Combating Identity Attacks in Online Social Networks: A Multi-Layered Framework Using Zero-Knowledge Proof and Permissioned Blockchain

Md Jahangir Alam*, Ismail Hossain†, Sai Puppala‡ and Sajedul Talukder§
School of Computing, Southern Illinois University Carbondale
IL, USA
Email: *mdjahangir.alam@siu.edu, †ismail.hossain@siu.edu, ‡saimaniteja.puppala@siu.edu, §sajedul.talukder@siu.edu

***Abstract*—Identity attacks, such as impersonation, identity theft, and fraudulent account creation, pose significant threats to the security and trustworthiness of Online Social Networks (OSNs). In this paper, we propose a robust and secure framework to verify user identities without compromising their privacy by developing a multi-layered framework leveraging zero-knowledge proof (ZKP) and Hyperledger Fabric private blockchain. We introduce a blockchain-based government identity provider system, coupled with a zero-knowledge proof-based signup process for social networks. Our prototype authenticates user identities in multiple layers, effectively mitigating fraudulent, cloned, and multiple account creations. Our experiments with n (n = 50) users showed a 100% success rate for our system, highlighting its effectiveness compared to other OSNs.**

***Index Terms*—blockchain, privacy, social network, zero-knowledge proof**

I. Introduction

In the current digital age, online social networks (OSNs) have emerged as significant platforms for communication, information dissemination, and social interaction. However, this increasing reliance on OSNs also reveals a critical vulnerability: identity attacks [1]. Fake, cloned, and multiple accounts compromise user security, undermine the credibility of information, and erode trust in these platforms. One of the significant challenges facing OSNs is the inability to effectively verify the identity of a user before account creation. Current systems employed by OSNs primarily rely on post-creation methods to detect and deactivate fake accounts [2]. These methods include machine learning algorithms that analyze account behavior, user reports, and manual verification processes. However, these approaches are reactive rather than preventative, which means fake accounts can still exist temporarily and potentially engage in harmful activities before being detected [3]. This situation highlights the pressing need for proactive solutions that can effectively authenticate a user's real identity prior to account creation, significantly reducing the risk and impact of fake accounts. The current systems primarily rely on email or phone verification, which can easily be manipulated, leading to fake, cloned, or multiple accounts. This issue underscores the need for a more robust, reliable mechanism for user identification to enhance the authenticity and security of user interactions within these platforms, without divulging confidential information. This study explores such a solution, deploying a unique amalgamation of zero-knowledge proof (ZKP) [4] and private blockchains for secure user verification. Zero-knowledge proof is a cryptographic principle enabling one party to demonstrate to another that they possess a specific piece of information, without revealing any other details except the proof of knowledge. Coupled with the decentralized and immutable nature of blockchain technology, ZKP becomes a formidable tool for identity verification.

This paper introduces a novel multi-layer framework aimed at preventing identity attacks on Online Social Networks (OSNs). We propose a blockchain-based government identity provider and a zero-knowledge proof (ZKP) based OSN signup process. These tools leverage cryptographic algorithms to generate unique identity strings for users.

Our private blockchain system, developed with Hyperledger Fabric 2.x, involves government organizations (e.g., Social Security Administration (SSA), Department of Motor Vehicle (DMV), Department of Public Health (DPH), etc.) in the secure data storage process, exploiting blockchain's permissioned, immutable, and decentralized

ASONAM '23, November 6-9, 2023, Kusadasi, Turkey
© 2023 Association for Computing Machinery.
ACM ISBN 979-8-4007-0409-3/23/11...$15.00
https://doi.org/10.1145/3625007.3627722

properties to bolster user identity security in OSNs. Based on this, we developed a prototype of a government identity provider system and social network. Through rigorous experiments involving fake, cloned, and multiple account creation attempts on our prototype system, we demonstrate a 100% success rate in thwarting fraudulent account creation, highlighting the stark contrast with existing OSN platforms.

In summary, we introduce the following contributions:

- **Identity Verification for OSNs:** We developed a robust Identity Verification system for Online Social Networks (OSNs) to enhance security by authenticating users efficiently and mitigating risks of fraudulent accounts.
- **ZKP Signup Process:** We introduced a secure and private Signup Process utilizing Zero-Knowledge Proofs (ZKPs), allowing users to authenticate without revealing sensitive information, thereby enhancing user privacy and data security.
- **Prototype with Multi-Layered Framework:** We implemented a prototype based on a scalable, robust, and user-friendly multi-layered framework, demonstrating the feasibility and effectiveness of our proposed identity verification solutions.

II. Background

A. Zero-Knowledge Proof (ZKP)

Zero-knowledge proof (ZKP) is a cryptographic technique developed by Goldwasser et al. [4] in 1985. It allows a party to prove their knowledge without revealing the information itself, effectively preserving confidentiality during exchanges. ZKPs have broad applications, including personal information checks and authentication systems. Two primary ZKP variants are Interactive ZKP, involving message exchange between prover and verifier, and Non-Interactive ZKP, condensing multiple messages into one. Other versions include Graph Isomorphism, zk-SNARK, zk-STIK, zk-STARK, Designated Verifier, Bulletproof, and Lattice-Based ZKPs.

B. Blockchain

Blockchain, first introduced by Satoshi Nakamoto for Bitcoin [5], is a decentralized ledger technology that securely records transactions in a transparent manner. It's unique because every node in the network maintains an identical copy of the ledger, and once a transaction is recorded, it's immutable. Alterations require a consensus among nodes, achieved via mechanisms like Proof of Work (PoW) [6], Proof of Stake (PoS) [7], and Delegated Proof of Stake (DPoS) [8]. Blockchain transactions begin with a request, proceed with block creation and network-wide validation, and culminate in block addition to the blockchain. Blockchains can be either permissionless, open to all like Bitcoin and Ethereum, or permissioned, restricted to specific groups like Hyperledger Fabric [9].

III. Identity Attack Models

In this section, we describe different identity attack models in OSNs. In this attack scenario, the prevalent attack models are identity clone attack, Sybil attack, sockpuppet attack, etc. Usually, these attacks specifically target users' personal information as well as the personal information of their friends.

Identity Clone Attack. Identity clone attacks (shown in Figure 1) occur when a malicious actor creates a profile using the same name, photos, and other personal details of a legitimate user, effectively cloning their identity. The objective of such attacks usually involves deceiving the friends or followers of the legitimate user into believing they're interacting with a genuine person.

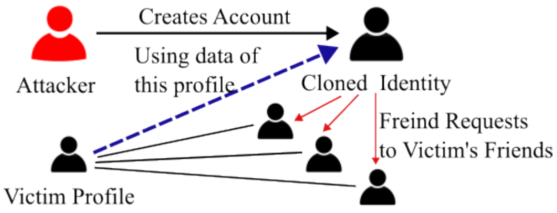

Creates Account
Attacker
Using data of this profile
Cloned Identity
Freind Requests to Victim's Friends
Victim Profile

Figure 1: Identity Clone Attack Model

Sybil Attack. A Sybil attack (shown in Figure 2) involves a single malicious entity creating numerous fake profiles to manipulate the network or its users. These attacks can influence various network aspects, such as spreading false information, manipulating popularity metrics, or directly targeting individual users. Combating these attacks typically involves anomaly detection, user verification, and restrictions on new or unverified accounts.

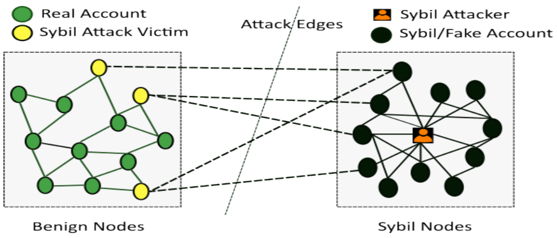


Figure 2: Sybil Attack Model

Sockpuppet Attack. This is an identity attack model in which malicious users (puppetmasters) register for multiple accounts (sockpuppets) [10]. Using these accounts puppetmasters do different malicious activities. To advertise fake things, spam or cause controversy on social networks. If any account gets banned by a social network site puppetmasters create another account to continue their activities. The sockpuppets are multiple fake accounts created by the same user. They are the source of several types of manipulation such as those created to praise, defend, or support a person or organization or to manipulate public opinion. Figure 3 shows the sockpuppet attack model.

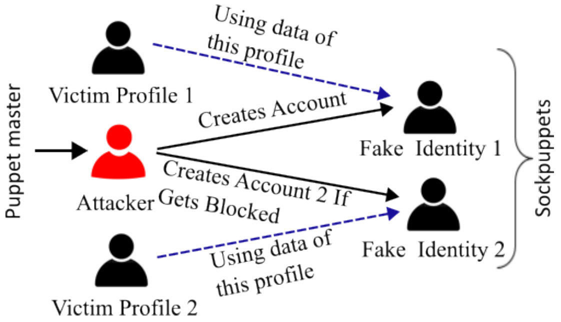


Figure 3: Sockpuppet Attack Model

IV. Threat Model

In outlining our threat model, we conceptualize the adversary within our framework presumed to possess the capability to launch diverse attacks with the aim to either impersonate legitimate users, or compromise user accounts on the Online Social Network (OSN). We now delve into specific attack vectors associated with each identified threat.

1) **Identity Provider Threat:** This involves the risk of adversaries obtaining access to the government module accounts and posing as legitimate users on the Online Social Network (OSN).
2) **Fake Proof & Man-in-the-Middle Attacks:** This threat involves attempts to create accounts using fraudulent identity strings and the interception or analysis of transmitted identity strings and Zero-Knowledge Proof (ZKP).
3) **Replay & Denial of Service Attacks:** This involves malicious attempts to re-submit acquired ZKP proofs and the overwhelming of system resources by excessive requests, causing prolonged downtimes.

We encapsulate multi-dimensional security considerations inherent in our framework to thwart various malicious attempts and safeguard user information.

V. Related Works

In this section, we present an overview of existing literature related to the concepts that form the backbone of our proposed Zero-Knowledge Proof (ZKP) signup method. We delve into pertinent research on the application of ZKP in identity verification and authentication, the role of challenge questions in user authentication, as well as previous works involving the use of blockchain technology, specifically Hyperledger Fabric, in the context of OSNs.

A. Zero-Knowledge Proof And Identity Provider

Amid the rising identity issues on digital platforms such as Online Social Networks (OSNs), Song et al. [11] employed Zero-Knowledge Proof (ZKP) and blockchain to develop an anonymous user registration scheme. In parallel, Yang [12] integrated zk-SNARK into the claim identity model, ensuring identity privacy through a new privacy attribute token. Salleras [13] presented a library designed for generating ZKPs on resource-limited devices, highlighting the potential application of zk-SNARKs in the Internet of Things (IoT) scenarios. In healthcare, Sharma [14] proposed an innovative Blockchain and Proxy re-encryption-based framework to overcome conventional Electronic Health Record (EHR) architectures' limitations, with the aid of ZKP. Meanwhile, Feng [15] combined ZKP with smart contracts to create a novel data-sharing solution for the Industrial Internet of Things (IIoT), underlining the synergy between system security (provided by blockchain) and privacy (afforded by ZKP).

In contrast, this article proposes a methodology that combines zero-knowledge proof and smart contracts. This approach facilitates data validity and consistency between data owners and cloud service providers. Additionally, proxy re-encryption technology is employed to securely share data among multiple participants. Leveraging the tamper-resistant and traceable characteristics of the blockchain, data can be verified and transactions can be traced.

B. Challenge Question In Authentication and Identity Verification

Security questions, commonly used as fallback authentication in various sectors, have demonstrated vulnerabilities due to easily guessable responses. Anvari et al. [16] proposed a novel method for generating unique story-based security questions to curb hacking attempts. Dhekane et al. [17] suggested a shift towards recognition-based security questions from recall-based ones. Micallef et al. [18] studied user strategies in memorizing security questions and proposed enhancements based on the identified weaknesses. In our system, we use AI-based Challenge-Response Authentication that involves the user (the claimant) proving their identity to a system (verifier) by correctly answering challenges or questions. Our Challenge Question Generator Engine (CQGE) helps in generating, presenting, and validating the responses to these challenges.

C. Hyperledger Fabric Private Blockchain

Recent developments in blockchain technology have facilitated privacy preservation and security enhancements across various fields. Li et al. [19] proposed a Hyperledger Fabric-based scheme for privacy-preserving identity verification in ridesharing. Ravi et al. [20] examined how blockchain can improve supply chain management, providing secure alternatives to traditional web technologies. Lastly, Stamatellis et al. [21] presented a method for securely storing patient records on Hyperledger Fabric, ensuring anonymity and unlinkability. Given these developments, we have also opted to utilize Hyperledger Fabric as the private blockchain database in our approach.

VI. Methodology

This section outlines the design and structure of our innovative framework, which seamlessly combines zero-knowledge proof (ZKP) with blockchain technology. We begin with an overview before describing the elements of the framework we suggest.

A. Overview

This subsection presents the procedures for identity string generation within the Government module. It encompasses the Zero-Knowledge Proof (ZKP) protocol, the Online Social Networks (OSNs) signup via ZKP-encrypted identity string and proof, and the creation of challenge questions. The Government module acts as the Identity Provider, the user is the Prover, and the Blockchain Client Application serves as the Verifier, executing the necessary smart contracts. In essence:

1) A user creates an account on the Government module.
2) The module produces an identity string with the user-selected attributes, proving and verifying keys, and saves them to the blockchain.
3) The user generates a proof for the OSN using the identity string and proving key. After the proof's verification, the Blockchain Client provides challenge questions.
4) Successful ZKP verification and challenge completion allow user account creation on the OSN, ensuring authentic user identities without revealing sensitive information.

B. User Registration in the Government Module

A user initiates the registration process in the Government Module (GM) by entering and verifying their email, followed by password creation. The credentials, comprising the email and password, serve as the primary access keys to the account. To bolster account security, the GM platform incorporates an enhanced dashboard. This dashboard facilitates advanced authentication measures, such as One-Time Passwords (OTPs) delivered to the user's registered email or phone, in addition to the provision for token set generation tailored for authentication purposes. It's worth noting that user passwords are stored in an encrypted format. Specifically, the encryption leverages the BCrypt algorithm (employing version 2a, with a strength or cost factor of 31 and utilizing SecureRandom for random number generation) [22].

C. Identity String and ZKP Key Generation

Users intending to generate an identity string must first have an account with the Government Module (GM). This module has an identity creation interface that provides a thorough attribute list, sourced from various government organizations such as the SSA, DMV, and DPH. This attribute list, which encompasses details like first name, last name, email, date of birth, and gender, is curated based on the registration forms of prevalent OSNs. Depending on the specific requirements of a target OSN — for instance, Facebook necessitates first and last names, an email or phone number, birth date, and gender — the user selects the relevant attributes. Figure 4 shows the identity string and ZKP key generation process.

The selected attributes undergo a hashing process as illustrated in Algorithm 1, resulting in an "identity string." Simultaneously, a key pair, comprising the proving and verifying keys, is generated by the GM. This identity string, paired with the key set and the current timestamp, is subsequently stored on the blockchain. To counteract replay attacks, we incorporate a timeout period with the timestamp.

The designated algorithm accepts two primary inputs: a list of chosen attribute keys, $\mathbb{A}_{\mathcal{K}}$, and their corresponding values, $\mathbb{A}_{\mathcal{V}}$. If either list is void, the algorithm terminates without executing further operations. All attribute keys and values are respectively concatenated into singular strings, $\mathcal{K}_{\mathcal{C}}$ and $\mathcal{V}_{\mathcal{C}}$. Post concatenation, a hash is derived from the collective value using SHA-256, producing a byte array that is eventually transformed into the identity string.

Algorithm 1 Identity String Generation Using SHA-256

Input: $\mathbb{A}_{\mathcal{K}}$: User's selected personal attribute keys, $\mathbb{A}_{\mathcal{V}}$: User's selected personal attribute values
Output: $H_{identity}$: User's identity string (sha-256 hash)

```
GenerateIdentityString($\mathbb{A}_{\mathcal{K}}$, $\mathbb{A}_{\mathcal{V}}$):
if $|\mathbb{A}_{\mathcal{K}}| == 0 \vee |\mathbb{A}_{\mathcal{V}}| == 0$ then
    return
end
for $(K, V) \in (\mathbb{A}_{\mathcal{K}}, \mathbb{A}_{\mathcal{V}})$ do
    $\mathcal{K}_{\mathcal{C}} \leftarrow \mathcal{K}_{\mathcal{C}} + K$
    $\mathcal{V}_{\mathcal{C}} \leftarrow \mathcal{V}_{\mathcal{C}} + V$
end
$\mathcal{B}_{\mathcal{C}} \leftarrow SHA256(\mathcal{K}_{\mathcal{C}} + \mathcal{V}_{\mathcal{C}})$
$H_{identity} \leftarrow ConvertByteArrayToString(\mathcal{B}_{\mathcal{C}})$
```

D. ZKP Proof Generation and Verification

This section elucidates the ZKP proof generation and verification mechanism. The process, visualized in Figure 5, commences when the user triggers a proof generation request within the GM. In response, the GM liaises with the blockchain client application to obtain the proving key and identity string, enabling the subsequent creation of the ZKP proof. Once generated, this proof is dispatched to the user's email.

When a user attempts to register on an OSN platform, they input this ZKP proof along with the identity string (as shown in Figure 6). The OSN then cross-references the provided proof with its database. If the proof isn't previously recorded (indicating a new user), the OSN forwards it to the blockchain client for verification. The client scrutinizes the timestamp consistency and the

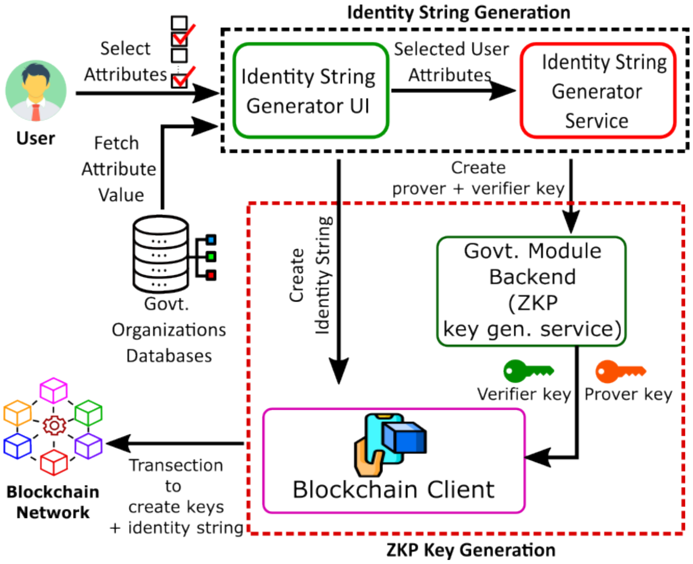

Identity String Generation
Select Attributes
User
Identity String Generator UI
Selected User Attributes
Identity String Generator Service
Fetch Attribute Value
Create prover + verifier key
Create Identity String
Govt. Organizations Databases
Govt. Module Backend (ZKP key gen. service)
Verifier key
Prover key
Blockchain Network
Transection to create keys + identity string
Blockchain Client
ZKP Key Generation

Figure 4: Process of Identity String Generation and ZKP Key Generation Phase

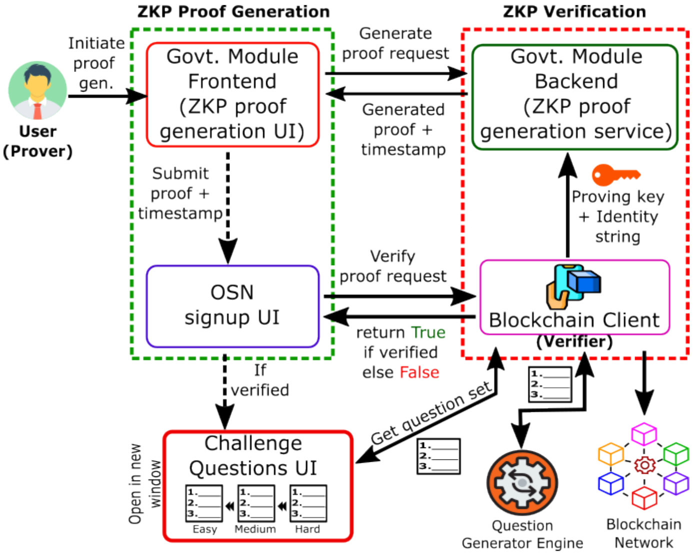
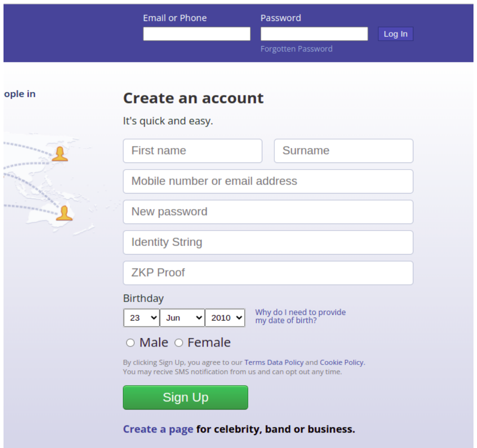


Figure 5: ZKP Proof Generation and Verification Phase

proof's validity using the verifying key. Upon successful verification, the blockchain client relays pertinent user attributes back to the OSN. The OSN ensures the receipt of all necessary attributes before finalizing the account creation; otherwise, registration is withheld.

E. Challenge Question Generation and OSN Signup

Once the proof has been successfully verified, the Blockchain Client (BC) application extracts the user's data associated with the identity string from the blockchain. Additionally, it sources more detailed information from governmental databases (like SSA, DMV, DPH, etc.) using the primary identifiers retrieved from the blockchain, such as the user's SSN, birth certificate number, and driver's license number. Using this comprehensive dataset, the Challenge Question Generator Engine (CQGE) crafts challenge questions across three distinct difficulty tiers: Easy, Medium, and Hard. The generation mechanics are bolstered by algorithms and APIs from the IBM PrimeQA framework, visualized in Figure 7.


Email or Phone
Password
Log In
Forgotten Password

ople in

Create an account

It's quick and easy.

First name
Surname
Mobile number or email address
New password
Identity String
ZKP Proof

Birthday
23 Jun 2010
Why do I need to provide my date of birth?

○ Male ○ Female

By clicking Sign Up, you agree to our Terms Data Policy and Cookie Policy.
You may recive SMS notification from us and can opt out any time.

Sign Up

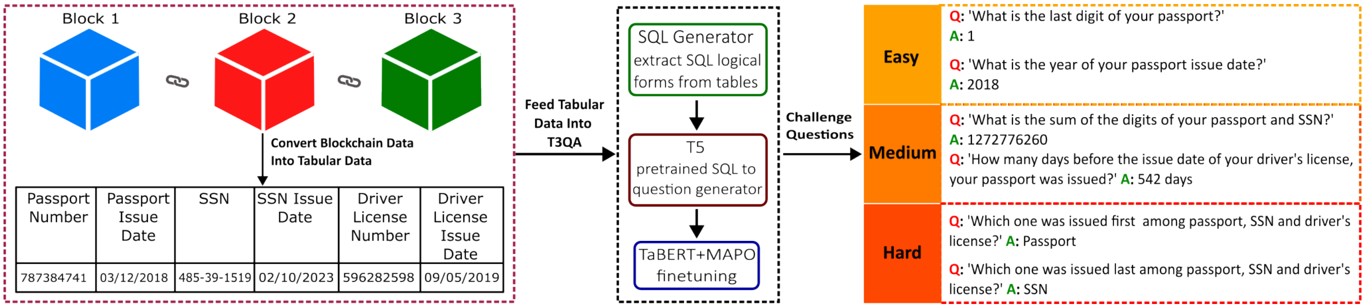
Create a page for celebrity, band or business.

Figure 6: OSN Signup UI. The UI Includes Two Extra Input Fields for the Identity String and ZKP Proof.

On completion of proof validation, the OSN is provided with a URL that displays the challenge questions in an isolated window. This separate window is designed to prevent the OSN from accessing potentially sensitive user data. Authentic users are uniquely positioned to answer these questions correctly. Questions are presented progressively: starting with the easiest tier. Successfully answering the majority of initial questions grants access to the next tier of medium-difficulty questions. Sufficiently answering these enables the participant to progress to the final set of more challenging questions, where a substantial number of correct answers is necessary for progression. Each challenge question offers ten potential answers to heighten the complexity of the task. If a user successfully navigates all tiers, the OSN is notified of their success, paving the way for account creation.

F. Challenge Question Generator Engine

Our CQGE is fundamentally designed based on the t5-base-table-question-generator model, a contribution by Chemmengath et al. [23], as part of the IBM PrimeQA framework, an advanced version of the T5-base model by Raffel et al. [24]. This model efficiently generates questions from tables with column headers and cell values. When the Blockchain Client (BC) application retrieves user data from the blockchain network, it is relayed to the CQGE and converted into a structured tabular format, utilizing an information extractor to delineate the user data for table header templates.


Block 1
Block 2
Block 3
Convert Blockchain Data Into Tabular Data
Feed Tabular Data Into T3QA
SQL Generator extract SQL logical forms from tables
T5 pretrained SQL to question generator
TaBERT+MAPO finetuning
Challenge Questions
Easy
Medium
Hard

Passport Number	Passport Issue Date	SSN	SSN Issue Date	Driver License Number	Driver License Issue Date
787384741	03/12/2018	485-39-1519	02/10/2023	596282598	09/05/2019

Easy

Q: 'What is the last digit of your passport?'
A: 1

Q: 'What is the year of your passport issue date?'
A: 2018

Medium

Q: 'What is the sum of the digits of your passport and SSN?'
A: 1272776260
Q: 'How many days before the issue date of your driver's license, your passport was issued?' A: 542 days

Hard

Q: 'Which one was issued first among passport, SSN and driver's license?' A: Passport

Q: 'Which one was issued last among passport, SSN and driver's license?' A: SSN

Figure 7: Architecture of Challenge Question Generator Engine

G. Hyperledger Fabric as Private Blockchain Database

We utilize Hyperledger Fabric for its security and scalability, ideal for governmental entities. This permissioned blockchain offers tailored control and robust intra-network interactions, facilitating specific block data requests and transaction verification. Our blockchain client application ensures secure interactions and data integrity in line with government standards. While OSNs can only read, the GM has both read and write privileges, enhancing the system's security and integrity.

VII. Security Analysis

This subsection deliberates on the countermeasures and strategies embedded within our system to mitigate the vulnerabilities and attacks outlined in our Threat Model.

- **Identity Provider Attack:** The system thwarts adversaries from acquiring user accounts in the government module by sending identity strings to registered emails only, thereby preventing unauthorized Zero-Knowledge Proof (ZKP) generation.
- **Fake Identity String Attack:** Our approach, by generating identity strings and ZKP proofs concurrently, ensures the impossibility of forging a deceptive identity string.
- **Man in the Middle Attack:** Our method employs public-key encryption, securing the payloads and preventing adversaries from capturing genuine identity strings and computed ZKP.
- **Replay Attack:** The incorporation of a timestamp to the proving key validates the proof within a predetermined period, thereby negating any outdated proofs and securing the verification process.
- **Denial of Service Attack:** The system mitigates potential service disruptions through rate limiting, capped at 100 requests per minute, in both government module backend and blockchain client application.

VIII. Experimental Evaluation

A. Experimental Setup

To assess the efficacy of our ZKP signup system, built upon the principles of Zero-Knowledge Proofs and challenge questions and leveraging the Hyperledger Fabric framework, we develop a prototype and conduct a suite of experiments to analyze its performance.

1) *Development Environment:* Our experimental setup includes prototypes for both a Government (Govt.) module and an Online Social Network (OSN). The backend services for these modules are developed using Spring Boot, while ReactJS is employed for frontend development. The Blockchain Client is developed in Java, utilizing the Hyperledger Fabric Gateway (version 1.3.0). Chaincode is developed with Hyperledger Fabric Chaincode version 2.2.x. The entire blockchain network is constructed on Hyperledger Fabric v2.5.x and its performance is assessed using the Hyperledger Caliper benchmark tool.

2) *Blockchain Network Configuration:* The blockchain network comprises four organizations: the Govt. module organization, SSN organization, Driver license organization, and Birth Certificate organization, with an additional configuration of an Orderer service. Each organization incorporates a committing peer functioning as an endorser, a certificate authority, and a state database (CouchDB). These endorsing peers have the capacity to endorse transaction proposals originating from blockchain clients.

3) *Testbed Configuration:* The experimental test network is implemented and modified on an Ubuntu 22.04 operating system, equipped with an Intel i7-10700 CPU @2.90GHz and 32GB DDR4 RAM. The experiments are designed to meticulously analyze the performance and scalability of the system under varying conditions.

B. Hyperledger Fabric Network

We install Hyperledger Fabric's latest version (2.5.x) on Ubuntu. Then start with the test network provided with the fabric samples. Then modify the test network configuration as mentioned in Section VIII-A. We write chaincode (version 2.2.x) to read and update (we define a block data structure for user details) fabric network. The attributes of the user detail object are *ID* (we use birthcertificate number as ID), *IdentityList*, *SSNDetails*, *DriverLicenseDetails* and *BirthCertificateDetails*. *IdentityList* will contain a list of identities generated from the Govt. module dashboard. *SSNDetails* will contain SSN-related data which will be submitted by SSN authority to the network. *DriverLicenseDetails* will contain the driver license-related data submitted by the Department of Motor Vehicle (DMV) to the network. *BirthCertificateDetails* will contain birth certificate-related data submitted by the Department of Public Health.

C. Performance Evaluation

In this subsection, we evaluate the performance of the ZKP encryption process, challenge question generator module, and read/write from/to the blockchain network. To measure the performance of the ZKP encryption process we conducted a series of ZKP proof generation and proof verification for different sets of user attributes. We perform benchmark tests using Hyperledger Caliper [25] with varying settings for transaction send rates and endorsement policies. We experiment with endorsement policies of 1-of-any, 2-of-any, and 3-of-any. The transaction send rate defines the rate at which transactions are input to the blockchain network. An endorsement policy defines the number of peers required to reach an agreement on the result of a transaction before committing to the ledger.

1) ZKP Performance: Since we use hashed (SHA-256, length 256 bit) identity string as the message to be encrypted by the ZKP encryption process, we get the same time complexity. Although the user may select a variable number of user attributes (for different OSNs signup) to generate an identity string from the Govt. module dashboard. We found an average time complexity of 26ms to generate ZKP proof.

2) Challenge Question Generator Performance: We utilize the IBM PrimeQA framework for the question generation step. Before passing to the PrimeQA table question generator (generates questions from tabular data) we convert the data received from the blockchain into a tabular format. leftmost box of Figure 7 shows the conversion from blockchain data to tabular format. We conducted a series of this data conversion and find variable time complexity since data volume can be different for different users. For example, some users may have only Birth Certificate and SSN. So question generator needs to convert these 2 types of data into tabular data. If someone has a birth certificate, SSN, and driver's license then the question generator will need to convert 3 types of data into tabular data.

3) Experimental Result of User Signup: In our study, we conducted a signup experiment using our developed prototype. We invited a group of 50 participants to engage in the ZKP signup process with our system. Initially, users created accounts in the Government module and generated a ZKP proof. Subsequently, they attempted to signup on the OSN. Prior to the experiment, we stored synthetic user details. We observed a 100% success rate in the ZKP signup process, indicating that no participant could circumvent the challenge questions or establish a counterfeit account using falsified details. All genuine attempts culminated in successful account creation.

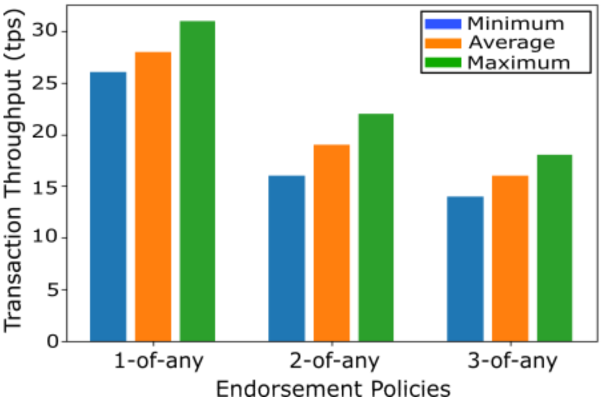


Figure 8: Minimum, Average & Maximum Transaction Throughput vs. Hyperledger Fabric Endorsement Policies.

4) Transaction Throughput: The transaction throughput is the measure of the flow rate of processed transactions through the blockchain network. It is measured in the number of transactions per second (tps). We experiment using endorsement policies 1-of-any, 2-of-any, and 3-of-any. We see the decrease in tps with the increase of endorsing peers, as the increase in number of peers requires more time for the endorsement process. The results of the endorsement policies versus transaction throughput experiment are depicted in Figure 8.

5) Transaction Latency: Transaction latency is quantified as the duration between the initiation of a request-response from the blockchain client and the moment it is committed into the ledger. An increase in endorsement policies correlates to elevated transaction latency; this is attributed to the additional time required for processing the endorsement task by the increasing number of peers involved. We conducted experiments using the blockchain client APIs with a variety of read/write requests and observed an average latency of below 1 second.

D. Scalability Analysis of Hyperledger Fabric

Decentralized Applications (DApps) operate on distributed ledgers and have become prevalent in various domains, including supply chain management [26], auto insurance claims [27], and decentralized voting [28]. These applications exploit decentralization to mitigate the risks inherent to centralized systems.

1) Throughput Analysis: Even considering the lowest throughput from our experiments, which averaged 16 transactions per second (tps), the system is capable of handling the maximal signup request rate as compared to platforms like Facebook—one of the largest Online Social Networks (OSNs), which experiences around 500,000 new users every day or approximately 6 new accounts every second [29].

2) Enhancement Strategies: To further enhance scalability, strategies proposed by [30] can be employed. Multiple other strategies exist to address blockchain scalability, divided broadly into Layer1 and Layer2 solutions. Layer1 solutions include blockchain sharding, optimized consensus algorithms, and forking, while Layer2 solutions encompass off-chain transactions, sidechains, and the creation of child chains. Additionally, a hybrid approach combining Layer1 and Layer2 techniques can also be employed to realize both high throughput and robust security.

IX. Conclusion

In this paper, we introduce an innovative privacy-preserving signup process utilizing ZKP principles, fortified by an additional layer of challenge questions. We utilize Hyperledger Fabric, a private blockchain, to ensure secure data storage. Our approach facilitates account creation through real identity verification, negating the need for users to share sensitive information with OSNs. We've constructed a prototype and conducted thorough experiments to gauge its effectiveness in preserving privacy during account creation, also measuring time complexity to estimate the system's user usability.

X. Acknowledgment

This research was supported by NSF grant CNS-2153482.

References

[1] Oana Goga, Giridhari Venkatadri, and Krishna P Gummadi. The doppelgänger bot attack: Exploring identity impersonation in online social networks. In *Proceedings of the 2015 internet measurement conference*, pages 141–153, 2015.

[2] Sajedul Talukder and Bogdan Carbunar. A study of friend abuse perception in facebook. *ACM transactions on social computing*, 3(4):1–34, 2020.

[3] Sajedul Talukder and Bogdan Carbunar. Abusniff: Automatic detection and defenses against abusive facebook friends. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12, 2018.

[4] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. volume 18, pages 186–208. 1989.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized business review*, page 21260, 2008.

[6] Markus Jakobsson and Ari Juels. Proofs of work and bread pudding protocols. In *Secure Information Networks: Communications and Multimedia Security IFIP TC6/TC11 Joint Working Conference on Communications and Multimedia Security (CMS'99) September 20–21, 1999, Leuven, Belgium*, pages 258–272. Springer, 1999.

[7] Fahad Saleh. Blockchain without waste: Proof-of-stake. *The Review of financial studies*, 34(3):1156–1190, 2021.

[8] Daniel Larimer. Delegated proof-of-stake (dpos). *Bitshare whitepaper*, 81:85, 2014.

[9] Elli Androulaki, Artem Barger, Vita Bortnikov, Christian Cachin, Konstantinos Christidis, Angelo De Caro, David Enyeart, Christopher Ferris, Gennady Laventman, Yacov Manevich, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In *Proceedings of the thirteenth EuroSys conference*, pages 1–15, 2018.

[10] Ying-Ho Liu and Chia-Yu Kuo. Simaim: identifying sockpuppets and puppetmasters on a single forum-oriented social media site. *The Journal of Supercomputing*, pages 1–32, 2023.

[11] Tianlin Song, Jingqiang Lin, Wei Wang, and Quanwei Cai. Traceable revocable anonymous registration scheme with zero-knowledge proof on blockchain. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[12] Xiaohui Yang and Wenjie Li. A zero-knowledge-proof-based digital identity management scheme in blockchain. *Computers & Security*, 99:102050, 2020.

[13] Xavier Salleras and Vanesa Daza. Zpie: Zero-knowledge proofs in embedded systems. *Mathematics*, 9(20):2569, 2021.

[14] Bhavye Sharma, Raju Halder, and Jawar Singh. Blockchain-based interoperable healthcare using zero-knowledge proofs and proxy re-encryption. In *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, pages 1–6. IEEE, 2020.

[15] Tao Feng, Pu Yang, Chunyan Liu, Junli Fang, and Rong Ma. Blockchain data privacy protection and sharing scheme based on zero-knowledge proof. *Wireless Communications and Mobile Computing*, 2022:1–11, 2022.

[16] Armin Anvari, Lei Pan, and Xi Zheng. Generating security questions for better protection of user privacy. *International Journal of Computers and Applications*, 42(4):329–350, 2020.

[17] Radha Dhekane. *Towards a usable fallback authentication mechanism*. PhD thesis, California State University, Sacramento, 2020.

[18] Nicholas Micallef and Nalin Asanka Gamagedara Arachchilage. Understanding users' perceptions to improve fallback authentication. *Personal and Ubiquitous Computing*, 25(5):893–910, 2021.

[19] Wanxin Li, Collin Meese, Hao Guo, and Mark Nejad. Blockchain-enabled identity verification for safe ridesharing leveraging zero-knowledge proof. In *2020 3rd International Conference on Hot Information-Centric Networking (HotICN)*, pages 18–24. IEEE, 2020.

[20] Deebthik Ravi, Sashank Ramachandran, Raahul Vignesh, Vinod Ramesh Falmari, and M Brindha. Privacy preserving transparent supply chain management through hyperledger fabric. *Blockchain: Research and Applications*, 3(2):100072, 2022.

[21] Charalampos Stamatellis, Pavlos Papadopoulos, Nikolaos Pitropakis, Sokratis Katsikas, and William J Buchanan. A privacy-preserving healthcare framework using hyperledger fabric. *Sensors*, 20(22):6587, 2020.

[22] spring.io. *BCryptPasswordEncoder Documentation*. docs.spring.io, 2022.

[23] Saneem Ahmed Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Jaydeep Sen, Mustafa Canim, Soumen Chakrabarti, Alfio Gliozzo, and Karthik Sankaranarayanan. Topic transferable table question answering. *arXiv preprint arXiv:2109.07377*, 2021.

[24] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[25] Hyperledger. *Measuring Blockchain Performance with Hyperledger Caliper*. Hyperledger, 2018.

[26] Vishal Naidu, Kumaresan Mudliar, Abhishek Naik, and Prasenjit Bhavathankar. A fully observable supply chain management system using block chain and iot. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–4. IEEE, 2018.

[27] Chuka Oham, Raja Jurdak, Salil S Kanhere, Ali Dorri, and Sanjay Jha. B-fica: Blockchain based framework for auto-insurance claim and adjudication. In *2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData)*, pages 1171–1180. IEEE, 2018.

[28] Jollen Chen. Devify: Decentralized internet of things software framework for a peer-to-peer and interoperable iot device. *ACM SIGBED Review*, 15(2):31–36, 2018.

[29] Brian Dean. *Facebook Demographic Statistics: How Many People Use Facebook in 2023?* Backlinko, 2023.

[30] Christian Gorenflo, Stephen Lee, Lukasz Golab, and Srinivasan Keshav. Fastfabric: Scaling hyperledger fabric to 20 000 transactions per second. *International Journal of Network Management*, 30(5):e2099, 2020.