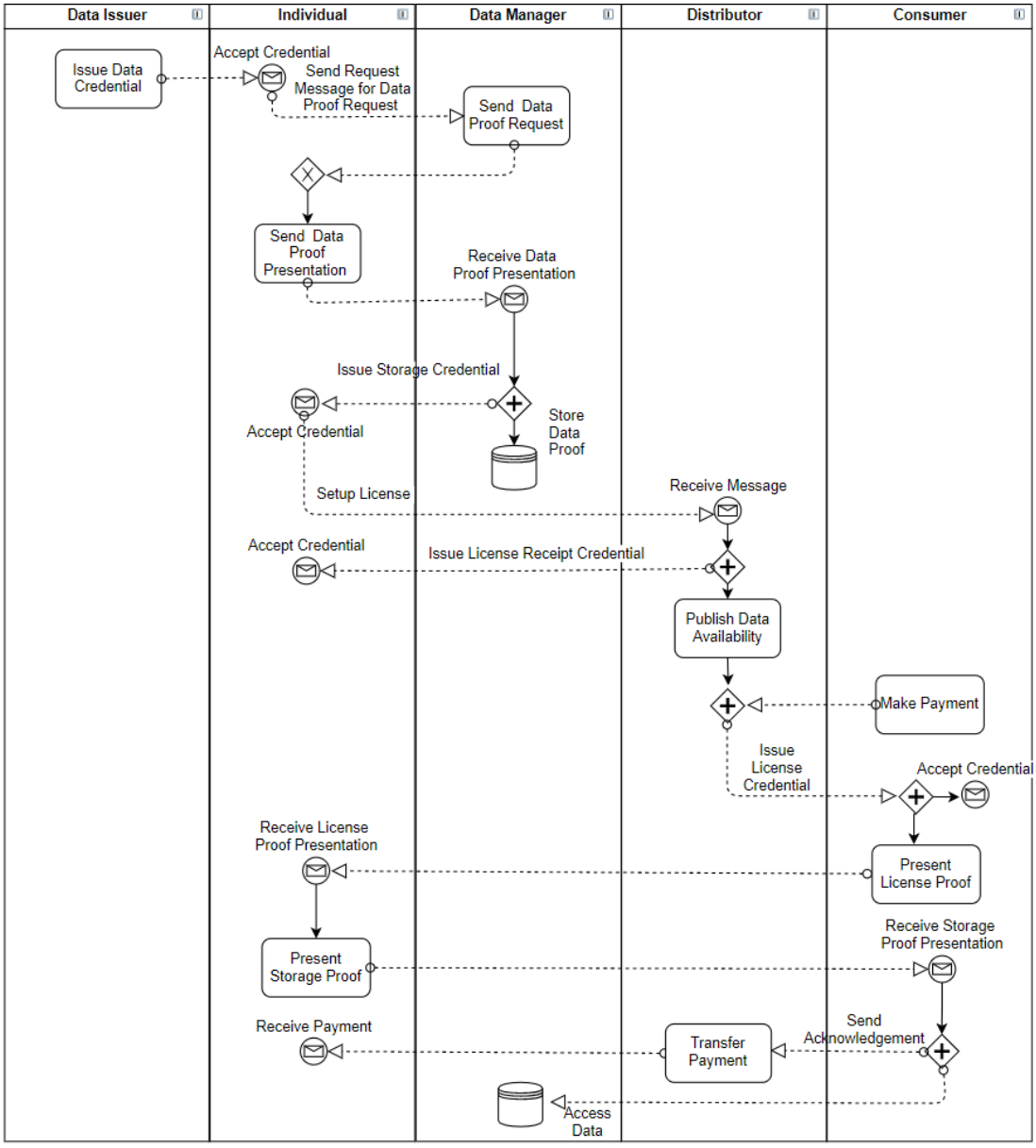


Fig. 6. Overview of roles and process flows.

4.3. System Architecture—For ease of presenting the design, the architecture is divided into two parts. First, the data sharing and storage architecture is presented. Second, the data licensing and consumption architecture is presented.

4.3.1. Data Sharing and Storage—Figure 7 shows the architecture of data sharing and storage. The data issuer and data manager use a web app that incorporates a Hyperledger cloud agent. The individual uses a mobile app with an agent. Agents can connect with other agents for messaging, credential issuing, and proof presentation.

To prevent the misuse of storage permissions, permissions should have a limited duration of validity. Data should never have a token that is valid indefinitely. This gives the actual controller temporal control over active permission tokens as well as authorization capabilities. The signed URL is a query string authentication that has been used by most cloud storage providers as part

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

of cloud storage access control mechanisms, which represents a concept of providing temporary access to specific resources.[60] All cloud service providers have an implementation of this technique, which is referred to as Signed URL in Google Cloud Storage, SAS in Microsoft Azure, and pre-signed URL in Amazon Web Services (AWS) for purposes of granting temporary access.[61,62] In this paper, we choose the AWS pre-signed URL. This URL is embedded as an attribute of the storage credential. Our design also considers the fact that the data manager is likely to need to provide a computing environment to further restrict data use; thus, the pre-signed url mentioned above should be more of an access token for the computing environment than just an access token for a data resource. Such an access token will provide the access of a virtual environment which contains a computing resource (*e.g.*, Amazon EC2) and the storage. That is, to prevent consumers from sharing homomorphically encrypted data, our design suggests that some level of protection and monitoring by the data manager as a service provider is required. These architectural enhancements require further research and involve the concept of a 'secure enclave.'

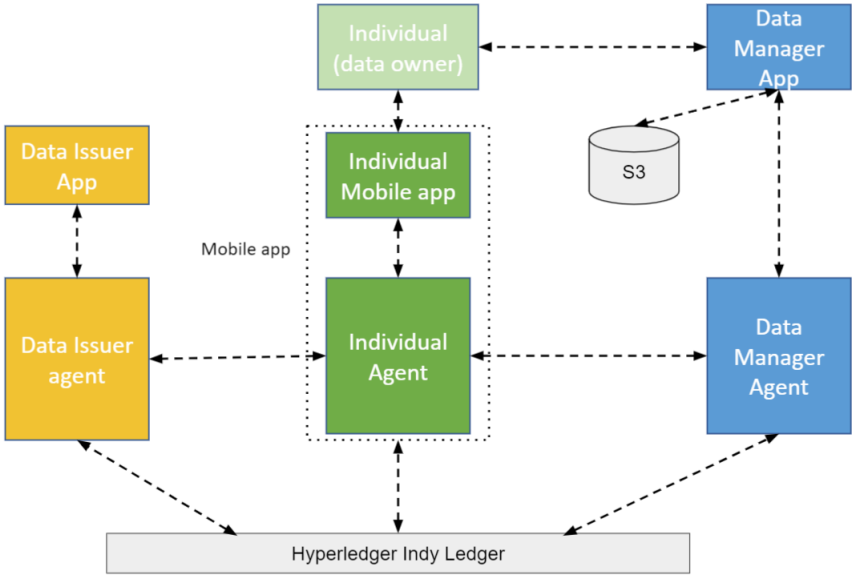


Fig. 7. Data issuing and storage architecture.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

The following algorithm shows the process of the data sharing and storage process.

Individual exchanges DIDs with the *Data Issuer* to establish a DIDComm connection channel.
Data Issuer uses the public key of the CKKS scheme (pk_{fhe}) to perform fully homomorphic encryption on the data of the individual. Then, *Data Issuer* uses the public key of the AES encryption scheme (pk_{aes}) to encrypt the data.
Data Issuer issues the data credential ($cred_{data}$) to the *Individual* with the (pk_{aes}) as an attribute of the credential.
Individual accepts and stores the health credential in his/her mobile wallet.
Individual exchanges DIDs with the *Data Manager* to establish a DIDComm connection channel through the generated invitation QR code on the *Data Manager* app.
Individual sends the schema for proof request to *Data Manager* through a message.
Data Manager generates proof request (req_{data}) according to the schema sent by the *Individual*, and sends req_{data}.
Individual presents the health data proof $proof_{data}$ based on $cred_{data}$.
Data Manager verifies the $proof_{data}$ by checking the signature and DID of the *Data Issuer*.
if $proof_{data}$ *has been verified* **then**
    Data Manager issues the initial storage credential ($cred_{storage_i}$) to the *Client* , with pre-signed URL and and pk_{aes} as attributes of $cred_{storage_i}$.
end

Algorithm 2: Data Sharing and Storage

4.3.2. Data Licensing and Consumption —Figure 8 shows the architecture of licensing and consumption. Similar to the previous part, the distributor and consumer have their web app with a Hyperledger cloud agent. The Individual uses a mobile app with an agent.

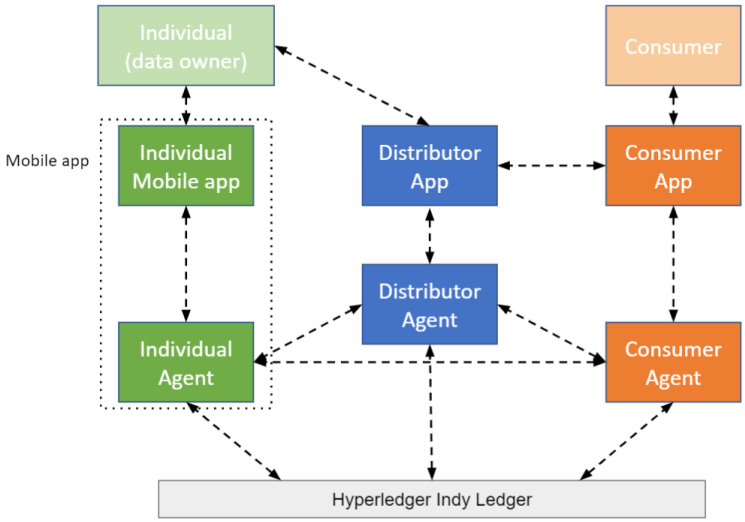


Fig. 8. Licensing architecture.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

The following algorithm shows the process of the data licensing and consumption process.

Individual exchanges DIDs with the *Distributor* to establish a DIDComm connection channel.
Individual sets up the license schema on distributor app.
Distributor generates license receipt credential ($cred_{receipt}$), and sends $cred_{receipt}$ to *Individual*.
while *Consumer selects the data with intention* **do**
    Consumer exchanges DIDs with the *Distributor* to establish a DIDComm connection channel.
    Consumer sends request for license credential ($cred_{license}$) to *Distributor* with the payment address.
    if *Distributor receives the payment* **then**
        Distributor adds the payment address to credential definition, issues $cred_{license}$ to *Consumer*.
        Individual exchanges DIDs with the *Consumer* to establish a DIDComm connection channel by scanning QR code on distributor app.
        Individual generates license proof request ($req_{license}$) and sends to *Consumer*.
        Consumer presents license proof ($proof_{license}$) to *Individual*.
        Individual verifies the $proof_{license}$ by checking the signature and DID of the *Distributor*.
        if $proof_{license}$ *has been verified* **then**
            Individual sends a request message for proof request of pre-signed URL (req_{url}) to *Consumer*.
            Consumer sends req_{url}, which asks for URL from $cred_{storage_i}$, and pk_{aes} from $cred_{data}$.
            if *pre-signed URL is expired* **then**
                Individual sends a message to *Data Manager* for a updated pre-signed URL credential ($cred_{storage_n}$).
                Data Manager issues client agent $cred_{storage_n}$.
            end
            Individual presents the URL credential $cred_{url}$ to *Consumer*.
            Consumer using pre-signed URL and pk_{aes} access health data. *Distributor* transfers the payment to the *Individual*
        end
    end
end

Algorithm 3: Data Licensing and Consumption

5. Implementation

We used the open source Python Aries Cloud Agent, Verifiable Organization Network(VON), which is the verifiable credential registry service, both of which were developed by the British

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

Columbia Government.[63] All cloud agents and apps are developed in the form of Docker containers. The TenSEAL library is adopted for doing homomorphic encryption operations on tensors,[64] which is built on top of Microsoft SEAL.[65] It provides ease of use through a Python API, while preserving efficiency by implementing most of the operations using C++. The Pytorch library is used to build the logistic regression model for computing over encrypted data.[66]

6. Evaluation

In this section, we evaluate the technical and workflow functionality of our implementation based on a scenario in which an individual controller of health data licenses the data for use to a health researcher. We base the data held by the individual on that available in the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database, and make the assumption that several individuals have been issued VCs containing data elements relating to themselves consistent with the data available in this database. We argue that this is not an unrealistic scenario in cases when a group of individuals has been issued VCs a part of a previous research study or clinical trial and now wishes to licence their data for use in a secondary use case.

6.1. Data Sharing and Storage—Figure 9 shows the health data credential and the storage credential in the individual's esatus mobile wallet (agent), issued by the data issuer and data manager respectively.[67]

The *AES_key* attribute of health data credential contains the key for the decryption of the *data_payload* , which will be shared with the consumer.

The *access_url* attribute of storage credential contains the AWS pre-signed URL for accessing the stored data payload in the S3 bucket for a limited time.

Fig. 9. The data credential and storage credential.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

6.2. License Setup and Purchase—Figure 10 shows the license receipt credential in the individual's wallet, issued by the distributor, as evidence that the data is posted on the market through the distributor.

Figure 11 shows the license credential stored in the consumer's MongoDB, issued by the distributor after the payment has been made. It is worth noting that the *payment_method* attribute in this case is a cryptocurrency method called 'Tether USD,' which is a kind of stablecoin pegged to the US dollar. The 'trc-20' means the coin is issued based on the TRON network using TRC20 protocol (which is compatible with ERC20 protocol of Ethereum).[68] This method is currently one of the most popular cryptocurrency payment methods.

Figure 12 shows the license proof presentation received by the individual. Since the esatus wallet mobile app does not have the developed function to request proof, this process is achieved using a Hyperledger Aries cloud agent instead.

Figure 13 shows the storage proof presentation received by the consumer, which contains the requested attributes from both the storage credential and the health data credential controlled by the individual. The consumer could access the stored data via the *access_url*, and decrypt with *AES_key*.

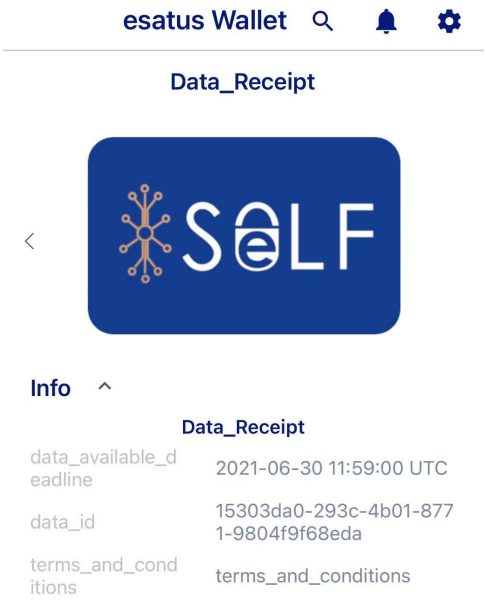


Fig. 10. The license receipt credential.

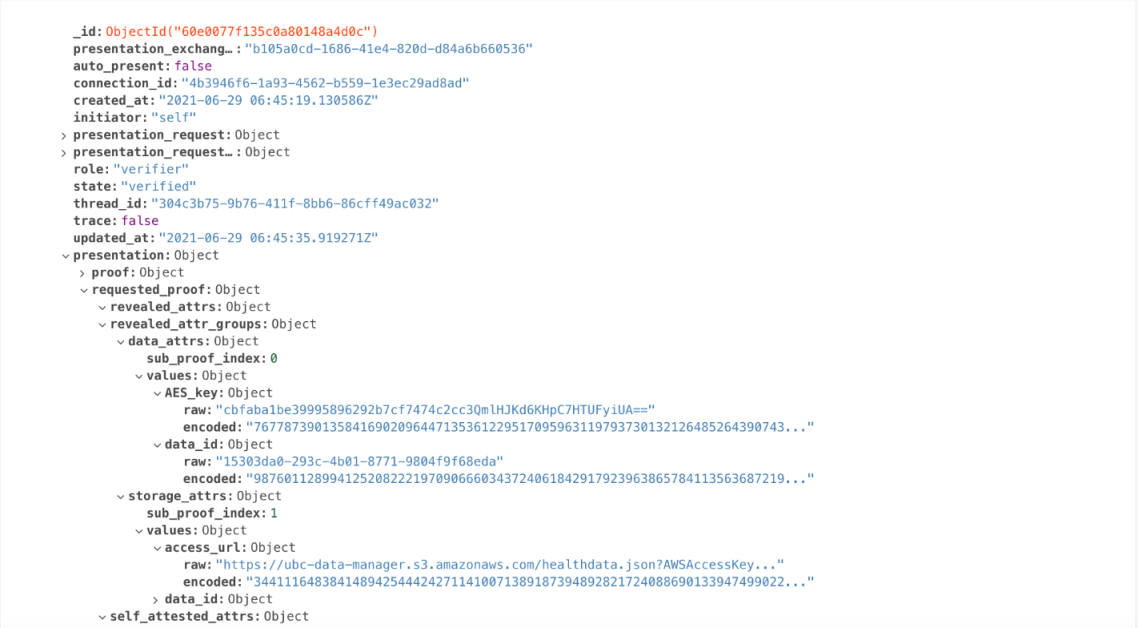
6.3. Data Consumption—The Molecular Taxonomy of Breast Cancer International Consortium (METABRIC) database is used in the evaluation of the FHE technical functionality of our solution. The purpose for the creation of this dataset is to predict the survival of breast cancer patients from the clinical and genomic data. The dataset has 1309 samples with 582 features. For the consideration of practical conditions, we assume that a consumer has purchased data from many individuals separately to obtain this homomorphically-encrypted dataset. The computation is performed on a computer with an Intel CPU i7 9700k @ 4.70 GHz, 32 GB RAM, and an Nvidia RTX 2070 Super GPU with 8 GB of memory.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

Fig. 11. The license credential.

Fig. 12. The license proof presentation.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239



```
_id: ObjectId("60e0077f135c0a80148a4d0c")
presentation_exchang… : "b105a0cd-1686-41e4-820d-d84a6b660536"
auto_present: false
connection_id: "4b3946f6-1a93-4562-b559-1e3ec29ad8ad"
created_at: "2021-06-29 06:45:19.130586Z"
initiator: "self"
› presentation_request: Object
› presentation_request… : Object
role: "verifier"
state: "verified"
thread_id: "304c3b75-9b76-411f-8bb6-86cff49ac032"
trace: false
updated_at: "2021-06-29 06:45:35.919271Z"
⌄ presentation: Object
  › proof: Object
  ⌄ requested_proof: Object
    ⌄ revealed_attrs: Object
    ⌄ revealed_attr_groups: Object
      ⌄ data_attrs: Object
          sub_proof_index: 0
        ⌄ values: Object
          ⌄ AES_key: Object
              raw: "cbfaba1be39995896292b7cf7474c2cc3QmlHJKd6KHpC7HTUFyiUA=="
              encoded: "767787390135841690209644713536122951709596311979373013212648526439074 3..."
          ⌄ data_id: Object
              raw: "15303da0-293c-4b01-8771-9804f9f68eda"
              encoded: "98760112899412520822219709066603437240618429179239638657841135636872 19..."
      ⌄ storage_attrs: Object
          sub_proof_index: 1
        ⌄ values: Object
          ⌄ access_url: Object
              raw: "https://ubc-data-manager.s3.amazonaws.com/healthdata.json?AWSAccessKey..."
              encoded: "3441116483841489425444242711410071389187394892821724088690133947499022..."
          › data_id: Object
    ⌄ self_attested_attrs: Object
```


Fig. 13. The storage proof presentation.

According to Kim *et al.*, the logistic regression is adopted here as an evaluation model.[69] A set of low-degree polynomials is used to approximate the sigmoid function in a bound range around zero, which is called Least Square Approximation (LSA). Since existing homomorphic encryption schemes only allow the evaluation of polynomial functions, a degree-3 approximate polynomial $\sigma(x) = 0.5 - 0.1167694x + 0.0008352x^3$ of sigmoid on the interval $[-20, 20]$ is adopted here.

In addition, in order to prove the effectiveness of the test method, we also compared it on the unencrypted dataset. 5-fold cross-validation (CV) is used: the dataset is randomly partitioned into five folds of approximately equal size, with each subset of four folds being used for training and the remaining one being used for testing the model.

Table 1 shows the performance of using logistic regression on plain-text and encrypted data, respectively, indicating comparable results. It took 3 minutes to train a logistic regression model on the plain-text, compared to 105 minutes on the homomorphic encrypted data. Thus, its efficiency in broader applications remains an open question.

Table 1. Results for METABRIC dataset with 5-fold CV

Method	Accuracy	AUC
Plain-text	0.752	0.746
Encrypted	0.740	0.719

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

7. Conclusion

In this paper, we contribute a novel design and implementation that:

- utilizes a decentralized identity-based blockchain solution to enable the licensing, sharing, storage, and consumption of individual-controlled data.
- utilizes homomorphic encryption to preserve privacy and help prevent unauthorized secondary use of individual-controlled data.

Though our novel design advances research aimed at preventing unauthorized secondary use of the individual's data, we cannot yet attest to the practicality of the solution in a real-world implementation. Processing efficiency, for example, remains a major barrier to adoption. We anticipate that another limitation of our approach is that it will be difficult to apply it to digital data that is not distributed by platforms or large organizations, which still control the modalities of data production and distribution. Thus, one future line of research will aim to test the usability of the solution in a real-world pilot. In addition, the solution currently only controls usage based on temporal considerations. In future work, we will aim to introduce other user-configured usage controls, such as type of organization, geographic restrictions, and so on. Our goal will be to gradually tip the balance of power regarding the control of data usage towards the individual and away from platforms or organizations in order to prevent unauthorized secondary use of personal data.

Acknowledgements

The authors would like to thank the two anonymous reviewers of this paper and acknowledge the financial support for this research of The Natural Sciences and Engineering Research Council of Canada (NSERC).

Notes and References

[1] Amer, K., Noujain, J. "The Great Hack." *Netflix* (2019) https://www.netflix.com/ca/title/80117542.

[2] Voigt, P., Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR), A Practical Guide*. Cham: Springer International Publishing (2017).

[3] "Health Insurance Portability and Accountability Act of 1996, Pub. L. No. 104-191." *104th Congress of the United States of America* (accessed 18 November 2021) https://www.congress.gov/104/plaws/publ191/PLAW-104publ191.pdf.

[4] Gaber, T., Ahmed, A., Mostafa, A. "PrivDRM: A Privacy-Preserving Secure Digital Right Management System." In *Proceedings of the Evaluation and Assessment in Software Engineering* 481–486 (2020) https://doi.org/10.1145/3383219.3383289.

[5] Belchior, R., Putz, B., Pernul, G., Correia, M., Vasconcelos, A., Guerreiro, S. "SSIBAC: Self-Sovereign Identity Based Access Control." In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* 1935–1943 (2020) https://doi.org/10.1109/TrustCom50675.2020.00264.

[6] Liu, Y., Lu, Q., Zhu, C., Yu, Q. "A Blockchain-Based Platform Architecture for Multimedia Data Management." *Multimedia Tools and Applications* **80** 30707–30723 (2021) https://doi.org/10.1007/s11042-021-10558-z.

[7] Papadopoulos, P., Abramson, W., Hall, A. J., Pitropakis, N., Buchanan, W. J. "Privacy and Trust Redefined in Federated Machine Learning." *Machine Learning and Knowledge Extraction* **3.2** 333–356 (2021) https://doi.org/10.3390/make3020017.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

[8] Lemieux, V. L., *et al.* "Having Our "Omic" Cake and Eating It Too?: Evaluating User Response to Using Blockchain Technology for Private and Secure Health Data Management and Sharing." *Frontiers in Blockchain* **3** 558705 (2021) https://doi.org/10.3389/fbloc.2020.558705.

[9] Nagaratnam, N. "What is Confidential Computing?" *IBM* (2020) https://www.ibm.com/cloud/learn/confidential-computing.

[10] Gentry, C. "Fully Homomorphic Encryption Using Ideal Lattices." In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* 169–178 (2009) https://doi.org/10.1145/1536414.1536440.

[11] Cavoukian, A. "Privacy by Design: The 7 Foundational Principles." *Information and Privacy Commissioner of Ontario* (2011) (accessed 18 November 2021) https://www.ipc.on.ca/wp-content/uploads/Resources/7foundationalprinciples.pdf.

[12] Yaga, D., Mell, P., Roby, N., Scarfone, K. "Blockchain Technology Overview." *arXiv* (accessed 18 November 2021) https://doi.org/10.6028/NIST.IR.8202.

[13] Kazdin, A. E. "The Token Economy." In *Applications of Conditioning Theory.* Routledge (1981, 2017) http://doi.org/10.4324/9781351273084.

[14] Au, S., Power, T. *Tokenomics: The Crypto Shift of Blockchains, ICOs, and Tokens*. Birmingham: Packt Publishing Ltd. (2018).

[15] "Investing in Verified Information." *University of Washington APL and Digital ID and Authentication Council of Canada (DIACC)* (academic conference, held 6-7 November 2019 in Seattle, Washington) https://depts.washington.edu/uwconf/wordpress/ivi2019/.

[16] Nakamoto, S. "Bitcoin: A Peer-to-Peer Electronic Cash System." (2008) (accessed 18 October 2021) https://bitcoin.org/bitcoin.pdf.

[17] Buterin, V. "A Next-Generation Smart Contract and Decentralized Application Platform." *Ethereum.org* (2014) (accessed 18 November 2021) https://ethereum.org/en/whitepaper/.

[18] Allen, C. "The Path to Self-Sovereign Identity." *Life with Alacrity* (2016) (accessed 18 November 2021) http://www.lifewithalacrity.com/previous/2016/04/the-path-to-self-soverereign-identity.html.

[19] Tobin, A., Reed, D. "The Inevitable Rise of Self-Sovereign Identity." *The Sovrin Foundation* (2017) (accessed 18 November 2021) https://sovrin.org/wp-content/uploads/2017/06/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf.

[20] No Author. "Hyperledger Indy." *Hyperledger* (accessed 22 March 2021) https://www.hyperledger.org/use/hyperledger-indy.

[21] Tobin, A. "Sovrin: What Goes on the Ledger?" *Evernym* (2018) (accessed 18 November 2021) https://www.evernym.com/wp-content/uploads/2017/07/What-Goes-On-The-Ledger.pdf.

[22] Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C. "A Survey on Essential Components of a Self-Sovereign Identity." *Computer Science Review* **30** 80–86 (2018) http://dx.doi.org/10.1016/j.cosrev.2018.10.002.

[23] Aydar, M., Ayvaz, S. "Towards a Blockchain Based Digital Identity Verification, Record Attestation and Record Sharing System." *arXiv* (accessed 18 November 2021) https://arxiv.org/abs/1906.09791v2.

[24] C. Lundkvist, R. Heck, J. Torstensson, Z. Mitton, and M. Sena. "Uport: A Platform for Self-Sovereign Identity." (2018) (accessed 18 November 2021) https://www.uport.me/. [Editor's note: As of the date of publication, Uport has split into two projects: Serto (http://www.serto.id/) and Veramo (https://veramo.io/)].

[25] Naik, N., Jenkins, P. "Your Identity Is Yours: Take Back Control of Your Identity Using GDPR Compatible Self-Sovereign Identity." In *2020 7th International Conference on Behavioural and Social Computing (BESC)* IEEE 1–6 (2020) https://doi.org/10.1109/BESC51023.2020.9348298.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

[26] Naik, N., Jenkins, P. "Governing Principles of Self-Sovereign Identity Applied to Blockchain Enabled Privacy Preserving Identity Management Systems." In *2020 IEEE International Symposium on Systems Engineering (ISSE)* IEEE 1–6 (2020) https://doi.org/10.1109/ISSE49799.2020.9272212.

[27] Hofman, D., Lemieux, V. L., Joo, A., Batista, D. A. "The Margin Between the Edge of the World and Infinite Possibility: Blockchain, GDPR, and Information Governance." *Records Management Journal* **29.1/2** 240–257 (2019) https://doi.org/10.1108/RMJ-12-2018-0045.

[28] No Author. "Starting January 1, Businesses Must Follow More Robust Guidelines on Meaningful Consent for Personal Information." *Office of the Privacy Commisioner of Canada* (2018) (accessed 20 November 2021) https://www.priv.gc.ca/en/opc-news/news-and-announcements/2018/an_181221/.

[29] Romm, T. "U.S. Government Issues Stunning Rebuke, Historic $5 Billion Fine Against Facebook for Repeated Privacy Violations." *The Washington Post* (2019) (accessed 20 November 2021) https://www.washingtonpost.com/technology/2019/07/24/us-government-issues-stunning-rebuke-historic-billion-fine-against-facebook-repeated-privacy-violations/.

[30] Rosenblatt, B., Trippe, B., Mooney, S. *Digital Rights Management*. New York: Wiley (2002).

[31] Pretschner, A., Hilty, M., Basin, D. "Distributed Usage Control." *Communications of the ACM* **49.9** 39–44 (2006) https://doi.org/10.1145/1151030.1151053.

[32] Russinovich, M., *et al.* "Toward Confidential Cloud Computing." *Communications of the ACM* **64.6** 54–61 (2021) https://doi.org/10.1145/3454122.3456125.

[33] Bonawitz, K., *et al.* "Towards Federated Learning at Scale: System design." *arXiv* (2019) (accessed 20 November 2021) https://arxiv.org/abs/1902.01046v2.

[34] Sabt, M., Achemlal, M., Bouabdallah, A. "Trusted Execution Environment: What It Is, and What It Is Not." In *2015 IEEE Trustcom/BigDataSE/ISPA*. **1** IEEE 57–64 (2015) https://doi.org/10.1109/Trustcom.2015.357.

[35] No Author. "Intel Software Guard Extensions." (accessed 20 November 2021) https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions.html.

[36] No Author. "TRUSTZONE." *ARM* (accessed 20 November 2021) https://www.arm.com/why-arm/technologies/trustzone-for-cortex-a.

[37] Goldreich, O. "Secure Multi-Party Computation." (1998, 2002) Preliminary manuscript (accessed 20 November 2021) https://www.wisdom.weizmann.ac.il/~oded/pp.html.

[38] Damgard, I., Geisler, M., Kroigard, M. "Homomorphic Encryption and Secure Comparison." *International Journal of Applied Cryptography* **1.1** 22–31 (2008) https://doi.org/10.1504/IJACT.2008.017048.

[39] Beimel, A. "Secret-Sharing Schemes: A Survey." In *International Conference on Coding and Cryptology* Springer 11–46 (2011) https://doi.org/10.1007/978-3-642-20901-7_2.

[40] Yao, A. C.-C. "How to Generate and Exchange Secrets." In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)* IEEE 162–167 (1986) https://doi.org/10.1109/SFCS.1986.25.

[41] Diffie, W., Hellman, M. "New Directions in Cryptography." *IEEE Transactions on Information Theory* **22.6** 644–654 (1976) https://doi.org/10.1109/TIT.1976.1055638.

[42] Rivest, R. L., Shamir, A., Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." *Communications of the ACM* **21.2** 120–126 (1978) https://doi.org/10.1145/359340.359342.

[43] Rivest, R. L., Adleman, L., Dertouzos, M. L., *et al.* "On Data Banks and Privacy Homomorphisms." *Foundations of Secure Computation* **4.11** 169–180 (1978).

[44] ElGamal, T. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." *IEEE Transactions on Information Theory* **31.4** 469–472 (1985) https://doi.org/10.1007/3-540-39568-7_2.

[45] Cheon, J. H., Kim, A., Kim, M., Song, Y. "Homomorphic Encryption for Arithmetic of Approximate Numbers." In *International Conference on the Theory and Application of Cryptology and Information Security* Springer 409–437 (2017) https://doi.org/10.1007/978-3-319-70694-8_15.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

[46] Cheon, J. H., *et al.* "Toward a Secure Drone System: Flying with Real-Time Homomorphic Authenticated Encryption." *IEEE Access* **6** 24325–24339 (2018) https://doi.org/10.1109/ACCESS.2018.2819189.

[47] Chen, H., *et al.* "Logistic Regression Over Encrypted Data from Fully Homomorphic Encryption." *BMC Medical Genomics* **11.4** 81 (2018) https://doi.org/10.1186/s12920-018-0397-z.

[48] Kim, A., Song, Y., Kim, M., Lee, K., Cheon, J. H. "Logistic Regression Model Training Based on the Approximate Homomorphic Encryption." *BMC Medical Genomics* **11.4** 23–31 (2018) https://doi.org/10.1186/s12920-018-0401-7.

[49] Kim, D., Son, Y., Kim, D., Kim, A., Hong, S., Cheon, J. H. "Privacy-Preserving Approximate GWAS Computation Based on Homomorphic Encryption." *BMC Medical Genomics* **13.7** 1–12 (2020) https://doi.org/10.1186/s12920-020-0722-1.

[50] Kim, M., Song, Y., Li, B., Micciancio, D. "Semi-Parallel Logistic Regression for GWAS on Encrypted Data." *BMC Medical Genomics* **13.7** 1–13 (2020) https://doi.org/10.1186/s12920-020-0724-z.

[51] Blatt, M., Gusev, A., Polyakov, Y., Goldwasser, S. "Secure Large-Scale Genome-Wide Association Studies Using Homomorphic Encryption." *Proceedings of the National Academy of Sciences* **117.21** 11608–11613 (2020) https://doi.org/10.1073/pnas.1918257117.

[52] Reed, D., *et al.* "Decentralized Identifiers (dids) v1.0." *World Wide Web Consortium (W3C)* (2020) (accessed 20 November 2021) Latest draft available at: https://www.w3.org/TR/did-core/.

[53] Saint-Andre, P., Klensin, J. "Uniform Resource Names (URNs)." *Internet Engineering Task Force (IETF)* (2017) Internet Requests for Comments 8141 (accessed 20 November 2021) https://www.rfc-editor.org/rfc/rfc8141.html.

[54] Hardman, D., *et al.* "Peer DID Method Specification." *GitHub* (2019) (accessed 20 November 2021) https://github.com/decentralized-identity/peer-did-method-spec.

[55] Hardman, D. "Hyperledger Aries RFC 0005: DID Communication." *GitHub* (2019) (accessed 20 November 2021) https://github.com/hyperledger/aries-rfcs/blob/master/concepts/0005-didcomm/.

[56] Verifiable Credentials Working Group. "Verifiable Credentials Data Model 1.0: Expressing Verifiable Information on the Web." *World Wide Web Consortium (W3C)* (2019) (accessed 23 February 2021) https://www.w3.org/TR/vc-data-model/.

[57] Camenisch, J., Lysyanskaya, A. "A Signature Scheme with Efficient Protocols." In *Security in Communication Networks, SCN 2002* 268–289 (2003) https://doi.org/10.1007/3-540-36413-7_20.

[58] Preukschat, A., Reed, D. *Self-Sovereign Identity: Decentralized Digital Identity and Verifiable Credentials*. New York: Manning Publications Co. (2021).

[59] Huynh, D. "CKKS Explained: Part I, Vanilla Encoding and Decoding." *OpenMined* (accessed 14 May 2021) https://blog.openmined.org/ckks-explained-part-1-simple-encoding-and-decoding/.

[60] Graupner, H., Torkura, K., Berger, P., Meinel, C., Schnjakin, M. "Secure Access Control for Multi-Cloud Resources." In *2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops)* IEEE 722–729 (2015) https://doi.org/10.1109/LCNW.2015.7365920.

[61] "V4 Signing Process with Your Own Program." *Google Cloud* (2021) (accessed 29 March 2021) https://cloud.google.com/storage/docs/access-control/signing-urls-manually.

[62] "Sharing an Object with a Presigned URL." *AWS* (2021) (accessed 29 March 2021) https://docs.aws.amazon.com/AmazonS3/latest/userguide/ShareObjectPreSignedURL.html.

[63] Digital Trust Team, Government of British Columbia, *et al.* "Hyperledger Aries Cloud Agent Python (ACA-Py)." *GitHub* (accessed 23 February 2021) https://github.com/hyperledger/aries-cloudagent-python.

[64] Benaissa, A., *et al.* "TenSEAL: A Library for Doing Homomorphic Encryption Operations on Tensors." *GitHub* (2020) (accessed 20 November 2021) https://github.com/OpenMined/TenSEAL/.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239

[65] Chen, H., Laine, K., Player, R. "Simple encrypted arithmetic library-SEAL v2. 1." In *International Conference on Financial Cryptography and Data Security, FC2017* Springer 3–18 (2017) https://doi.org/10.1007/978-3-319-70278-0_1.

[66] Paszke, A., *et al.* "Pytorch: An Imperative Style, High-Performance Deep Learning Library." *arXiv* (2019) (accessed 20 November 2021) https://arxiv.org/abs/1912.01703v1.

[67] "esatus SSI Wallet." (2020) (accessed 6 May 2020) https://web.archive.org/web/20210506165649/https://esatus.com/esatus-ssi-wallet-app-ab-sofort-fuer-ios-und-android-verfuegbar/?lang=en [Editor's note: the esatus SSI wallet branded "SeLF" has at the time of publication been rebranded "esatus SOWL," rendering the previous link deprecated. A Wayback Machine link has been substituted. For more information on esatus SOWL, see: https://esatus.com/solutions/self-sovereign-identity/sowl/?lang=en].

[68] No Author. "TRON Advanced Decentralized Blockchain Platform, Whitepaper v. 2.0, TRON Protocol v. 3.2." *TRON Foundation* (2018) (accessed 20 November 2021) https://tron.network/static/doc/white_paper_v_2_0.pdf.

[69] Kim, M., Song, Y., Wang, S., Xia, Y., Jiang, X. "Secure Logistic Regression Based on Homomorphic Encryption: Design and Evaluation." *JMIR Medical Informatics* **6.2** e19 (2018) https://doi.org/10.2196/medinform.8805.

SSN 2379-5980 (online)
DOI 10.5195/LEDGER.2021.239