Development of a National Identity Management System using Blockchain Technology

Oluyemi Amujo\textsuperscript{1}, Christopher U. Ebelogu\textsuperscript{2}, Edward O. Agu\textsuperscript{3} and M. B. Hammawa\textsuperscript{4}

\textsuperscript{1,2,4}Department of Computer Science, University of Abuja, Abuja, Nigeria.
\textsuperscript{3}Department of Computer Science, Federal University Wukari, Taraba State, Nigeria.

Email:\textsuperscript{1} oluyemiamujo@gmail.com, \textsuperscript{2} christopher@uniabuja.edu.ng, \textsuperscript{3} aguedward@fuwukari.edu.ng, \textsuperscript{4} mohammed.hammawa@uniabuja.edu.ng

**ABSTRACT**

Nigeria’s national identity systems that exist today, which were developed on centralized data management system, may soon present operational risks for citizens and government agencies due to single points of trust and failure which make the system vulnerable to attack. Decentralized identity management system is a solution to address this challenge, which specifies a user-focused approach that gives full control of identity back to the individual. This paper proposes the blockchain, which is a secure and decentralized system, as the platform to achieve this. Taking Nigerian’s National Identity Management System as a case study, a blockchain-based identity management system is proposed. Models for identity creation, modification, authentication and disclosure are presented. Prevention of Sybil attack and developing of handshake protocol model for identity attribute disclosure are other contributions. Lastly, a prototype of the proposed system is developed in Python programming language to demonstrate the salient features that meet the research objectives.

**Keywords:** Identity management, Blockchain, authentication, Privacy, Security, Decentralization.

---

**African Journal of Computing & ICT Reference Format:**


© Afr. J. Comp. & ICT, December 2019; ISSN 2006-1781
I. INTRODUCTION

Identity management refers broadly to the administration of individual identities within a system, such as a company, a network or even a country. An identity system is necessary for business and social affairs [1]. Having a reliable national identification system and citizens register is a good compliment for national planning, election, census board, law enforcement, banking, tax collection, pension board, education and in fact the judiciary [2]. Without functional identification systems, citizens of many developing countries miss out on the benefits of official identification.

For most countries, a government-issued national identity card is the main program that is being used for both identification verification and for at least one functional authentication purpose. In five of the countries reviewed in [3], including Burkina Faso, Cambodia, Nigeria, Ukraine, and Zambia, there is more than one prominent national identity program. In addition to their national identity card programs, Burkina Faso and Zambia have implemented voter’s card programs even before Nigeria. Furthermore, among other nations, Estonia and Finland have been recognized for the success achieved in developing e-government tools. Estonian e-Residency system now creates a new digital nation for global citizens [36]. Every person can provide digital signatures using their ID-card, Mobile-ID or Smart-ID, to safely identify themselves and use e-services. The system can be used for company registration, banking, payment services and document signing.

Technological innovations have opened up new possibilities for governments to develop comprehensive identity management systems that link peoples’ identities through their entire life, from birth certificate, civil registration, driver’s license, to marriage certificate, voter registration and national identity card [3]. At the same time, governments in developing countries are expected to carry out many of the same functions that richer countries are capable of performing; these functions include “providing universal access to healthcare and education, implementing know your customer (KYC) rules for financial institutions, and administering a wide variety of transfer programs” [4]. However, the identity being managed by system constitutes a uniqueness of an individual and has attracted security concerns. Much sensitive personal information is often misused, somehow duplicated or leaked and financial assets are hacked etc. These security events directly or indirectly cause the economic losses for the affected country. Thus, how to manage identity becomes an important problem for people in industry and academic settings [5].

Many efforts have been sought to find effective approaches in protecting national identity data [3, 6] such as centralized server which was adopted in developing National Identity management system in Nigeria [7]. However, these approaches still center on centralized data management system associated with single-point-of-failure problem which makes the system vulnerable to attack as it is possible for adversary to achieve its malicious goals of stealing/misusing/manipulating these data at the centralized server.

The ideal solution would be a decentralized system that eliminates a single-point of failure and secure the information in such a manner that it is not only difficult to hijack the system but also impossible to manipulate information without the key.

Blockchain technology offers a decentralized transaction and secured data management and eliminates those failures attributed to centralized data management system. Blockchain was first developed for data management of Bitcoin cryptocurrency when the idea was coined in 2008 by a group called Satoshi as motivated by several literatures [13, 14, 15, 16]. A blockchain is not a currency but a system for validating, clearing, settling, tracking and recording the ownership of assets as they are traded [17]. Because of the popularity and fast development of Bitcoin, the technology that supports Bitcoin begins to take the public attention. In other words, the Bitcoin starts a new era for the Blockchain technology [18], where it is possible to create and transfer values without trusted medium on the Internet [19]. Therefore, the blockchain concept can be defined regardless of the Bitcoin ecosystem.

Blockchain development and applications have been identified as Blockchain 1, 2 and 3 (BL1-2-3) [20]. Blockchain 1 refers to the initial applications to currencies; Blockchain 2 refers to contracts while Blockchain 3 concerns applications to further legal and economic aspects. More specifically, BL3 includes value attestation services, notary services, identity and property verification, intellectual property rights protection and others.

This paper proposes a model for secured national identity management system that is based on blockchain technology, taking identification management system for Nigeria as a case study. The model covers authentication, verification and disclosure of user identity. Also, techniques that prevent Sybil attack is proposed by binding the user’s entity information with the UUID address and determine the true identity of a virtual user on the Blockchain. Furthermore, handshake protocol model is proposed for identity disclosure that allows user to decide which attribute of the identity to disclose to another user on the blockchain. Lastly, a prototype of the proposed system is developed in Python programming language. The prototype implemented those models which are salient features for the proposed national identity management system.
II. LITERATURE REVIEW

2.1 Possible Problems of Identity Management Based on Centralized Database Model

Identity Management (IdM) encompasses the processes and policies involved in managing the lifecycle of attributes in identities for a particular domain [8]. The general objectives of identity management are described in a white paper by [9] with the following features:

- ownership of identity: each person wants to be responsible for his or her own identity
- privacy: most individuals want to keep personal information private, and to restrict access to it to a few known other people
- efficiency: individuals want to maintain their identities in as few places as possible, yet have those identities recognized and accepted by the different IT systems that they use everywhere
- reputation: this is information used to make a value judgment about an object or a person [10]
- personalized services: users want services to react to their specific needs, and are becoming disenchanted with those that do not, to cater for this, systems are becoming personalized, event-driven, and real-time.

Conventional identity management systems are based on centralized authorities such as corporate directory services, certificate authorities, or domain name registries. From the standpoint of cryptographic trust verification, each of these centralized authorities serves as its own root of trust. To make identity management work across these systems requires implementing federated identity management [30]. Consequently, most IdM schemes today are centralized where a single entity controls the system. Thus the centralized database system becomes the security bottleneck of an identity management system. It is always subject to the attacking hackers, system downtime, unavoidable software license fees and upgrades, as well as hardware limitations and network traffic restrictions [11]. Therefore, personal identity information is often leaked or even tampered with.

Identity management on central server is currently the predominant technique both in public and private systems. Figure 2.1 shows the one proposed in [12] for National Identity management system in Nigeria. The system consists of a central server which functions as the processing system that stores data from other four major components such as (a) Front-end Enrolment System that captures registrant’s data send the data to the central server and send confirmation back to registrant. (b) Automated Fingerprint Identification System (AFIS) which is a biometric identification methodology that uses digital imaging technology to obtain, store, and analyze fingerprint data (c) Card Personalization System that enables designing of identity card to be tailored to individual user (d) Web-Based Verification System that enables identity verification to be integrated on web application.

Furthermore, a conceptual design of a low cost management system for Nigeria was proposed in [6] as illustrated in Figure 2.2; however, those systems can still suffer from failure. This is owned to single point of trust and failure problem, system hijacking and DOS problem inherent in centralization of data and resources.

2.2 Overview of Blockchain Architecture

As illustrated in Figure 2.3, the blockchain data structure is a back-linked list of blocks of transactions, which is ordered. It can be stored as a flat file or in a simple database. Each block is identifiable by a hash, generated using the SHA256 cryptographic hash algorithm on the header of the block. Each block references a previous block, also known as the parent block, in the “previous block hash” field, in the block header [34].

A hash, also known as cryptographic hash functions are mathematical algorithms or one-way functions that take an input and transform it into an output of specific length, e.g. a series of 256 bits, called the hash output [32]. The idea behind a hash functions use is to facilitate a thorough means for searching for data in a dataset. Hash functions are collision-free too. That means it’s impossible to find two messages that hash to the same hash value [33].

A block is a container data structure, which brings together transactions for inclusion in the public ledger, known as the blockchain. The block is made up of a header; containing metadata, followed by a long list of transactions. A block can be identified in two ways, either by referencing the block hash, or through referencing the block height through its position in the blockchain.

Blockchains run on digital networks. Data transmission in such networks is equivalent to copying data from one place to the other, e.g. in the cryptocurrency domain this is equivalent to copying digital coins from one user’s electronic wallet to another’s. The principal challenge is how to ensure that coins are only spent once and there is no double-spending. A traditional solution is to use a central point of authority, such as a central bank, who acts as the trusted intermediary between
transacting parties and whose job is to store, safeguard the valid state of the ledger and keep the records up to date. If multiple parties need to write in the ledger at the same time, a central authority also implements concurrency control and consolidates changes in the ledger [31]. To buttress the point, Figure 2.4 shows the visual representation of a blockchain transaction. Users agree on a transaction which is included in a block, its validity is confirmed by distributed nodes of the network and the block is added to the growing chain of blocks before transaction is confirmed and payments finalized.

The reason for the interest in Blockchain is its attributes that provide security, anonymity and data integrity without any third party organization in control of the transactions [21]. These features make it become a potentially ideal solution for authenticating and protecting identity management system, among other uses. In other words, the users are able to store their personal information in the Blockchain without worrying about anyone to illegally steal or modify their data, ensuring the information security requirement of an identity management. Therefore it creates interesting research areas, especially from the perspective of technical challenges.

The biggest feature of the Blockchain is its decentralization where the whole database is maintained by all the nodes on the network. A consensus mechanism ensures that the creation and modification of data are agreed by all the nodes or the majority of the nodes. In this way, the Blockchain has the features of high security, not-easy to be tampered with [22] and so on. These features make it become a potentially ideal solution for authenticating and protecting identity management system among other uses. Meanwhile, the users are able to store their personal information in the Blockchain without worrying about anyone to illegally steal or modify their data, ensuring the information security requirement of an identity management.

2.3 Concepts of Identity Management on Blockchain

Blockchain is a decentralized distributed ledger technology. It records every transaction that ever happens in an incorruptible way, and no records are ever deleted [23]. Regardless of the approach, one function that is fundamental to IdM is securely binding together an identifier: a value that unambiguously distinguishes one user from another in a domain; and attributes (sometimes called certifications or claims): entitlements or properties of a user such as name, age, credit rating etc.

The first steps taken to tailor the use of distributed ledger technology (DLT) for establishing a secure and decentralised identifier-attribute mapping were taken in the design of Namecoin: the longest surviving software derived from Bitcoin. Namecoin provides a human-readable, decentralised and secure namespace for the “.bit” web domain [29]. This achievement contradicted conventional wisdom that a naming system exhibiting all three characteristics could not be designed [24]. Blockstack [25] has extended Namecoin’s scheme, to create a decentralized public key infrastructure (PKI): it registers bindings between a public key and a human readable identifier.

The decentralized approach offered by the blockchain eliminates this problem because the chain is 100% open to the public and no sensitive data is stored in the clear on the blockchain. Recently, several decentralized identity systems, such as Evernym, ShoCard, and uPort, have emerged that extend beyond naming and aim to provide a more complete suite of IdM functions. However, until now, there has been no evaluation of these proposals. We were interested in whether DLT-based IdMs have potential to go beyond previous approaches, or would simply create new “identity one-offs”.

The center of blockchain authentication would be a blockchain ID. This ID is essentially a block of data on the chain that can be both verified by any third party and can display necessary information such as date of birth. The secret to this verification is the ECDSA (elliptic curve digital signature algorithm). When adding an ID to the blockchain, an identification issuing service binds a public key by default and then transfers ownership of the private key to the user. This allows the user, and only the user, to sign a signature that can be verified against the public key stored in the blockchain. This identification of a user would serve as a decentralized source of authentication. It would essentially be a single-sign-on portal that can be accessed by any app while not being owned by any single entity. A protected app would only have to request a digital signature and an ID from a user requesting access. The app could then verify that the signature is valid and that the user’s ID verifies who they say they are. Given that DLT is suited to assuring consensus, transparency, and integrity of the transactions that it contains, a number of benefits of applying DLT to IdM also include decentralization, tamper-resistant, inclusiveness, cost saving and user control.

Lastly, it is worth noting Decentralized Identifiers (DIDs) are a new type of identifier for verifiable, "self-sovereign" digital identity. DIDs are fully under the control of the DID subject, independent from any centralized registry, identity provider, or certificate authority. DIDs are URLs that relate a DID subject to means for trustable interactions with that subject. DIDs resolve to DID Documents — simple documents that describe how to use that specific DID. Each DID Document may contain at least
three things: proof purposes, verification methods, and service endpoints. Proof purposes are combined with verification methods to provide mechanisms for proving things. For example, a DID Document can specify that a particular verification method, such as a cryptographic public key or pseudonymous biometric protocol, can be used to verify a proof that was created for the purpose of authentication. Service endpoints enable trusted interactions with the DID controller.

2.4 Security Features of Blockchain

Based on the literature review, the blockchain relies on many factors; ordering facts, block, consensus system, distributed ledger, and trustless, mining and the security is derived from several proofs challenges. These factors are briefly discussed below;

2.4.1 Ordering Facts: - Decentralized peer-to-peer (P2P) networks, such as Napster and BitTorrent are not new. Similarly, members of the blockchain network exchange facts. The point is that P2P networks, like other distributed systems, have to solve a very difficult computer science problem: the resolution of conflicts, or reconciliation to sustain their existence. Relational databases offer referential integrity, but there is no such thing in distributed system. If two incompatible facts arrive at the same time, the system must have rules to determine which fact is considered valid [32].

2.4.2 Blocks: - Blocks are a smart trick to order facts in a network of non-trusted peers. The idea is simple: facts are grouped in blocks, and there is only a single chain of blocks, replicated in the entire network. Each block references the previous one. So if fact F is in block 21, and fact E is in block 22, then fact E is considered by the entire network to be posterior to fact F. Before being added to a block, facts are pending, i.e. unconfirmed [32].

2.4.3 Distributed Ledger: - This means that every user stores the current ledger, preventing someone from altering a single point of truth. In traditional cryptography, a single point of truth could be a certificate authority; however, if that certificate authority was to be breached, a malicious attacker could replace the stored keys with their own keys, thus enabling them to masquerade as a plethora of users. By distributing the ledger, an attacker would have to breach every member machine and replace the blockchain with their own, making it functionally impossible for an attacker to alter the chain [20].

2.4.4 Consensus System: - In this context, consensus is the process of agreement among the network of users on which blockchain will be a recognized chain since there are replicated copies. This could theoretically be done given enough time; however, the usage of a proof of work problem makes this mathematically impossible. Consensus algorithms [26] for distributed systems are a very active research field. Some of these algorithms are Paxos or Raft algorithms [27], Monte Carlo algorithms [28]. The blockchain implements another algorithm, the proof-of-work consensus, using blocks.

2.4.5 Proof of Work: - Proof of work is a computational problem tied to the data of each block and required for the blocks to be accepted. The nature of this problem makes it mathematically impossible for someone to change the blockchain. Changing a block (which can only be done by making a new block containing the same predecessor) requires regenerating all successors and redoing the work they contain. Recently, many public blockchains are secured not by mining (proof-of-work) but by proof-of-stake [17] achieved by demonstrating control over some of the assets native to the ledger. Other proofs are noted in [31] such as Delegated Proof of Stake (DPoS) that utilises distributed voting to elect delegates and witnesses that participate in the validation process; Proof of Authority (PoAu) by granting special permission to one or more members to make changes in the blockchain; Proof of Elapsed Time (PoET) that aims to replicate a fair and random block generation process without spending valuable resources, such as coins, computational power or electricity; Proof of Activity (PoAc) which is a hybrid protocol that combines proof of work and proof of stake; Proof of Burn (PoB) which aims to replicate PoW cost for validation by charging validator nodes, who pay in coins to earn the privilege of validating blocks; Proof of Capacity (PoC) and other variants known as proof of space or proof of storage require that validator nodes commit hard drive space to increase their chances of producing the next block and earn its reward.

2.4.6 Trustless: - The security of blockchain is considered trustless because it prevents malicious parties from doing harm by nature of the protocol, without having to authenticate a transaction. Since the network of users will be adding blocks at a much faster rate than any single person could add blocks, it has therefore become mathematically impossible for a single person to breach the system [32].

2.4.7 Mining: - Some nodes in the chain create a new local block with pending facts. They compete to see if their local block is going to become the next block in the chain for the entire network. Blockchain challenges imply generating a random key to validate a block then it earns the ability to
publish their local block, and all facts in this block become confirmed [34]. This block is sent to all other nodes in the network. All nodes check that the block is correct, add it to their copy of the chain, and try to build a new block with new pending facts. The process of looking for blocks is called mining. This is because, just like gold mining, block mining brings an economical reward - some form of money [32]. That's the reason why people who run nodes in a blockchain are also called miners. By default, a node doesn't mine - it just receives blocks mined by other nodes. It's a voluntary process to turn a node into a miner node.

2.5 Existing Identity Systems based on Blockchain Technology

2.5.1 Blockstack

Blockstack is a decentralised identity, discovery and storage platform, built on blockchain technology. It makes use of virtualchains that allow the output of arbitrary state machines to be pinned to underlying blockchain infrastructure. Blockstack is similar to Ethereum in that it supports decentralised applications, but instead performs its computation off-chain. The underlying blockchain technology is used to authenticate an application before it is run by the user. Applications are not Turing-complete by design, but they can interface with the Turing-complete Ethereum blockchain by use of a virtualchain [25].

2.5.2 Evernym

Evernym was originally conceived as a top to bottom self-sovereign identity (SSI) solution, encompassing everything from the application to a proprietary distributed ledger and everything in between. Evernym is an identity system built on the permissioned Distributed Ledger Technology (DLT) known as Sovrin, which is dedicated solely to decentralised identity [35]. The Sovrin network is supported by the Sovrin Foundation, and it consists of inter-connected nodes forming a consensus on a shared ledger. Users create self-sovereign identities with personal attributes, and request claims from trusted third parties to build reputation. Currently, there is no financial incentive to host a Sovrin node, so the majority of the network is research-focused. The network plans to introduce premium claims in the future to provide rewards to nodes that distribute and verify identities.

2.5.3 Estonian e-Residency

E-Residency is a new digital nation for global citizens, powered by the Republic of Estonia. Unlike in many other countries, every Estonian, irrespective of their location, has a state issued digital identity without physical contact. Every person can provide digital signatures using their ID-card, Mobile-ID or Smart-ID, so they can safely identify themselves and use e-services. Residents can use the system for company registration, banking, payment services and document signing. The cards use 2048-bit RSA encryption for document signing and verification. Legal documents can be digitally signed using this technology, with the full support of the Estonian legal system [36]. E-Residency is a profound change and the recent announcement that the Estonian government is now partnering with Bitnation to offer a public notary service to Estonian e-Residents based on blockchain technology is of significance [37]. The application of blockchain to e-Residency has the potential to fundamentally change the way identity information is controlled and authenticated.

2.5.4 ShoCard

ShoCard is a digital identity application using a blend of blockchain-based data and facial recognition techniques. It focuses on user identification for Airlines, Secure Enterprise Identity Authentication, Financial Institutions and Website & App Login. It offers a mobile application for storing identities, while also pinning hashed and signed identity data to the Bitcoin blockchain. Users scan their document with the application, which reads each Machine Readable Zone (MRZ) and stores an encrypted version on the device. Each field is then one-way hashed, signed with the user's private key, and published to the blockchain [38]. Disclosure of user data to any third party is done by encrypting the local copy of information with the receiver's public key and transferring it via a QR code. The receiver can then validate the information against the signed version published on the blockchain. ShoCard presents a useful data disclosure process; ensuring transferred data is checked against the blockchain during each transaction. It fails to support any key or identity recovery protocols, however, and is inherently tied to identity supported by physical documents only.

2.5.5 uPort

uPort is a self-sovereign identity platform built on the Ethereum blockchain. At its core, it utilises a network of smart contracts for each user and offers a mobile application and accompanying developer libraries. uPort presents a novel recovery concept for digital identity, where a newly-generated ownership key is used to replace the former one for the user to gain access to an identity. This is achieved by allowing the user’s contact and
associates to vote for the replacement [39]. It can be established from the literature review that information security and blockchain are intertwined. This means blockchain can be implemented to achieve high level of security and blockchain itself is not secure without some concepts of information security embedded. Also, the scheme has resisted some rather extensive testing for weaknesses, both mathematically and computationally. The few failures that have occurred in practice have generally been because users were not careful in protecting their private keys, or else they used a fairly standard pseudorandom number generator to produce the private keys, which attackers then exploited. Also, given the proposed benefits of incorporating DLT into IdM schemes, the path to new forms of DLT-based IdM is really inevitable.

III METHODOLOGY

In this paper, National identity management system based on the Blockchain technology is proposed. Taking Nigeria as a case study, the current National identity management system [6, 12] is studied and it was gathered that the system is run on centralized database management system. The aim of the survey is to underscore the security challenges attributed to central database system as outlined in Section 2.5 of the literature review. Therefore, in the proposed model, identity creation, modification, disclosure, verification and authentication management are considered.

In carrying out the research work, some open source blockchain were surveyed for proper study of blockchain and its security concepts. These tools include Ethereum and MIT owned blockchain. The former allows concepts to be run on the system in form of command. Such concepts as contracts, proof-of-work and identity reputation are coded and run by user. The latter is an online tool with graphical user interface that allows user to experiment with various concepts of blockchain in real time and allows developers to write smart contract with solidity program.

Finally, a sample of local blockchain with salient features that capture the goal of this paper is developed in Python to demonstrate the application of blockchain for identity management using experimental method. Since this work is basically for identity management, more effort is put on identity management rather than transaction of currency.

3.1 System Analysis

The proposed model for Nigeria’s national identity system is illustrated in Figure 3.1. It replicates digital identity concepts recommended by both World Bank and the NIMC model [12] which harmonizes government agencies with National Identity Number (NIN). The model shows that disclosure of personal identity to external agent can be achieved, although with permission as explained in Section 3.3. Also, the government is the issuer of identity documents from birth certificates, driver licenses, etc. It is the government that has to accept the distributed ledger technology identity. Technically, the system requires the end user information stored on a device such as smart-card or an application. In the case of the former, the user needs to hold the card.

3.2 System Architecture

The architecture required for the proposed system is shown in Figure 3.2. The architecture consists of distributed application (DApp), ethereum system and The InterPlanetary File System (IPFS). The DApp is the software application that provides user interface and read the identity information of individual from the smart card. The IPFS is a new hypermedia distribution protocol, addressed by content and identities, as opposed to traditional location addressing. IPFS is designed to decentralize the web while simultaneously increasing the speed and security of the Internet.

The main concept is to store the encrypted identity information on the IPFS owing to its volume, then referenced it from the ethereum using address key. In that way, any time a user identity information is needed a call is made to the ethereum using public address/key which is then used to reference the main identity information after signing.

3.3 The Proposed Concepts

The operating mechanism of the proposed system is presented in this section. Unlike other identity management systems, our system consists of two parts: identity authentication module and reputation management module. The purpose of the identity authentication module is to allow an identity to correspond to blockchain-system public key address, ensuring that an entity in the real world can and only can create a unique corresponding virtual identity in our system. In this work various notational model are developed to solve various challenges as discussed further.
3.3.1 Identity Creation

It is a pre-requisite for every user whose information will be managed by the system to provide their personal information. In this work, personal identity information, denoted as IdentityInfo, is defined as a set of attributes used to identify a realistic user. The IdentityInfo is the combination of data usually in the form of text, image and biometric data, which is managed by the user and encrypted by the user’s private key in the Blockchain.

IdentityInfo = (name, contact, sex, occupation, biometric ...)

3.3.2 Identity Authentication

When identity is created, a pair of key (public and private) is generated. The hashed value of Public Key acts as the local identifier for the user and is also system address used to transfer value. In that case, a real user can create multiple public key addresses as his identity resulting to Sybil attack. Another negative effect of this situation is that a reputation result using this as a parameter would always give wrong results.

In order to solve Sybil attack and prevent people from opening multiple account, an identity solution is proposed, which establishes a virtual identity and real identity mapping relationship, where a unique public key address is assigned to a single virtual identity, and any real identity can and only have the unique virtual identity. In formal term, ID of user i is defined as the hash value of the combined real identity data as follows;

\[ ID_i = \text{Hash(IdentityInfo)} \] (ii)

Then, by definition the Identity_i is the identity of the user i in the system. Identity_i consists of two parts: the user’s ID and blockchain-system UUID address.

\[ Identity_i = (UUID_i, Address_i, ID_i) \] (iii)

where UUID_i is the user’s universal identity while Address_i generated by Blockchain-system such as Ethereum address for accepting or transferring information and values.

3.3.3 Identity Modification

In reality, the realistic user identity may be modified which may definitely change his Blockchain-System identity parameters. Two cases are considered in this work;

(i) Modification of user information: In order to ensure that the modification of identity information will not affect the user’s reputation, while preventing the creation of multiple identities, the specific rules are as follows:

\[ Identity'_i = (UUID_i, Address_i, ID'_i) \] (iv)

where Identity'_i represents the new identity value after the identity information is updated.

Therefore, when a user changes his personal information, three tasks occur (i) his ID will change (ii) the system will re-create the hash ID based on the new identity information and (iii) the old ID information is kept and maintained in the Blockchain. In this way, the attackers are unable to create multiple identities by personal information modification.

(ii) Change of users’ Blockchain-system address. When a user requests to change his Blockchain-system address, the system will also generate a new Identity, and the old address will remain stored in the Blockchain. In this way, the identity information is transferred from the old identity to the updated one, avoiding the attackers hide their identity by changing their addresses. It is noted that the modification of a user’s address requires the user provide his old address of the ID to ensure the legitimacy of the address modification operation. We use Address'_i to represent the changed address; the new identity is defined as follows:

\[ Identity'_i = (UUID_i, Address'_i, ID_i) \] (v)

3.3.4 Attribute Disclosure

To disclose a user’s attributes to another party such as government agency, hand-shaking protocol model is proposed. The third party service creates a disclosure request with the required attributes as shown in Model (vi). When the user receives this request on the device, they can confirm or deny the disclosure of the requested data (vii).

User B \rightarrow (Req) \rightarrow (attr_i, Addr_B, Addr_A) \rightarrow User A \leftarrow Req(Accept, Decline) \tag{vi} \tag{vii}
The user then encrypts the attributes and any associated signatures with the third party’s public key. The user also signs the request with the private key on their device and sends over the result (viii).

The receiving party then decrypts this message and verifies that the public key used to create the signature is linked to the correct identity on the blockchain. The validity of any attribute signatures is also verified by querying the blockchain and checking that the signing parties are trusted (ix). Attributes of a user can only be considered valid if they are accompanied by attestations from third parties that are trusted by the receiver. An extension to this solution would allow a chain of trust to be created that links entities to trusted roots.

3.4 System Model

In order to capture the entire populace in a nation, at the point of digital identity registration, identity is created on the blockchain usually with the public key hashed to generate the Ethereum address (Figure 3.3).

To prevent user from creating multiple accounts, the virtual identity is mapped with realistic identity by first generating the hash value of each field of the identity information, then map the result with the Ethereum address. All strings are transferred into a Smart Card given to the user while a copy is stored on the blockchain. The card stores an encrypted version of the identity information with each field one-way hashed, signed with user private key.

Disclosure of user data to any third party (Figure 3.3) such as government agencies is done by encrypting the local copy of information with the receiver’s public key and transferring it via a QR code. The receiver can then validate the information against the signed version published on the blockchain.

The third party can then create certification records confirming this physical and digital link, and hand them back to the user. This is referred to as verification token. The user can then present this token at subsequent/similar checks to streamline verification. However, each checkpoint is still required to compare the token against the blockchain. This is done to check for continued validity and possible revocation of the token.

3.5 Identity Contract

This is the file that contains the authoritative version of the user’s identity as well as the access control logic for attribute modification, hence contract. Once this contract is deployed to the blockchain, it is expected to link to the data storage in JavaScript Object Notation (JSON) format, manipulate the information based on the function therein and then return Identity described in Section 3.3.2. The reference address that is returned is the user’s Universally Unique Identifier (UUID). This address never changes, and it ensures the user maintains a persistent identifier even if their personal access keys are recovered or updated. The sample of data stored in the contract is shown in Table 3.1 below.

In conclusion, the models developed and discussed are for identity creation, modification, authentication and disclosure. The UUID is related to the ID on the fact that the latter is used to compute the former. Also, the Address is generated from the public key while the private key is only meant for signing the identity attributes to disclose to any other users.

Although, blockchain protocol already possesses the main security concepts to prevent unauthorised access to user’s identity information, however, data management logic is handled in smart contract.

3.6 System Pseudocode

We present the pseudocode which was implemented in Python. The code in python is attached to the Appendix I.

System initialization:

Generate genesis block if not exist:

\[
\begin{align*}
Index & \leftarrow 0 \\
auto_gen & \leftarrow timestamp \\
set data to & \leftarrow null \\
set block_hash & \leftarrow 0
\end{align*}
\]

while (add_more_block) :

\[
\begin{align*}
index & \leftarrow prev_block_index + 1 \\
timestamp & \leftarrow current_time_stamp \\
data_info & \leftarrow input_raw_data \\
uuid & \leftarrow uuid \\
block_hash & \leftarrow prev_block.hash \\
identity & \leftarrow (uuid, block_hash, hash_of_data_info)
\end{align*}
\]
end while

identity_modification:
if(identity_authentication \(\leftarrow\) successful):
    new_identity \(\leftarrow\) (uuid, block_hash, hash_of_new_data_info)
end if

identity_disclosure:
request \(\leftarrow\) packet(req_data_info, sender_block_addr, target_block_addr)
if(accept \(\leftarrow\) true):
    view_req_data_info \(\leftarrow\) target_data_info

blockchain = [create_genesis_block()]
previous_block = blockchain[0]

block_to_add = next_block(previous_block)
blockchain.append(block_to_add)
previous_block = block_to_add

IV EXPERIMENT

The tools used for the development of the prototypes are the following; Ethereum; a platform developed as an alternative protocol for building decentralized applications, Node.js; an open source javascript-based platform, Truffle; the most popular development framework for Ethereum with a mission to make life a whole lot easier, Testrpc; a Node.js based Ethereum client for testing and development, Web3.py a python library for interacting with Ethereum blockchain and ecosystem. Part of the logic code written in Python programming language is attached to Appendix I.

In order to verify the feasibility of our proposed identity authentication and verification model, we conduct a set of experiments by building a simple system in a local Blockchain.

The table 4.1 shows the public Ethereum address of the 10 users which is automatically generated in the development environment.

4.1 Identity Creation

The first task for every user is to enter personal information to be managed on the system. Initially, when no information is entered, the block hash value is set to normal value which, in this case, four leading zeros (0000) as shown in Figure 4.1. For demonstration purpose, only text data is used, although the technique will still work well for any form of data. The technique, as described in section 3, is to merge the whole information into a single entity and generate the hash value. It is important to note that every character that forms user information is taken into consideration in the process, including space. It therefore shows that a name without space, before, within or after, will definitely produce different hash value from that with space.

Figure 4.2 shows the demonstration of hashing of user information. This generates a hash value and carry along the “proof of work” to be done in order to restore the initial value of the hash i.e. the 4 leading zeroes. To solve this computational work, the Nonce value has to be adjusted (guess) continuously until the hash value is found (Figure 4.2). This process is called mining.

A sample of blockchain is shown in Figure 4.3. The blocks are connected by a block pointing to the hash value of the previous one (prev attribute) except the first (genesis) block that point to nothing but a default value (zeroes).

4.2 Identity Authentication

In the identity authentication model we propose that the creation of an identity requires the provision of unique physical information. The same physical information and the system address does not allow to create a second identity. We use three sets of input data to verify the identity authentication.

Case 1: Any attempt to create a user with different identity information with any users in the dataset but has the same system addresses with one of the existing users will not be successful. The experimental data are shown as in Table 4.2.

Discussion: From the experimental data shown in Table 4.2, in the two sets of data input, although the identity information are different, the new identity failed to be created by not recorded in a block in the system. It indicates that the system cannot create a new identity with the public key address the same as one of users already exist in the database.

Therefore, our system can ensure that users of the same public address cannot be created repeatedly.

Case 2: Any attempt to create a user with same identity information with any users in the dataset but has different system addresses with one of the existing users yield null
computed output ID. The experimental data are shown in Table 4.3.

Discussion: Creating new identity failed in Table 4.3 as the identity data is not recorded in the block. Even though the new identity has different public address with any existing users, however, the system still fails to create a new identity if the identity information is the same with one of the existing users in the system. It indicates that the users with the same identity information cannot be created repeatedly.

Case 3: Try to create an identity with different identity information and different public key address, and the experimental data are shown in Table 4.4.

Discussion: The identity creation is successful by recording the new identity in a block. It indicates that the system is able to create identities with unique identity information and unique address, which is consistent with the objective of the identity authentication.

The above three experimental cases demonstrate that once an identity information and address are binded successfully through creating an identity, the system will not allow the same identity information or address to be utilized in any other identity.

4.3 Identity modification

According to the proposed identity authentication model, if a user needs to modify his address or identity information, it should provide his identity ID or system address, respectively. Two cases will be used to verify this property of the identity modification model and the tabulation of the experiment is shown in Table 4.5.

Case 1: An attempt to modify a user identity information, by providing his true original ID and wrong ID, respectively.

Case 2: Try to modify the address of an existing user by providing (1) wrong ID information but correct original address; (2) correct ID information but wrong original address;

Case 3: Providing both ID information and the corresponding system address of the user are correctly provided.

Discussion: Table 4.5 shows that identity modification can be achieved only if both ID information and the corresponding System Address of the user are correctly provided. Any attempt to supply wrong value for either of the parameters will yield modification failure.

4.4 Attributes Disclosure

Attributes disclosure among two parties occurs using hand-shake protocol. The first step in the protocol is to send a request message to the attributes owner which either accept or reject the request as shown in Table 4.6. If the request is accepted, the information owner takes note of the request key, the address of the initiator of the request and the attribute requesting for. The attribute owner then encrypts the information along with the initiator address which will be decrypted by the latter.

V. Conclusion

Blockchain-based national identity management system was proposed in this paper. The work was motivated by the various security concepts provided by blockchain technology such as decentralization, proof-of-work, and anonymity just to mention few. These concepts gave developer a way out of the challenges posed by centralized data management system. The paper therefore proposed solutions for secured identity creation and modification, identity authentication model, and identity disclosure. Prevention of sybil attack is also achieved by binding a single virtual identity with physical identity which prevents user from creating multiple accounts on the system.

VII. Future Work

Although, efforts were made in this paper to meet the stated objectives, however, further work is required in the area of identity reputation and third party access and permission to user information. The hand-shake protocol for identity disclosure described in the work was not robust enough and not properly implemented. Therefore, the future work will focus on how to develop a whole and flexible hand-shake protocol that allows user to decide which attribute to disclose to people. It is important the identity management system has reputation sub-system that reflects user real behavior and to indicate how often users interact (transact) with each other. This may also lead to e-voting on blockchain in such a way that reputation will be what people lobby for by allowing people to agree or disagree with them. Furthermore, since the proposed system is developed for national identity, it is expedient that a private blockchain with large scale real data-based is developed for this purpose.
REFERENCES


AUTHORS’ BIODATA

AMUJO, Oluymei E. holds Master of Science (M.Sc.) degree in Computer Science (Information Security as a Major) from University of Abuja, Nigeria. In 2007, obtained his National Diploma (ND) in Computer Science from Rufus Giwa Polytechnic, Owo Ondo State and later obtained his B.Sc. (Hons) degree in Computer Science from Adekunle Ajasin University, Akungba-Akoko, Ondo State, Nigeria in 2014. He is a Practitioner who sees the widening gap between slow classroom and fast paced industrial halls and decided to bridge the gap. His research interests include; Information Security, Machine Learning, Software Engineering, IoT Development & Security. He can be reached via oluyemiamujo@gmail.com or oluyemi.amujo@hotmail.com or +2347063717176.

EBELOGU Christopher Ubaka is currently a Technical Officer in the Department of Computer Science, Faculty of Science, University of Abuja. He received his B.Sc Degree in Computer Science from the University of Abuja, FCT Abuja, Nigeria in 2008. He later proceeded to the University of Agriculture, Makurdi Benue State, Nigeria for his Post Graduate
Diploma in Computer Science. He currently holds an M.Sc Degree in Computer Science (Information Security as a Major) from the University of Abuja, FCT Abuja, Nigeria in 2019. His research areas are Information/Data Security, Big Data Analytics, and System Analysis. He can be reached via Christopher@uniabuja.edu.ng or +2348039394866.

Dr. Agu Edward Onyebueke obtained his PhD in computer science from Nnamdi Azikiwe University, Awka Nigeria. He is a lecturer in the Department of Computer Science at the Federal University Wukari, Taraba State, Nigeria where he has been a staff since October, 2012. He obtained Ordinary National Diploma (OND) in Computer Science, Higher National Diploma (HND) in Computer Science, Bachelor of Technology (B.Tech) in Computer Science and Information Technology and Masters of Technology (M.Tech) in Computer Science. His Research interests area includes hardware system, Internet/System security, Data Security, Artificial Intelligence and Expert System, System Analysis application, Data Mining and Machine Learning, and with recent focus on cloud and grid computing.

Dr. Agu Edaward has acquired CISCO certifications and is a member of Association for Computing Machinery (ACM), Nigerian Computer Society (NCS), International Society of Electrical and Electronics Engineering (IEEE), National Institute of Management (NIM), Society of Digital Information and Wireless Communications, etc. He has over thirteen (13) scholarly publications. He has supervised many thesis and graduated many outstanding students from the field of computing. He can be reached via aguedwardo@gmail.com, aguedward@fuwukari.edu.ng, +2348033894391, +2347018899150.
Figure 2.1: Overall description of NIMS
Source: [12]

Figure 2.2: Structure of Database Server for NIMS
Source: [6]
Figure 2.3: Visual representation of a block's data structure
Source: [34]

Figure 2.4: visual representation of a blockchain transaction
Source: [31]

Figure 3.1: Proposed framework for National Identity System

Figure 3.2: Proposed system architecture for National Identity System
hash(Public Key) → System Address IdentityInfo(attr₁, attr₂, …, attrₙ)

(UUID, IdentityAddr(hash(attr₁), hash(attr₂), …, hash(attrₙ)))

Smart Card

Figure 3.3: System design for identity creation

Sign (Priv.keyₐ(encrypt(attr, Addressₐ)))

User A

Req (attr, Addrₐ, Addrₐ)

User B

Figure 3.3: System design for attributes disclosure
Table 3.1: Data fields in data storage

<table>
<thead>
<tr>
<th>Block #</th>
<th>Serial number as block label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hash</td>
<td>The hash value of the block usually 64 length hex code</td>
</tr>
<tr>
<td>Previous Hash</td>
<td>The hash value of the previous block linked to the current block, usually 64 length hex code</td>
</tr>
<tr>
<td>User_Info</td>
<td>User information to be stored on the block</td>
</tr>
<tr>
<td>TimeStamp</td>
<td>The time at which process occurs within the block</td>
</tr>
<tr>
<td>System Address</td>
<td>The system (e.g. Ethereum) address generated in addition with public/private key. This is used for exchange of data in the blockchain</td>
</tr>
<tr>
<td>User Output ID</td>
<td>This is the computed ID that prevents user from creating multiple accounts on the blockchain.</td>
</tr>
</tbody>
</table>

Table 4.1: Creating identity with the same identity information

<table>
<thead>
<tr>
<th>User Address information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>
Figure 4.1: Block sample when data is nil

Figure 4.2: Block sample when data is entered

Figure 4.3: Sample of blockchain
Table 4.2: Creating identity with the same system address

<table>
<thead>
<tr>
<th>S. No</th>
<th>Identity</th>
<th>System Address</th>
<th>Computed ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61dfdf5995ac8</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b4be2c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
</tr>
<tr>
<td>2</td>
<td>Ahmed Brown</td>
<td>0x41f4ec6840e3875d88f8283160a61dfdf5995ac8</td>
<td>Null</td>
</tr>
</tbody>
</table>

Table 4.3: Creating identity with the same identity information

<table>
<thead>
<tr>
<th>S. No</th>
<th>Identity</th>
<th>System Address</th>
<th>Computed ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61dfdf5995ac8</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b4be2c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
</tr>
<tr>
<td>2</td>
<td>John Jimoh</td>
<td>0x41fb114cc3079e19d44f827dcecc669082bcf07f4</td>
<td>Null</td>
</tr>
</tbody>
</table>

Table 4.4: Creating identity with the same identity information

<table>
<thead>
<tr>
<th>S. No</th>
<th>Identity</th>
<th>System Address</th>
<th>Computed ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61dfdf5995ac8</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b4be2c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
</tr>
<tr>
<td>2</td>
<td>Smart Diko</td>
<td>0xc31067d903c425798ba01700a83cde5902b326</td>
<td>12610561f3c8a8ccabb84c8b0861c8217348a418d8f7517f3d193304f805a1:71e545d314e04da3a71102e7e956a10d</td>
</tr>
</tbody>
</table>
Table 4.5: Modification of System Address, IdentityInfo and Computed ID

<table>
<thead>
<tr>
<th>S. No</th>
<th>Accuracy</th>
<th>IdentityInfo</th>
<th>System Address</th>
<th>Computed ID</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Computed ID is right, System Address wrong</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61ddf5995sdq</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b42c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
<td>false</td>
</tr>
<tr>
<td>2</td>
<td>Both System Address and Computed Id are wrong</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61ddf5995sdq</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4354h868cc5ad3bf1b42c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>IdentityInfo is wrong</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61ddf5995ac8</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b42c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
<td>false</td>
</tr>
<tr>
<td>3</td>
<td>Both is right,</td>
<td>John Jimoh</td>
<td>0x41f4ec6840e3875d88f8283160a61ddf5995ac8</td>
<td>441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b42c702:2d6427e7c1b04ad9abe226b0c7e3b7e9</td>
<td>true</td>
</tr>
</tbody>
</table>

Table 4.6: Attributes disclosure

<table>
<thead>
<tr>
<th>Initiator</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>{firstname, 0xc31067d903c425798ba01700aee83cede5902b326, 0x41f4ec6840e3875d88f8283160a61ddf5995ac8}</td>
<td>{2d6427e7c1b04ad9abe226b0c7e3b7e9, 441adcd84dfab5b1ec5e8a8862a0d22b4667e4179d9868cc5ad3bf1b42c702, 0xc31067d903c425798ba01700aee83cede5902b326}</td>
</tr>
</tbody>
</table>
Appendix I

Snapshot of Program Output

Enter your data: Jimoh Adamu
Identity is authentic!!!
Block #1 has been added to the blockchain!
Hash: f078701e8f766f19b417ab5dc546ee4b5f9c7ae596b35fbb1369f331fe29c645
Previous Hash: a726fcf07ec0c6cd084511a89d8ecb691349a3b305e205dc8d63e5670fbc
Data: Jimoh Adamu
Timestamp: 2019-09-17 22:36:13.681703
System Address: 0x41fbb114c38f79e19d44f27dec669082bcf074
User Output ID: ca3e18089aab6ca58bb25ea143a8840dfb25c9612ed600f4fbc99c41aa219:9132561fa63e4643ac6a03fd546d44f

Enter your data: Alex Pam
Identity is authentic!!!
Block #2 has been added to the blockchain!
Hash: a20563cd7f9322aa50a72e8a876bcb1c5aa94f6cd8b9721ab9388b1c33ca5c
Previous Hash: f876761e8f766f19b417ab5dc546ee4b5f9c7a596b35fbb1369f331fe29c645
Data: Alex Pam
System Address: 0x310679d0c342579ba087a008ac3d6e5992b326
User Output ID: e917671eab2e91164968aa9671f19fc84067abf5173212f941300aa48d1e223:ddc38260f0974ebfac62938c58944e40
Enter your data: Oghonna James, Aba Street, Abuja
Identity is authentic!!!
Block #3 has been added to the blockchain!
Hash: 1a5e8f2c3b9a76f16a8b8619e2f529ad7a383efb4c2988a2a8ff9c748c5f99f
Previous Hash: a2053dcd79322aa50ac72ef8a07e8bc1c5aa9df6c0db9721ab930b1c33cfa54
Data: Oghonna James, Aba Street, Abuja
TimeStamp: 2019-09-17 22:36:30.537648
System Address: 0xb8c015e3ab7953ec6f774a108f5df897c6f173ee
User Output ID: 23d8b877e35fbb94aa3b2c2d2f4ebe792cc806e6eb02667627f8573de36aa:
7c758b18546604fe3abd36f7754a79c18

Enter your data: Adeleke John
Identity is authentic!!!
Block #4 has been added to the blockchain!
Hash: 93814f0e7404aa4ae6aebea9494d2e9f738923cf80fa548f4b6e8738ba36e
Previous Hash: 1a5e8f2c3b9a76f16a8b8619e2f529ad7a383efb4c2988a2a8ff9c748c5f99f
Data: Adeleke John
TimeStamp: 2019-09-17 22:41:57.847039
System Address: 0x41f41e66848e3875db88828316da61d0c5995ac8
User Output ID: b6762a75ba6ab435c129832c5f949954b7a94a987d7f2e5ec97d5d48b02efd:
2ed18852b76040f60f30e9200c2a2d2e9