The ISC Int'l Journal of Information Security
ISeCure

November 2022, Volume 14, Number 3 (pp. 33–42)
http://www.isecure-journal.org

Presented at the ISCISC'2022 in Rasht, Iran.

On the Suitability of Improved TrustChain for Smartphones ☆

Seyed Salar Ghazi [1], Haleh Amintoosi [1,*], and Sahar Pilevar Moakhar [1]
[1] *Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran.*

ARTICLE INFO.

Keywords:
Distributed Systems, Blockchain, TrustChain, Distributed Cloud, Whitewashing, Battery Consumption

Type:
Research Article

doi:
10.22042/isecure.2022.14.3.4

dor:
20.1001.1.20082045.2022.14.3.4.8

ABSTRACT

In recent years, blockchain technology has been used in many fields, including IoT and Smartphones. Since most of these devices are battery constrained and have low processing capabilities, conventional blockchains are not suitable for these types of systems. In this field, critical challenges that need to be addressed are providing security for transactions and power consumption. An available solution to meet the mentioned challenges is TrustChain. Unlike conventional blockchains, TrustChain does not have a single global chain. Instead, each node is responsible for building and maintaining its local chain. With all the benefits, TrustChain is vulnerable to the whitewashing attack and suffers from client vulnerability issues. Moreover, once a fatal error occurs, the recovery time of each TrustChain node is considerably high. In this paper, we propose a solution to address the attacks mentioned above by implementing an authentication system with MongoDB on top of TrustChain. Moreover, we connected TrustChain to the distributed cloud storage to significantly reduce the recovery time of nodes in fatal errors (up to 80%). Finally, we evaluate improved TrustChain with the PoW-based smartphone-oriented blockchains from two aspects of security and power consumption, proving that improved TrustChain does not significantly affect the lifetime of the smartphone battery. Its power consumption is less than mentioned blockchains and is more secure than these systems against main attacks.

1 Introduction

In recent years, blockchain technology has been used in many fields, including machine learning, cloud computing, e-commerce, e-voting, and the Internet of Things (IoT). This technology is distinguished from other distributed data structures' independence from a trusted third party. Another advantage is that its records are non-manipulative and indelible. However, scalability, security threats, and the need for a consensus mechanism have posed challenges. Consensus mechanisms are often very energy-intensive and are the main bottlenecks in using this technology. The most well-known consensus mechanism is the Proof-of-Work (PoW) introduced with Bitcoin [1]. Reconstructed blockchain systems have been proposed to

* Corresponding author.
☆ The ISCISC'2022 program committee effort is highly acknowledged for reviewing this paper.
Email addresses: seyedsalar.ghazi@mail.um.ac.ir, amintoosi@um.ac.ir, sahar.pilevarmoakhar@mail.um.ac.ir
ISSN: 2008-2045

avoid the traditional blockchain's high latency and low scalability. These systems were created based on Directed Acyclic Graph (DAG). Examples are COTI, IOTA, NANO, Fantom, and TrustChain. In these systems, each node is responsible for building and maintaining its chain and monitoring other network nodes' behavior. The nodes involved in this process are called monitor nodes [2, 3]. In the TrustChain system, specifically designed for smartphones, the task of monitor nodes is to maintain the network's consistency and ensure its security and integrity. TrustChain does not have a consensus mechanism; instead, it uses an accounting mechanism called NetFlow, which makes it resistant to the Sybil attack. In the Sybil attack, the attacker subverts the service's reputation system by creating multiple pseudonymous identities and using them to gain a disproportionately large influence. TrustChain is also resistant to the Replay attack and Double-spending attack. Double-spending is a potential flaw in a digital cash scheme in which the same single digital token can be spent more than once, and the Replay attack is a form of network attack in which valid data transmission is maliciously or fraudulently repeated or delayed [2]. Despite all these advantages, TrustChain is vulnerable to whitewashing and client vulnerability issues. A whitewashing attack occurs when an attacker resets a poor reputation by rejoining the system with a new identity. Client vulnerability is an issue that leads to the loss of users' personal information on the network [4]. Moreover, the TrustChain node's recovery time is considerably high for fatal errors. In the fatal errors, the TrustChain application has been stopped, its cache has been cleared, and finally, this node has to update its local chain from other network nodes and Bootstrap servers. As mentioned above, TrustChain has been specifically designed for smartphones. Some blockchains, such as uPlexa and MIB, integrate the PoW algorithm for mobile devices [5, 6]. However, these blockchains face challenges such as high battery consumption for cryptographic operations [7], due to the smartphones' limited battery supply and constrained hardware specifications. In this paper, we extend TrustChain to address its challenges and shortcomings with the following methods:

- We empower TrustChain with distributed cloud storage to significantly reduce the recovery time of each node when a fatal error occurs. Considering that most smartphones suffer from a lack of resources such as memory and CPU, the probability of errors has increased for such types of devices. This necessitates the use of distributed cloud storage as it facilitates mass and rapid data storage [8].
- We add a distributed authentication system with MongoDB to the TrustChain that prevents whitewashing attacks and client vulnerability issues, both of which are serious for smartphones.
- Finally, we evaluate the improved TrustChain platform from the aspects of security and battery consumption for smartphones and prove its superiority over PoW-based blockchains such as uPlexa and MIB.

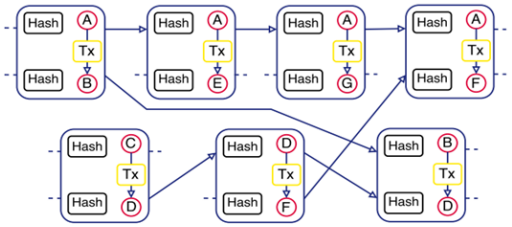

Figure 1. The architecture of TrustChain [2]

The remainder of the paper is as follows. Section 2 discusses several relative works and required backgrounds of TrustChain. The proposed method is presented in Section 3. Results are discussed in Section 4, and finally, Section 5 includes conclusions and future works.

2 Background and Related Work

2.1 TrustChain

Traditional blockchains like bitcoin maintain one single chain containing the history of all transactions carried out by users. The TrustChain architecture is however different in the sense that every participant is responsible for building and maintaining their local chain. Another difference is that in common blockchains like bitcoin, multiple transactions are stored in one block to increase scalability. However, in TrustChain, each block packs only one transaction [2]. The architecture of TrustChain is shown in Figure 1. As can be seen, each block in TrustChain has two incoming and two outgoing pointers. An extra pointer to the other party's chain makes it difficult to reorder or delete blocks in one chain because it can be identified by the other party involved in a transaction. This mechanism will add an extra security layer on top of the TrustChain network. Before a block is inserted into the local storage of each node, validation of blocks is executed. During this stage, sequence numbers, incoming and outgoing pointers, signatures, and transaction data are checked. Only if a block is marked as valid, it is appended to the local chain and shared with other network nodes. TrustChain can be considered as a mechanism by which consensus among participants on a particular transaction is reached rather than on a global consensus. TrustChain blocks interchange by the agents

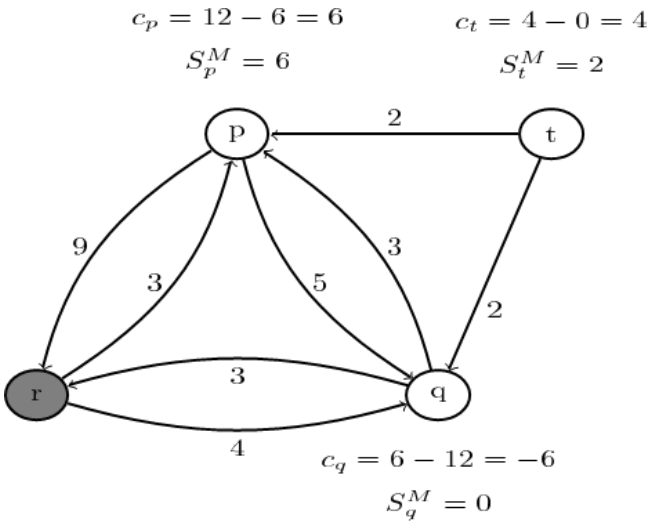

Figure 2. NetFlow computation performed from the perspective of agent r [2]

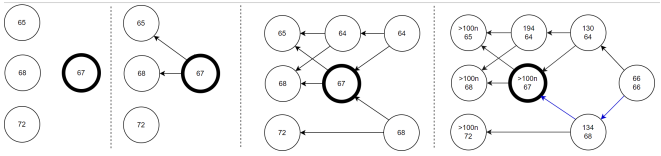

Figure 3. The life cycle of new transactions [7]

using gossiping and widespread replication. Therefore, in TrustChain, each agent publishes its unique chain, monitors the interactions of other nodes, and collects TrustChain data to calculate trust levels [2]. The point that makes this architecture an advantage over a traditional blockchain is that in the traditional blockchain, the double-spending attack is prevented at the cost of increased computational overhead, due to the need for global consensus. TrustChain eliminates such overhead by not relying on the global consensus and at the same time, enabling only the detection of the double-spending attack [2]. As mentioned earlier, TrustChain has no consensus mechanism and instead uses an accounting mechanism called NetFlow, which makes it resistant to the Sybil attack. NetFlow uses the TrustChain graph as input. An interaction graph involving four agents has been shown in Figure 2. In this graph, agent r has contributed a total of 8 units of work to agent q. The NetFlow algorithm assigns a particular score to each node using maximum flow which represents the maximum units of work that can flow through an agent in the interaction graph [2]. Figure 2 shows NetFlow computation performed from the perspective of agent r. The interaction graph G_r has four agents p, q, r, and t. In (1), the NetFlow algorithm assigns the node capacities (c_j) for each agent in the network and, $MF_{G_i}(j,i)$ is the maximum flow from j to i. Note that, the weights of the edges are contributed works from j to i. In (2), $s_P^M(score)$ is calculated by NetFlow and assigned to the node that, in this case, is equal to 3. It means that at most 3 units of work can flow through node p in the interaction graph. G_i^N in (2), is the graph G_i modified with c_j as node capacities for each agent. For example, the maximum flow from node p to r is 12 and the maximum flow from node r to p is 9. This means that the node capacity of p (c_p) is max (12 - 9, 0) = 3. On the other hand, it proves that NetFlow makes TrustChain resistant to the Sybil attack [2].

$$c_j = max\{MF_{G_i}(j,i) - MF_{G_i}(i,j), 0\} \quad (1)$$

$$s_j^M(G_i, C_i) = MF_{G_i^N}(j,i) \quad (2)$$

A critical bottleneck in TrustChain is the calculation of the maximum flow. One of the well-known algorithms is Ford-Fulkerson with the computational complexity of $O(m|f|)$ where n denotes the number of nodes, m denotes the number of edges, and $|f|$ is the maximum flow magnitude. Another algorithm used for the calculation of the maximum flow is the Dinitz algorithm. If implemented with the dynamic tree, this algorithm can reduce the complexity to $O(nm\log(n))$. It should be noted that an algorithm of $O(nm)$ is also presented in the article [2]. Codes of the TrustChain are located in [9].

2.2 Related Work

As TrustChain is a DAG-based blockchain, in this section, we first inspect other similar DAG-based and reputation-based projects. Then, we study the works on converging blockchain and smartphones. Finally, we review the articles related to converging blockchain and cloud computing.

2.2.1 DAG-Based & Reputation-Based Blockchains

A closely related work to TrustChain is R3 Corda [10]. Similar to TrustChain, it avoids traditional consensus mechanisms, PoW, and Fork. The main difference between these two technologies is a central element of R3 that binds contracts between two parties and regulatory adoption. Another key difference is that R3 has no widespread replication of transactions and avoids gossiping because transaction records are only saved by the two or more directly involved parties [10].

COTI is another DAG-based blockchain that is similar to TrustChain [11]. In the COTI network, each user's reputation level is assigned to him based on a combination of behavioral data and objective information using a unique machine learning algorithm. The level of reputation allocated to users is used to validate and verify transactions. In this case, each transaction is linked to two previous transactions with the

same degree of trust. This motivates users to increase their level of trust. COTI has established mechanisms for monitoring, detecting, and defending against potential attacks such as double-spending [11]. Figure 3 shows the life cycle of a new transaction in the COTI cluster. At first, a new transaction is proposed to the network. Then, it is added to the cluster by validating two precedent transactions with trust scores like its own. In the third step, other transactions validate it, and eventually, it is confirmed and attached to the cluster once the accumulative trust score of the heaviest path confirms it exceeds the set threshold [11]. The next project which uses DAG as its blockchain data structure is Avalanche [12]. It is a permissionless system based on a new type of approach that deviates from the BFT mechanism (Byzantine Fault Tolerance). Avalanche uses the protocol called Slush which is a CFT mechanism (Crash Fault Tolerant). It also uses concepts of gossip algorithms and epidemic networks. In Avalanche, the Snow protocol family has been introduced that is based on a Mobile network sampling mechanism. In this mechanism, there is no need to agree on the precise membership of the system and nodes can reach low latency and high throughput via it [12]. In [3], other DAG-based blockchains have been reviewed, compared with each other, and analyzed regarding performance and security. J. Yu *et al.* introduced RepuCoin [13] to address the majority attack. Based on the authors' claim, RepuCoin is the first work that guarantees the efficiency of the entire network even if for a short time the attacker can provide more than 50% of the processing power. In this system, the score of each user is determined according to his reputation in the network and during his activity in the whole blockchain. In [14], the authors proposed a reputation-based crowdsourcing framework at the top of a blockchain system (Hyperledger Fabric). They first set up a blockchain-based platform to carry out the management of crowdsourcing trading and user-reputation evaluating works. A reputation model then computes the reputation values of contributors and discloses any harmful action. Finally, they use queueing theory to evaluate the blockchain-based framework and optimize system efficiency.

2.2.2 Blockchain for Smartphones

Blockchains that exist in this field are divided into three categories: 1) Proof of Work systems, 2) Proof of Transaction systems, and 3) Cloud systems which we will explain in the following [7]. MIB is one of the PoW blockchains that has almost twenty thousand active users. MIB is an environmentally friendly, low-cost PoW mining algorithm based on Bitcoin. Note that, MIB's Mobile Proof-of-Work uses CPU resources for block mining, and to avoid the device overheating, users can opt for proper mining complexity. It is important to note that the MIB network is centralized. The network servers allocated by the project owners distribute the cryptographic tasks between the nodes and then propagate the blocks. Therefore, mobile devices do not support a distributed network and network control responsibility remains with the MIB administrator [6]. Another PoW-based blockchain that has been customized for mobile devices is uPlexa. uPlexa's main purpose is to create a platform for IoT devices to form an anonymous blockchain-based payment system. uPlexa also relies on the PoW concept in which, the CPU or GPU of IoT devices are used to add transactions to the ledger. As we know, the two main problems of Bitcoin are slow transaction time and hefty fees. uPlexa overcomes these issues by introducing a model where micropayment fees increase when the network is overloaded [5]. Smartphone-based blockchain applications may rely not only on computational resources but also on the transfer of value between the network nodes. An example of this system is TAU coin [7]. The algorithm used in this project is called Proof-of-Transaction (PoT), which determines a new block generation address based on accumulated transaction history. For each node, there is a linear proportionality between the probability of generating a new block and the node's transaction history, called mining power. A greater number of transactions made by a node is equivalent to a higher probability of receiving rewards. It should be noted that TAU has been secured against 51% attack [7]. Cloud mining is another type of system for smartphones, that enables users to contribute to the network regardless of their hardware and knowledge about details (e.g., mining algorithms). In this system, a remote data center has been used for token mining. This concept has been utilized in Electroneum and Phoneum [7], both of which are compatible with the Know Your Customer (KYC) process, meaning that platforms force users to fill out legal names, surnames, and other identifying information and upload their photo to verify their identity. This, however, may threaten users' privacy since the information is shared with third parties. Although in these projects, the smartphones do not carry out any activity and do not help confirm new blocks [7].

2.2.3 Blockchain Meets Cloud Computing

In [15], the authors proposed a cloud-oriented blockchain called CloudChain, which has three layers containing a network layer, a consensus layer, and a blockchain layer. In CloudChain, nodes' communication is synchronous and is based on direct memory access and the shared-memory model. The authors in this work proposed a shared-memory consensus algorithm that satisfies persistence and liveness and

proved its feasibility and efficiency. Kapil Aggarwal *et al.* integrated Inter Planetary file system (IPFS) with blockchain technology for achieving high-performance security and anonymity-based transactions [16]. In this work, the IPFS protocol has been used in blockchain for transferring data via secured channels to avoid hacking attempts. In [17], the authors proposed a comprehensive survey of blockchain-based cloud services publications. The authors classified articles into three categories: (1) blockchain-based Infrastructure-as-a-Service (2) blockchain-based Platform-as-a-Service, and (3) blockchain-based Software-as-a-Service. They then identified state-of-the-art works in blockchain-based storage-as-a-service, VNF-as-a-service, microservice-as-a-service, computation-as-a-service, and data aggregation-as-a-service. Finally, they explained current and potential challenges and future work directions. A remarkable point about the reviewed DAG-based and reputation-based works is that none of them are designed to be used in smartphones and on the other hand, none of the blockchains for smartphones are DAG-based and reputation-based. TrustChain is the only system that has both features at the same time. In the following, we combine TrustChain with the Storj distributed cloud and add a distributed authentication system with MongoDB to it.

3 Improved TrustChain

3.1 Security Improvements

As mentioned earlier, the permissionless nature of TrustChain allows anyone to create a new identity for himself/herself without spending much cost. If a node suffers from a lack of credibility in the network, it can easily leave the previous identity and start working again in the system with the new identity. This process is called whitewashing.

On the other hand, client vulnerabilities are a set of vulnerabilities that can potentially cause the user to lose personal information, including IP addresses, personal chains, or native tokens, leading to wallet theft, crypto-jacking, and even eclipse attacks. In the eclipse attack, the attacker seizes all incoming and outgoing connections of the victim node and disconnects them from the network [4].

To address these attacks, we propose the use of a distributed authentication system for TrustChain. Note that this system is at the top of TrustChain. The user will not be allowed to log in, if not authenticated. Specifically, we use the No-SQL MongoDB and its Sharding feature as it enables storing biometric information of individuals and nodes in the database in a distributed manner. Using the Sharding feature enables data distribution among multiple machines. MongoDB uses Sharding to support deployments with large datasets and high throughput operations [18]. In this work, we use the mentioned feature to distribute users' fingerprint information across multiple devices which causes the highest availability and larger storage capacity. Consider that each sharded cluster consists of three components: 1) shared data that can be deployed as a replica set, 2) mongo that is a query router, and 3) config server that saves metadata and configuration settings of the cluster. Alongside using this feature, we utilize the Client-Side Field Level Encryption feature of MongoDB for increasing the security levels of this system [19]. It should be noted that large decentralized projects such as the Internet Computer also use fingerprints for authentication [20]. Internet Computer has been called the world's first blockchain that is web-speed and internet-scale which allows smart contracts to run securely on it and use the open governance system called NNS (Network Nervous System). The NNS allows the Internet Computer network to be governed in an open, decentralized, and secure manner. An Internet Identity is required to log in to the NNS D-app. Internet Identity is a modern new blockchain authentication system that is based on advanced cryptography and enables users to sign in to the D-apps using fingerprint sensors [20].

To address the whitewashing attack, we record the time of each user's first login to the network. Specifically, we make changes in peers.kt class to add a property called Register˙time to the Block.kt class. The content of this field is then sent along with other information to other nodes. Now, the system can deal with the whitewashing attack from the passive and active perspectives, as described below. In the passive method, each user can use the authentication information sent by other nodes in the network to identify new users and set a lower priority to interact with them. Such a policy has a considerably low overhead, and the overall performance of the system will be less affected by such a method.

In the active method, new users entering the network at the login time are only allowed to act as monitor nodes until the number of blocks stored in them reaches a certain quorum (which in this paper is static and from experimental data, we considered this quorum to be 1000 blocks). This method will incur more overhead compared to the previous one.

In the traditional TrustChain, since there is no mechanism for authenticating users, in addition to making all user information publicly available, any attacker can log in to the system. The implementation of the authentication system prevents client vulnerability issues as it blocks illegal intrusion into

the system, especially due to the use of biometric information like fingerprints for login. Moreover, with this method, the TrustChain application will also be resistant to wallet theft and crypto-jacking attacks. It will also be protected from illegal and unauthorized access to the system.

3.2 Distributed Cloud Deployment

The purpose of the distributed cloud implementation and connecting it to the TrustChain is to improve the performance of the system in fatal errors. This method will reduce the restart time of the out-of-reached nodes in fatal errors. It should be noted that updating a local chain through the TrustChain system requires more time than the cloud, and imposes more overhead on the system too. Considering that most smartphones suffer from a lack of resources such as memory and CPU, the probability of errors occurring has increased for such types of devices.

To achieve these goals, we have used Storj as a distributed cloud [21]. Storj is a cryptographic-based cloud storage platform that allows any system running specific software to rent unutilized hard space to users intending to save the files. In fact, Storj is a cloud storage platform similar to those presented by Amazon, Google, or Dropbox. However, instead of a company owning and supporting the software, Storj depends on software and a network of computers to manage its data storage. Note that Storj controls its network via a random file verification every hour, which certifies that storage nodes hosted its files frequently. Three main components of the Storj blockchain include the following cases:

- Storage Nodes - Allow users to rent out surplus space on their system and reliably save and retrieve data for payment.
- Uplinks - Run on the client's system and upload files to the network. Uplinks also connect with peers to save and return data.
- Satellites - organize traffic between the storage nodes and uplinks. Satellites are responsible for saving metadata, controlling storage nodes' proper working, and distributing payments. Note that, each user has an account on a satellite.

In Figure 4, The steps for storing and retrieving information are shown on Storj blockchain [21]. The first reason for choosing Storj as a distributed cloud is the amount of free data available for users containing 50 GB of storage space plus 50 GB of network traffic. The second reason is the lower price of the native token of this network compared to similar technologies. The third reason is the lack of Mainnet in this blockchain and its implementation on Ethereum. The

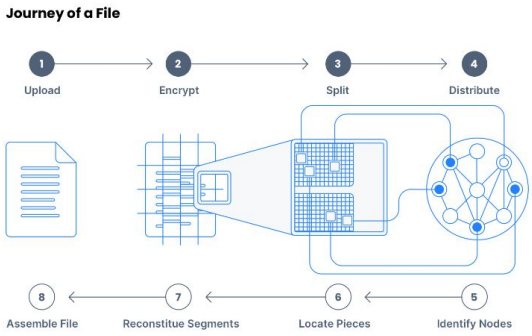

Figure 4. Steps of storing and retrieving information on Storj [21]

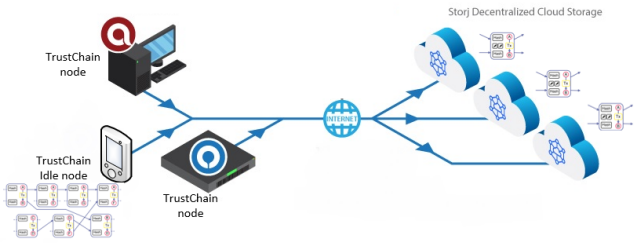

Figure 5. Idle nodes upload their local chain to Storj decentralized cloud

advantages of this method are higher security and reliability of the system, and its disadvantage is system dependency on the Ethereum network [21]. To use Storj, we need to create a bucket via its user interface. After creating this bucket, we must connect the TrustChain system to this distributed cloud using the APIs provided to the developers on the Storj website. Considering that, each node exchanges its chain with the Storj only at intervals where there is no sending and receiving data in the system via that. Figure 5 shows the TrustChain idle node when uploading its local chain to Storj. The results and evaluations of this method and its effects on the recovery time of each node will be explained in the next section.

4 Results and Discussion

Table 1. Specifications of analyzed devices

Device Model	Android Version	SoC	RAM	Battery	Processor
Samsung S9	10	Exynos 9810	4GB	Li-Ion 3000 mAh	4x2.7 GHz Mongoose M 63 4x1.8 GHz Cortex-A55
Samsung Tab A 10.1	8	Exynos 7870	3GB	Li-Ion 7300 mAh	2x1.8 GHz Cortex-A73 6x1.6 GHz Cortex-A53
Nokia 7+	10	Qualcomm SDM 660	4GB	Li-Ion 3800 mAh	4x2.2 GHz Kryo 260 Gold 4x1.8 GHz Kryo 260 Silver
Nokia 7.2	11	Qualcomm SDM 660	6GB	Li-Ion 3500 mAh	4x2.2 GHz Kryo 260 Gold 4x1.8 GHz Kryo 260 Silver
Samsung A12	11	MediaTek MT 6765 Helio P35	3GB	Li-Ion 5000 mAh	4x2.35 GHz Cortex-A53 4x1.8 GHz Cortex-A53

4.1 Evaluation Setup

In this work, we have selected devices from the two categories of smartphones and tablets. Table 1 shows

the corresponding specifications of the analyzed devices. Note that we focused on user-oriented metrics (i.e., battery consumption) and have tried to use devices with different configurations. Overall, all evaluations were performed under the same conditions for all devices, and we inspected the following scenarios for devices:

- Self-discharge with the display ON only (display only): where the devices are left aside with their display ON, without running any application, until they run out of battery.
- With smartphone-oriented blockchains: where Smartphone-based Blockchains such as uPlexa, MIB, as well as TrustChain, and the improved TrustChain are running on the device with the possibility of fatal errors occurring.
- Connectivity options, either LTE or WLAN.

These scenarios cover most of the smartphone's functional states. Since in the cloud-based smartphone-oriented Blockchains, smartphones do not help confirm new blocks and do not perform any tasks, in this work, we only compare the improved TrustChain with PoW-based blockchains, i.e., uPlexa and MIB. The java codes used for the evaluation are located in [22].

4.2 Security Analysis

As we discussed in the previous section, our solution to address whitewashing attacks and client vulnerability issues is the implementation of a decentralized biometric authentication system on the TrustChain network. Simultaneously, we store the time of the first login of each user in the header of packages and send it to other nodes along with other default information. As mentioned in the previous section, in the active approach, new users entering the network at the beginning of the login are only allowed to act as monitor nodes on the network, until the number of blocks stored in them or the Stored˙block feature, reaches a certain quorum (Which in this article is static and equal to 1000 blocks based on experimental data, added to the system as hard code). Although this method incurs higher overhead compared to the passive method, it is more efficient and reliable. Figure 6 shows the time it takes for each device to reach the threshold (1000 blocks) which is set empirically.

Since whitewashing is a kind of Sybil attack and the analysis of Sybil attack is also applicable to whitewashing, we use that for theoretical analysis of the proposed method [23]. In (3), let ω_-^n be the sum of work that agent i has performed for the network after n steps, including work performed before the start of the Sybil or whitewashing attack. Let ω_+^n be the amount of work that agents in i or any of their Sybils or new identities obtained from the network. Any

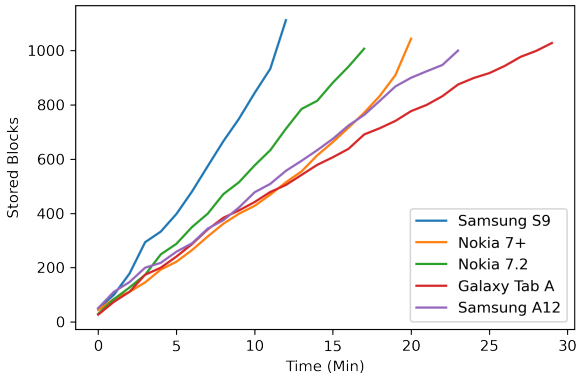

Figure 6. The required time to reach the threshold for each device

work obtained before the start of the Sybil or whitewashing is disregarded. The profit of the Sybil attack is obtained from (3) [2].

$$sup\{(\omega_+^n)/(\omega_-^n) : n \in N, \omega_-^n \neq 0\} \tag{3}$$

If this supremum is infinite, the Sybil or whitewashing attacks are strongly beneficial, if the supremum is finite and is larger than 1, these attacks are profitably weakly beneficial, and if the supremum exists and is smaller than or equal to 1, these attacks are unprofitably weakly beneficial. This case is known as "contributing to the network" [2]. We now prove that the profitability of the whitewashing attack in our proposed method is bounded. As the block storage rate is approximately linear in test devices shown in Figure 6 and burst transactions are prevented in the network, for a certain period, $\omega_+^n \leq \omega_-^n +$ (experimental quorum). Note that in this work, an experimental quorum was taken 1000 blocks. Thus:

$$sup\{(\omega_+^n)/(\omega_-^n + 1000) : n \in N, \omega_-^n \neq 0\} \leq 1 \tag{4}$$

This shows that in this period, the whitewashing attack does not have any benefits for the attacker and will impose additional overhead.

As shown in Figure 6, reaching the threshold of 1000 blocks takes approximately 13 minutes for the Samsung S9 device, 18 minutes for Nokia 7.2, 21 minutes for the Nokia 7+, 25 minutes for Samsung A12, and 31 minutes for the Galaxy tab A 2016. This means that each of these devices will only be able to send data over the network after the specified time has elapsed. This strategy makes the TrustChain system resistant to whitewashing attacks because the creation of new identities for the attacker has an additional overhead and practically is not beneficial.

Table 2 shows the security comparison between the improved TrustChain, TrustChain, uPlexa, and MIB. As can be seen, the improved TrustChain outperforms TrustChain and other Blockchains in terms of security.

Table 2. Security comparison between improved TrustChain & compared blockchains

Attacks	Improved TrustChain	TrustChain	uPlexa	MIB
Client Vulnerability	Secured	Vulnerable	Secured	Secured
Whitewashing	Secured	Vulnerable	N/A	N/A
Double-spending	guarantees that will be discovered	guarantees that will be discovered	Secured	Secured
Sybil attack	Secured	Secured	N/A	N/A
Selfish mining	Undefined	Undefined	N/A	N/A
Eclipse	Secured	Secured	N/A	N/A
Replay attack	Secured	Secured	N/A	N/A
Block Withholding	Secured	Secured	Secured	N/A

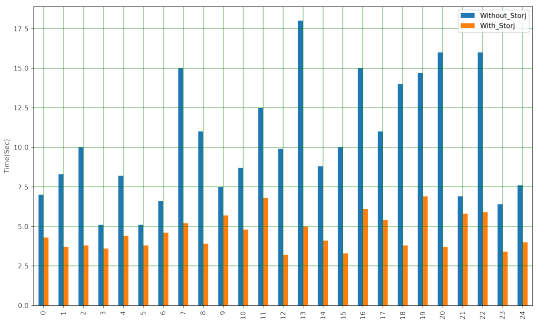

Figure 7. Required time to update the local chain of TrustChain after each fatal error, in two modes

Note that the replay attack, the block withholding attack, and the double-spending attack are patched in the TrustChain and hence, this system is secured against this attack [2].

4.3 Distributed Cloud Evaluation

In Section 3, we mentioned how we connected TrustChain to the Storj distributed cloud and the intuition behind this choice. In this section, we evaluate the impact of Storj on TrustChain in terms of performance. Figure 7 illustrates the required time to update the local chain in the Galaxy Tab A tablet, after each fatal error, in two modes (with and without distributed cloud connection). To simulate a fatal error, we erased the application cache of the device and restarted it. In this case, the node has to update its local chain from other network nodes and Bootstrap servers too. The reason for selecting the Tab A tablet for this test is that it has the lowest configuration among other devices.

As shown in Figure 7, the update time of the Storj is much shorter than the update time of the TrustChain network (without Storj) in all 25 experiments. Note that the loading of information from the Storj depends only on the connection speed, while the initial loading

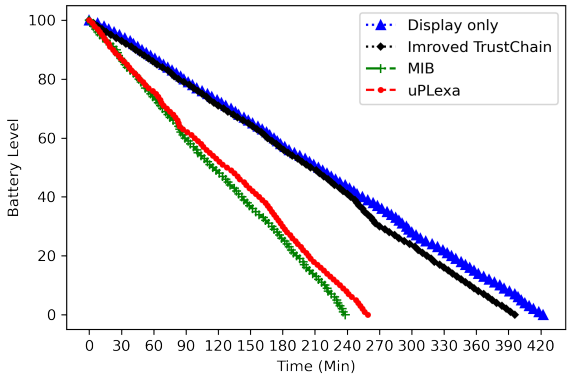

Figure 8. Impact of different blockchains on S9 battery level

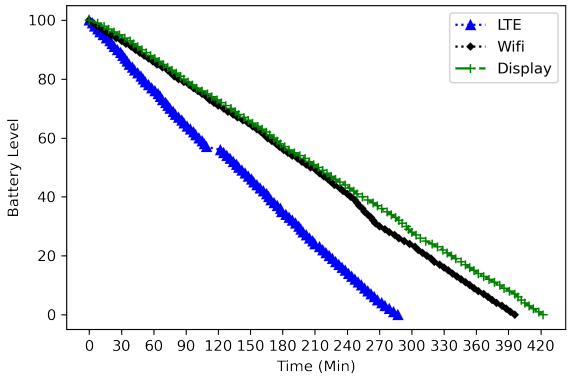

Figure 9. Impact of LTE technology on S9 battery level

of blocks from the TrustChain network has other complications such as connection to the Bootstrap server and re-registration on it, connection speed, and the retrieval of information from several nodes, which increases in the update time.

4.4 Improved TrustChain Vs. PoW-Based Blockchains

This section presents the results acquired from the performance evaluation regarding battery consumption. As Samsung Galaxy S9 is a powerful device for handling Blockchain computations, we run the first set of experiments on this device. We also simulated the occurrence of fatal errors as expressed above. Figure 8 illustrates the battery level of the Samsung S9. As can be seen, the battery usage of the improved TrustChain is closer to the 'display only' mode, compared to MIB and uPlexa. This implies that it incurs less battery consumption, compared to MIB and uPlexa. In other words, the improved TrustChain adds approximately an extra 6% to the discharge rate of display usage in Galaxy S9. This negative impact may reach 38% for MIB and uPlexa.

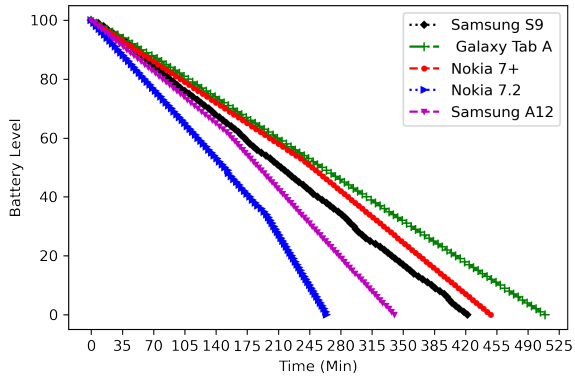

Figure 10. Battery discharge rate for display-only mode in different devices

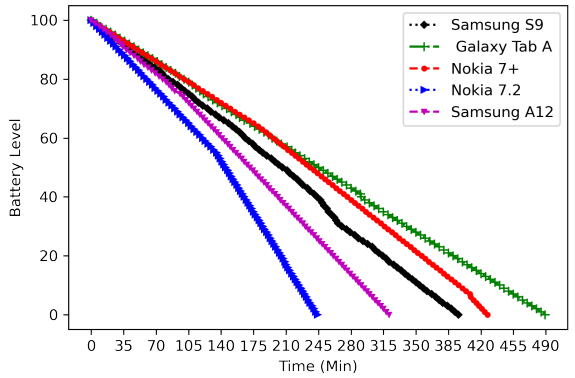

Figure 11. Impact of improved TrustChain on battery level in different devices

The previous experiment has been done using WLAN. In the following, we present the analysis of cellular network usage. For this purpose, we run the improved TrustChain on the Galaxy S9 smartphone. As shown in Figure 9, WLAN connection is approximately 25% more efficient than a cellular network connection.

In the following experiments, we consider all the devices listed in Table 1, Figure 10, and Figure 11 show the battery discharge rate for two modes: i) the display-only mode, and ii) while running the improved TrustChain, respectively. As we mentioned in Section IV.D, the improved TrustChain running on Galaxy S9 incurred a 6% extra discharge rate compared to the display-only mode, while this amount was 38% for MIB and uPlexa. As can be seen in Figure 10, Figure 11, all the inspected devices illustrate similar behavior to Galaxy S9 in terms of discharge rate, and in some cases, the improved TrustChain negative impact on battery consumption is even less than 6%. This implies that the improved TrustChain incurs less battery discharge in all the inspected devices, compared to other smartphone-based Blockchains such as MIB and uPlexa.

5 Conclusion and Future Work

This paper proposes an improved version of TrustChain that is secure against client vulnerability issues and whitewashing attacks. The improved TrustChain has been linked to a distributed Cloud (Storj), which improves the recovery time of devices after fatal errors. We compared the improved TrustChain with PoW-based smartphone-oriented Blockchains such as MIB and uPlexa in terms of battery consumption and illustrate that the improved TrustChain negative impact on the battery level of regular devices is much less than PoW-based Blockchains. We also demonstrated that the improved TrustChain is more secure against attacks such as Whitewashing and client vulnerability attacks.

In the future, we intend to implement an authentication system on smart contracts which is more compatible with the nature of distributed systems. We also plan to consider other Blockchains like File and Chia as distributed cloud systems and evaluate the performance of the proposed system in these cases.

References

[1] IMRAN. BASHIR. *MASTERING BLOCKCHAIN: Distributed Ledger Technology, Decentralization, and Smart Contracts Explained, ; distributed Ledger*. Packt Publishing, 2018.

[2] Pim Otte, Martijn de Vos, and Johan Pouwelse. Trustchain: A sybil-resistant scalable blockchain. *Future Generation Computer Systems*, 107:770–780, 2020.

[3] Qin Wang, Jiangshan Yu, Shiping Chen, and Yang Xiang. Sok: Diving into dag-based blockchain systems. *arXiv preprint arXiv:2012.06128*, 2020.

[4] Saurabh Singh, ASM Sanwar Hosen, and Byungun Yoon. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *IEEE Access*, 9:13938–13959, 2021.

[5] Uplexa white paper. https://uplexa.com/media/uPlexa-Whitepaper-EN.pdf. Accessed: 20221-8-17.

[6] Bitcoin white paper. https://www.mibcoin.io/. Accessed: 20221-8-17.

[7] Aleksandr Ometov, Yulia Bardinova, Alexandra Afanasyeva, Pavel Masek, Konstantin Zhidanov, Sergey Vanurin, Mikhail Sayfullin, Viktoriia Shubina, Mikhail Komarov, and Sergey Bezzateev. An overview on blockchain for smartphones: State-of-the-art, consensus, implementa-

tion, challenges and future trends. *IEEE Access*, 8:103994–104015, 2020.

[8] Keke Gai, Jinnan Guo, Liehuang Zhu, and Shui Yu. Blockchain meets cloud computing: a survey. *IEEE Communications Surveys & Tutorials*, 22(3):2009–2030, 2020.

[9] Trsutchain superapp. https://github.com/Tribler/trustchain-superapp. Accessed: 20221-8-17.

[10] R3-corda: Access the documentation for corda - our open source blockchain platform. https://docs.corda.net/. Accessed: 20221-8-17.

[11] Coti: a decentralized, high performance cryptocurrency ecosystem optimized for creating digital payment networks and stable coins. https://coti.io/files/COTI-technical-whitepaper.pdf. Accessed: 20221-8-17.

[12] Avalanche platform. https://assets.website-files.com/5d80307810123f5ffbb34d6e/6008d7bbf8b10d1eb01e7e16_Avalanche%20Platform%20Whitepaper.pdf. Accessed: 20221-8-17.

[13] Jiangshan Yu, David Kozhaya, Jeremie Decouchant, and Paulo Esteves-Verissimo. Repucoin: Your reputation is your power. *IEEE Transactions on Computers*, 68(8):1225–1237, 2019.

[14] Lijun Sun, Qian Yang, Xiao Chen, and Zhenxiang Chen. Rc-chain: Reputation-based crowdsourcing blockchain for vehicular networks. *Journal of Network and Computer Applications*, 176:102956, 2021.

[15] Minghui Xu, Shuo Liu, Dongxiao Yu, Xiuzhen Cheng, Shaoyong Guo, and Jiguo Yu. Cloudchain: a cloud blockchain using shared memory consensus and rdma. *IEEE Transactions on Computers*, 2022.

[16] Kapil Aggarwal and Santosh Kumar Yadav. Decentralized cloud and file system for blockchain environment. In *Soft Computing for Security Applications*, pages 331–339. Springer, 2022.

[17] Mallikarjun Reddy Dorsala, VN Sastry, and Sudhakar Chapram. Blockchain-based solutions for cloud computing: A survey. *Journal of Network and Computer Applications*, 196:103246, 2021.

[18] Mongodb. https://docs.mongodb.com/manual/sharding/. Accessed: 20221-8-17.

[19] Mongo data encryption. https://www.mongodb.com/basics/mongodb-encryption. Accessed: 20221-8-17.

[20] Dfinity, the internet computer for geeks. https://dfinity.org/whitepaper.pdf. Accessed: 20221-8-17.

[21] Storj whitepaper. https://www.storj.io/whitepaper. Accessed: 20221-8-17.

[22] extract battery information for android devices. https://github.com/seyedsalar/TrustChain-battery-parameters-/tree/main. Accessed: 20221-8-17.

[23] Jingpei Wang, Mufeng Wang, Zhenyong Zhang, and Hengye Zhu. Towards a trust evaluation framework against malicious behaviors of industrial iot. *IEEE Internet of Things Journal*, 2022.

Seyed Salar Ghazi received his B.Sc. degree in Computer Engineering from Urmia University, Iran, and his M.Sc. degree in Engineering from Ferdowsi University of Mashhad, Iran. He is also a researcher at the Ferdowsi University of Mashhad in the fields of distributed systems and blockchain technology. His main research interests include blockchain, machine learning, and network security.

Haleh Amintoosi is an associate professor at the Ferdowsi University of Mashhad, Mashhad, Iran. She is also a visiting senior lecturer at the School of Computer Science and Engineering, University of New South Wales (UNSW), Sydney, NSW, Australia. Her main research interests include trust and privacy in crowdsourcing and crowdsensing systems, authentication protocols, and blockchain. Haleh received her Ph.D. degree in computer science from UNSW.

Sahar Pilevar received her B.Sc. degree in computer engineering from Azad University of Mashhad, Iran. She is currently pursuing an M.Sc. degree in the field of software engineering at Ferdowsi University of Mashhad (FUM). Her research interests include blockchain technology, smart contract, edge computing, and mechanism design.