AuthChain: A Decentralized Blockchain-based Authentication System

Shu Yun Lim, Pascal Tankam Fotsing, Omar Musa, Abdullah Almasri

Faculty of Business and Technology, UNITAR International University, Selangor, Malaysia

ABSTRACT

Current online services rely blindly on authentication providers to perform identity management and authentication. User credentials in these authentication providers are susceptible to large-scale account hacking. Distributed Ledger Teachnology (DLT) in general and blockchain can offer a solution by decentralizing ownership of credentials and a protocol for verifying one's record in an immutable chain of data. Blockchain can create a secure platform for online service providers to authenticate users with no single point of failure and decrease the possibility of attacks and user data leakages via backdoors. The purpose of this research is to analyze the limitations of centralized authentication system and propose a blockchain-based authentication solution to overcome the issues. In this paper we propose a robust, transparent and secure blockchain-based authentication mechanism called AuthChain. The implementation and testing of the proposed web and android native application was successfully completed with a prototype implementation on single node Ethereum Blockchain.

Keywords: *authentication, blockchain, DLT, Ethereum.*

I. INTRODUCTION

Authentication is the process of determining whether someone or something is, in fact, who or what it is declaring to be. It is the key process of any online systems or services for users to establish their identity and to gain access. Many aspects of our lives are touched by online communications. Therefore, secure the authentication process are the utmost important to prevent identity theft and spoofing attacks. Even though there are many legislation issues surrounding the exchange of sensitive data attributes, personal privacy concerns are addressed inadequately or simply overlooked.

Blockchain-based authentication enables authentication and identification of devices and users. The immutable blockchain ledger verifies and ensures that the users, transactions, messages are legitimate. Blockchain authentication [1] is done by smart contracts which are written and deployed to blockchain. A smart contract generator can be programmed through a Smart Contract Authentication (SCA) layer to activate and execute every time an authentication is required by either party or self-govern itself within a predefined scope of actions. The need for a third party to authenticate transactions is eliminated. Costs can be reduced while security and privacy are greatly enhanced. Effort of hijacking the authentication process would be much greater in the distributed environment. With Blockchain-based authentication, signing and decryption keys stay on the device while verification and encryption keys are stored on the blockchain. In a way providing protections against critical cyber-attacks such as phishing, man-in-the-middle, replay attacks.

In this research, AUTHCHAIN, a decentralized authentication system is proposed. The architecture, technical specifications and the features of blockchain used for an authentication process is studied. After providing previous works related to authentication and identification mechanism, the design and the development of single node blockchain for authentication process is discussed. The proposed solution consists of two mobile apps, a front-end mobile app for users and an validation app for third party services to authenticate users. Lastly, the proposed solution is implemented and tested on single node Ethereum Blockchain.

II. RELATED WORKS

As a new technology with infinite opportunities, many individuals, companies as well as governments have started some researches and development on blockchain technology. While it may take years for blockchain technology to mature fully, many blockchain solutions and applications are already perfectly feasible in the near term, and new opportunities will continue to present themselves as the underlying technology evolves. In this section, blockchain solutions in the field of authentication are presented.

A. MyData

MyData [2] is a research commissioned by Finnish government for personal data management. This Nordic self-sovereign identity model is driven by the concept of human centric control, usability, accessibility and openness. MyData can be used to secure flow of data between sectors likes governments, healthcare and finances. The core of MyData authentication are user managed access, OpenID single sign-on and Oauth 2.0 which control access to Web APIs. Blockchain is used to distributed control of fraudulent activities to the entire network of stakeholders, as any attempt to tamper with the blockchain is easily detectable.

The research, which joint forces with Sovrin, aims at strengthening digital human rights while opening new opportunities for business to develop innovative personal data services. It is also aiming at addressing EU General Data Protection Regulation (GDPR) [3], new rules on controlling and processing personally information enforced since May 2018.

B. Waypoint

Waypoint [1] is a decentralized multi-factor authentication system that is deployed on the Ethereum Virtual Machine. This solution allows identity authentication to be performed on the Blockchain, with Web API based implementation.

With a mobile base apps and desktop version available, Waypoint allows application to secure multiple modules within one product by defining multiple functions. It provides feature to store user behaviour and perform analytics for real time behavioural based authentication. The commercial solution is currently at beta-stage.

C. Blockstack

Blockstack [4] [5] provides decentralized services for naming (DNS), identity, authentication and storage. developers can use JavaScript libraries to build serverless apps and not worry about handling infrastructure. Blockstack will replace the contemporary client/server model; users control their information, apps run client-side, and the open Blockstack network replaces server-side functionality.

D. CertCoin

CertCoin [6] is a decentralized authentication system based on the NameCoin [7] blockchain. This system carries the best aspects of transparent certificates authorities and web of trust. CertCoin is public and auditable. CertCoin helps the expected features of a full-fledged certificate authority such as certificate creation, revocation, chaining, and recovery. Domain purchases and transfers are executed with simple Bitcoin transactions to incentivize miners. The CertCoin layout additionally facilitates trusted key distribution that makes it more suitable for performance conscious applications. Besides that, it also addresses several issues inherent to current PKIs, such as the need for a trusted third party and limited accessibility.

III. PROPOSED MECHANISM

The current state of digital authentication requires significant trust in third parties. Users must trust the websites or services providers to safeguard their authentication data since personal information could be collected for data mining, profiling and exploitation without users' knowledge or consent. We aim to improve current systems with AuthChain solutions for authentication with the following features:

1) Trustless authentication:
   With blockchain based system authentication of any user must be verifiable not by just one node, but by other participating nodes at any moment of time. Authentication is done without having to rely on a centralized authentication provider
2) Not using password for login:
   A user generates a pair of keys, public and private for an account on a trusted device. The public key is used to identify every user. The public key stored on user device (in Android keystore or iOS keychain) provides access, as opposed to the use of password for login. Hence issues related to weak password chosen by user is not relevant.
3) One account:
   One AUTHCHAIN account will allow users to authenticate themselves on any websites that are using AUTHCHAIN as authentication service provider. AUTHCHAIN can be used to bind authentication of online accounts on the social media platforms, e-commerce, financial websites or verification of identity in the event of face to face meeting.
4) No storing of user credential data is required at services requesting authentication:
   The risks of data lost, and data violation is off the point. Only authentication transaction record will be recorded on chain.
5) Immutable:
   The logs of transactions, which are created by consensus among the nodes are hashed and added to the blockchain. Those data store in an encrypted form is immutable. Any changes to these logs require the rebuilding of the merkle tree, which is almost impossible.
6) Flexible and resilient authentication:
   Users do not have to carry around identifying document all the time. Users could prove their

identity by providing the generated hash or display the hash in the form of QR code for scanning in the event of face to face meeting. For authenticating a user, the third party will have our third-party authentication app which is also included in the development of solution. The app is used to verify user identity through QR code or hash signature.

A. Proposed architecture

The stakeholders in our solution are public users with mobile device, third party online services requesting for user authentication and AuthChain, the authentication blockchain. Proposed architecture is as shown in Fig. 1 and Fig. 2.

1) User register account on AUTHCHAIN mobile app and request for a token as a unique identifier
2) The user data are hashed using the SHA256 and this transaction is recorded on chain.
3) The generated unique identifier is sent to user and could be later used for authentication
4) Third party user authenticate user by scanning the QR code or through API for online services.
5) Third party user verify the user by referring to the data on chain.
6) Authentication result sent to third party user.
7) If the user data exist in the blockchain, authentication is successful.

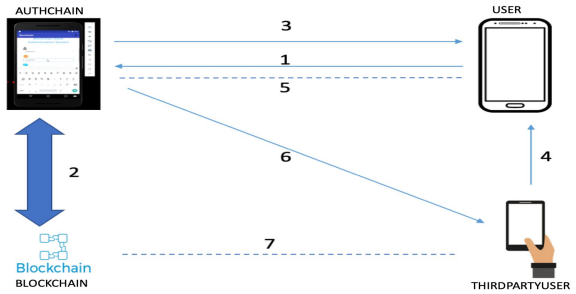

Figure 1 Proposed architecture

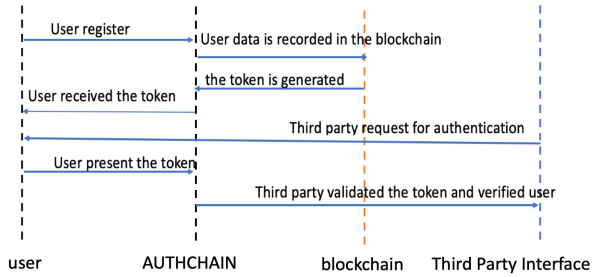

Figure 2 Flow of data for authentication

B. Authentication Algorithm

Our proposed solution can authenticate users from any public platform on the AuthChain blockchain, present proof of user authentication and complete any subsequent authentication to any third party at any time. Authentication will be done using the following algorithm.

1) User generates a pair of public and private keys (*K+*, *K-*) and select an unassigned permanent account number (*PAN*) on AuthChain blockchain. (*PAN* will be registered on the blockchain).
2) User generates a secret key *Ks*.
3) User calculates a hash function *H(Ks, PAN)* and publishes it to AuthChain blockchain.
4) Client sends (*Ks, PAN*) and public key *K+*, for which the account will be registered.
5) The third party then calculates user's hash and compares it with the hash found on AuthChain blockchain. If the hashes are identical, then the oracle considers the authentication process to be completed.
6) Third party calls on the registration method of the online account smart contract with arguments of (*Ks*, *PAN*, *K+*, *N*)
7) Online account smart contract then registers account *PAN* with the public key *K+* and adds information to the blockchain about successful authentication of a user *N*, a public key *K+*.

Main defense against this attacks comes from the impossibility of brute forcing the incoming data *H(Ks, PAN)* which would satisfy the authentication hash *H*. Intruder does not have an ability to generate a valid pair *(Ks, PAN)* not knowing the secret key *Ks* chosen by a valid user. Now that the authentication of a user can be proven, the permanent account number *PAN* can be subsequently used as an authenticator of the user *N*.
This process would also make it possible to create a transaction to any account of the public network. Platform smart contract will automatically complete the transaction to the *PAN* which the authentication of the receiver has been completed.

IV. IMPLEMENTATION

The project prototype consists of three components, AUTHCHAIN the Ethereum blockchain, a mobile front-end app for user, and a mobile app for third party users to perform user validation. The solution contains a bootstrap, img, a js folder (Fig. 3) and as well as data.json for the blockchain. For plugins and alerts, Bootstrap version v3.3.4 is used. The file data.json was developed in order to store the block. Our node genesis block code is as shown in Fig. 4.

```
package-lock.json
{
  "requires": true,
  "lockfileVersion": 1,
  "dependencies": {
    "crypto-js": {
      "version": "3.1.9-1",
```

```
      "resolved":
"https://registry.npmjs.org/crypto-js/-/crypto
-js-3.1.9-1.tgz",
      "integrity":
"sha1-/aGedh/Ad+Af+/3G6f38WeiAbNg="
    }
  }
}
```

Figure 3 js folder

```
genesisblock
{
  "alloc": {

"acc46a2a555c74ded4a2bd094e821b97843b40c0": {
      "balance": "1940000000000000000000"
    },

"5c75cb43354c9360865fcc170b6925278964bb2c": {
      "balance": "2194000000000000000000"
    }
  },

  "mixHash":
"0x0000000000000000000000000000000000000000000000
0000000000000000000",
  "minedBy":
"0x0000000000000000000000000000000000000000",
  "timestamp": "0x00",
  "parentHash":
"0x0000000000000000000000000000000000000000000000
0000000000000000000",
  "extraData":
"0x11bbe8db4e347b4e8c937c1c8370e4b5ed33adb3db6
9cbdb7a38e1e50b1b82fa"
}
```

Figure 4 Genesis block

The Ethereum smart contract was develop on Solidity for reading and writing user data. Pragma solidity ^0.5.8 indicates that this source file is not intended to be compiled with a compiler earlier than version 0.5.8 or with 0.5.9 and higher [8]. A contract in the sense of Solidity is a collection of functions and data that resides at a specific address on the Ethereum blockchain. State variable is seen as a single slot in a database that can be queried and altered by calling functions that manages the database. In the case of Ethereum, this is always the owning contract. The functions *getUser*, *addUser* and *editUser* in fig 5 are used to modify or retrieve the value of the variable.

User data is read from the mobile app includes full name, passphrase, permanent account number (PAN) and phone number as shown in Fig. 6. Permanent account number can be any identifier issued by government i.e. social security number or income tax number. The user data is first hashed with SHA256 algorithm. The system will then generate a QR code of the hash (Fig. 8) to be used as identifier for the validation party. This allows validation party to proceed with authentication without accessing the data itself.

```
pragma solidity ^0.5.8;

contract Users {
  address owner;

  struct UserData {
    string fname;
    string pphrase;
    string pan;
    string phone;
  }
  mapping (string => UserData) private users;
  string[] public userTemplates;

  constructor() public {
    owner = msg.sender;
  }

  modifier onlyOwner {
    require(msg.sender == owner);
  }
  function getUser(string memory key) public view
returns(string memory fname, string memory
pphrase, string memory pan, string memory phone)
{
    return (users[key].fname,
users[key].pphrase, users[key].pan,
users[key].phone);
  }
  function addUser(string memory fname, string
memory pphrase, string memory pan, string memory
phone,  string memory template) public {
    UserData memory newUser = UserData(fname,
pphrase, pan, phone);
    userTemplates.push(template);
    users[template] = newUser;
  }
  function editUser(string memory fname, string
memory pphrase, string memory pan, string memory
phone, string memory template) public{
    users[template].fname = fname;
    users[template].pphrase = pphrase;
    users[template].pan = pan;
    users[template].phone = phone;
  }
  function getTemplatesSize() public view
returns(unit) {
    return userTemplates.length;
  }
}
```

Figure 5 Ethereum smart contract

```
class uData{
    constructor(name, adhaar, pan, phone,
publicKey){
        this.name = name;
        this.adhaar = adhaar;
        this.pan = pan;
        this.phone = phone;
        this.publicKey= publicKey;
    }
}

class Block {
    constructor(timestamp, uData, previousHash =
"") {
        this.previousHash = previousHash;
        this.timestamp = timestamp;
        this.uData = uData;
        this.hash = this.cHash();
    }
```

```
    cHash() {
        return SHA256(this.previousHash +
this.timestamp +
JSON.stringify(this.uData)).toString();
    }
}
```

Figure 6 Class files

The Android mobile app shown in Fig. 7 and Fig. 8 makes it possible for validation party to authenticate any accounts participate in AuthChain network. Platform smart contract will automatically complete the authentication if valid user data exist in the blockchain.

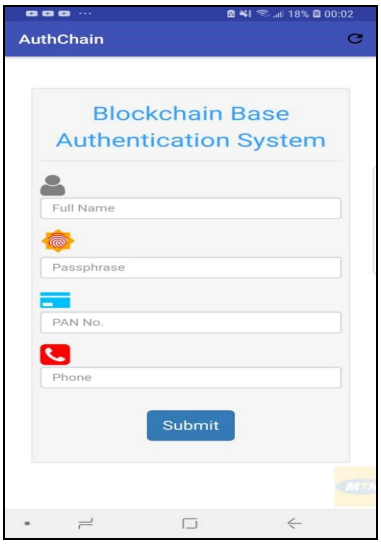

Figure 7 AuthChain Mobile App

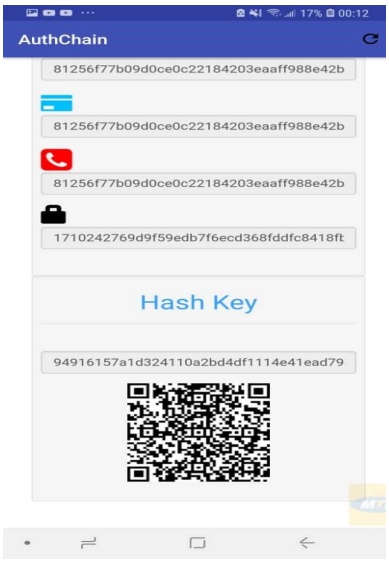

Figure 8 AuthChain mobile app Hash key and QR code.

V. OPEN ISSUES AND DISCUSSION

The discovery of this new mechanism creates a secure platform for service providers to authenticate users with no single point of failure and prevent attacks and leakages of user data. This solution is a tamper-proof reference point to verify personal data without having to expose the actual data to a service provider.

Blockchain authentication solution by design is distributed, decentralized and fault-tolerant which decreases the deployment and maintenance cost. However, scalability seems to be the biggest challenge with public blockchain. Some argued that by centralizing some parts of the technology, blockchain authentication will be more cost effective and secure.

On the other hand, instead of on-premise deployment of blockchain network, Blockchain-as-a-Service (BaaS) [9] allows customers to leverage cloud-based solutions to build, host and use their own applications and smart contracts on the blockchain. Cloud providers take over other necessary tasks to keep the infrastructure operational. Undeniably, BaaS is aiding the blockchain adoption across businesses. Companies such as IBM, Microsoft, or even google had started offering the cloud as a service business model based on blockchain technology.

Besides, enhancement of Ethereum and Hyperledger blockchain is required which in turn could improve the performance of blockchain network. In real world implementations, it will require an overhaul or at least a focused effort to integrate this technology with existing implementations of identity authentication to begin an initial acceptance of this technology in the market.

ACKNOWLEDGEMENT

This research is supported by FRGS research grant FRGS/1/2018/ICT04/UNITAR/03/1.

REFERENCES

[1] Ismail, R., "*Enhancement of Online Identity Authentication Though Blockchain Technology*". 2017: Malaysia.

[2] Atzori, M., "*Blockchain technology and decentralized governance: Is the state still necessary?*" 2015.

[3] Council of the European Union , E.P., Regulation (EU) 2016/679 of the European Parliament and of the Council Official Journal of the European Union, 2016.

[4] M. Ali, R.S., J. Nelson and M. J. Freedman, "*Blockstack: A New Internet for Decentralized Applications (Whitepaper)*". 2017.

[5] M. Ali, J.N., R. Shea and M. J. Freedman. "*Blockstack: A Global Naming and Storage System Secured by Blockchains*". in 2016 USENIX Annual Technical Conference. 2016.

[6] Conner Fromknecht, D.V., Sophia Yakoubov CertCoin: "*A NameCoin Based Decentralized Authentication System*". 2014.

[7] NameCoin. Namecoin. 2018; Available from: *https://www.namecoin.org/*.

[8] Solidity. Introduction to Smart Contracts. 2019; Available from: *https://solidity.readthedocs.io/en/v0.5.8/introduction-to-smart-contracts.html*.

[9] Samaniego, M., & Deters, R. . "*Blockchain as a Service for IoT. in IEEE Green Computing and Communications (GreenCom) and IEEE Cyber*", Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). 2016. IEEE