Received: 8 September 2019 | Revised: 15 October 2019 | Accepted: 21 October 2019 | Published on: 23 December 2019

DOI: 10.1002/spy2.98

WILEY

RESEARCH ARTICLE

Outsourcing IOTA proof-of-work to volunteer public devices

Umair Sarfraz[1] | **Sherali Zeadally**[2] | **Masoom Alam**[1]

[1] Department of Computer Science, Cyber Security lab, COMSATS University, Islamabad, Pakistan

[2] University of Kentucky, Lexington, Kentucky

Correspondence
Umair Sarfraz, Department of Computer Science, Cyber Security lab, COMSATS University, Islamabad, Pakistan.
Email: aquadestructor@gmail.com

Abstract

Distributed ledger technologies will play an important role in the communication and data integrity of internet of things (IoT) devices in the future. IOTA is a public distributed ledger that can serve as a scalable, secure and feeless settlement layer for IoT transactions. It enables micro-transactions for smart devices and offers a highly scalable architecture for network consensus and overcomes the inefficiencies of existing distributed ledgers. Like most mining-based ledgers, IOTA also require solving a computational puzzle for every request to the ledger to avoid unwanted (ie, spam) content added to the ledger. We propose a system architecture that consumes the available computational resources of public volunteer devices for solving the expensive computational puzzles. We show that our proposed architecture can integrate with the global data and network layer of the IOTA ledger and acts as a middleware for providing computational resources from public volunteer devices to the IOTA network.

KEYWORDS

directed acyclic graph (DAG), distributed ledgers, internet of things (IoT), IOTA, proof of work (PoW), security, volunteer computing.

1 | INTRODUCTION

Recent advances in network technologies, big data and smart devices have evolved the concept of Internet of Things (IoT) to describe a more intricate concept called internet of everything (IoE). The IoE includes people and processes as system stakeholders to the existing system of Internet of Things.[1] IoE connects end users, data, and smart devices in an intelligent manner. In addition to machine-to-machine (M2M) interactions, IoE includes machine-to-people (M2P), and people-to-people (P2P) interactions with the ubiquitous aggregation of networked devices. IoE will open up more relevant and valuable ways for P2P interactions. For example, interactions between doctors and patients can be improved by placing sensors on the skin, providing doctors continuous information about a patient's vital signs over a secure Internet connection. Thus, IoE can allow people to become data-centric nodes on the Internet.

Internet is evolving toward decentralization to solve the problems of scalability, data integrity, and data privacy. IoE will play a vital role in the global decentralization of Internet. Distributed ledger technologies will be an important component in the concept of IoE. However, to date, the majority of distributed ledgers still face operational challenges because they suffer scalability and throughput issues. Recently, we have witnessed an increasing amount of research aimed at addressing some of these problems and several distributed consensus protocols[2-4] have been proposed for enabling high throughput communication in a distributed network. The most promising one out of all in the context of IoE is the IOTA ledger[2], which enables micro-transactions for internet of things (IoT) devices and offers a highly scalable approach for network growth and the confirmations of transactions on the network.

Every public distributed network requires spam (unwanted requests and content) and denial of service (DoS)[5] prevention measures to ensure high availability and security of the network. Mining-based ledgers such as Bitcoin[6] employ similar techniques[7] for mitigating these attacks. These techniques require the *requester*[1] to solve a computational puzzle and only process the request with a valid proof against the computational puzzle. The IOTA protocol uses a similar mechanism for preventing unwanted content and requests (ie, spam). Many of these prevention techniques are computationally expensive which cause them to slow down transaction confirmation rates.

2 | MAIN CONTRIBUTIONS OF THIS WORK

We summarize the main contributions of this work as follows:

- In this work, we propose and discuss a system model that consumes the idle resources of public IoT devices for performing the computationally expensive operations for proof generation.
- We describe how our system works without altering any of the existing operations of the IOTA protocol. We show that our system acts as a public computational resources provider for the IOTA network.

3 | BACKGROUND

3.1 | Distributed ledger technologies

A distributed ledger is a distributed database spread across independent nodes (devices) on a peer-to-peer network, where each node in the network is responsible for independently storing, sharing and maintaining an identical copy of the electronic ledger. The distribution of new data added to the ledger is not communicated to nodes by a central authority, but is instead constructed independently by every node in the network. Consensus protocols are responsible for maintaining the integrity of transactions (data) stored on the ledger. This trust-less architecture creates a new system of records that removes the need for a trusted agent controlling the transactions happening in the network.

A universal distributed ledger cannot feasibly serve all industries considering the diverse needs of businesses and end users. This had led to the development of numerous distributed ledgers, each with marginally different sets of protocols while being based on same fundamental concepts of a distributed system. From a broader perspective, different distributed ledgers (as of today) can be categorized into two types namely *public* (also known as *permissionless*) and *private* (also known as *permissioned*).

3.1.1 | Public distributed ledgers

Distributed ledgers based on a *permissionless* model are closer to the original idea of blockchain. Distributed ledgers that are in this category include most of the known digital currencies. In theory, anyone can become part of public distributed ledgers because of their permissionless architecture. In addition to their open architecture, there are certain inherent characteristics associated with them. These characteristics include *transparency*, *decentralization* and *anonymity*.

Despite the huge potential inherent in the public distributed ledgers, several challenges limit their usability. Some of these challenges include:

- *Network size:* Public distributed ledger systems require a large number of nodes (devices) in the network to reap their full benefit. A small network has low resilience against bad actors in the network who can affect the functionality of the overall system.
- *Storage constraints:* Data storage imposes a significant cost on a public distributed ledger because every node participating in the network has to store increasing amounts of data. Consequently, storage remains a major challenge for any realistic application that utilizes a public distributed ledger.
- *Scalability:* The transaction processing capability of most public distributed ledgers is limited. As the number of transactions increases, the performance of new transactions is affected, which in turn affects the overall network scalability.[8]

- *Computational overheads:* The incentive mechanisms employed by many of the public distributed ledgers make inefficient use of computational resources hence constraining the network throughput. The Bitcoin[6] protocol consumes on average 592 kW per hour energy for a single transaction which is very high compared with the VISA payment network that has an average energy consumption of 169 kW per hour for 100 000 transactions.[9]

The challenges above are a major hurdle in the widespread deployment of public distributed ledgers by industries and businesses. Over the last few years, we have seen several research and development efforts in overcoming these challenges. For instance, protocols such as IOTA, Hashgraph, and Algorand[2-4] offer acceptable solutions to some of the challenges but more research is needed in the future to develop innovative, secure, scalable, and robust solutions.

3.1.2 | Private distributed ledgers

Distributed ledgers were originally intended to be open, free and public in contrast to private distributed ledgers. In principle, private distributed ledgers require permission for joining the network, viewing transaction information, or creating new transactions. The information stored on permissioned ledgers is validated only by trusted validators. One needs authorization to be able to read or write information from/to the ledger. One of the major barriers for enterprises wishing to adopt a public distributed ledger is that they require high computational power for achieving consensus. The completely transparent nature of the public distributed ledger also becomes a point of concern (eg, privacy) for businesses. Although the client access is the key difference between the public and private distributed ledgers, both share many similar properties and are equally efficient in their own ways.

3.2 | Public computing

As hardware technologies become more powerful, affordable and compact, the overall performance of devices, systems and networks keep improving. Traditionally, computer software has been written for *serial* computation.[10] One of the ways to speed up computations is to "parallelize" them, that is, break a piece of a program into a series of instructions that can be simultaneously executed. Historically, the concept of parallelization was only used for scientific computing but eventually the concept became the basis for the design of *high performance computing*.[11]

Most of the world's processing power is no longer in supercomputers. Rather, it is distributed in the billions of consumer devices all across the world. This shift of computational resources has become the basis of a new paradigm often referred to as "public computing" in which end users voluntarily make their system's resources available to scientific projects that have high computational requirements. Computational tasks also require storage capabilities and even in this new paradigm, public resources can provide high aggregate storage capacities. If devices share a fraction of their unused storage capacities, the total storage capacity would be much higher than the capacity of normal centralized storage systems.

Public computing is being used in fields including physical sciences, mathematics, molecular biology, medicine, astrophysics and cryptography. Projects such as SETI@home[12] uses computational resources from millions of computers in homes and offices across the globe for analyzing radio signals from space. While public computing is mostly used in the context of pure scientific computations, it can also be used in the context of global decentralization of the Internet infrastructure.

3.3 | IOTA—a public distributed ledger for IoE

The IOTA protocol[2] was developed to address current distributed ledgers. The protocol enables feeless micro-transactions and serves as a backbone for secure communication among IoT devices. The IOTA protocol was conceived to build a machine economy by enabling true interoperability between smart devices. Through its transaction settlement layer and scalable architecture, the IOTA protocol aims to become a carrier for transactions of IoT-enabled devices.

The IOTA protocol overcomes the shortcomings of existing distributed ledgers by introducing a new consensus structure called the *Tangle*.[2] This consensus structure is based on a directed acyclic graph (DAG) and it removes the need of blocks and miners from the network. All transactions that want to be added to the IOTA distributed ledger

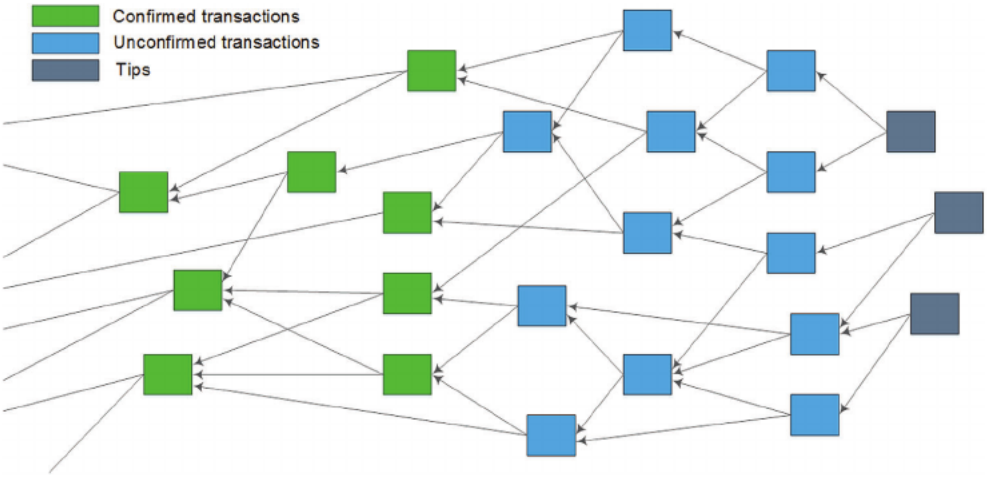

FIGURE 1 The tangle is a directed acyclic graph (DAG). Every transaction validates two tips. Green squares represent confirmed transactions. Blue squares represent unconfirmed transactions while gray squares represent tips

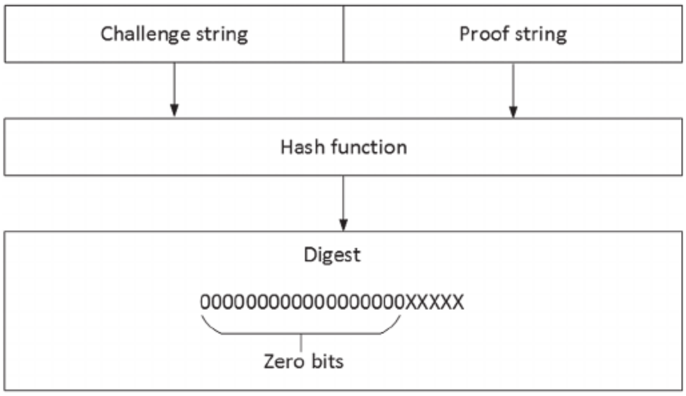

FIGURE 2 Proof-of-work computational puzzle

must validate two previous unconfirmed transactions on the ledger as shown in Figure 1. The validation process is done by doing a small amount of computational work similar to *Hashcash*.[7] This mechanism enables a scalable architecture and completely removes the need for specialized roles and responsibilities for actors in the network. Moreover, the IOTA protocol was designed with resource constraints of smart devices and quantum resiliency in mind that is why the IOTA protocol uses the hash-based signature[13,14] scheme instead of Elliptic Curve Cryptography (ECC).[15,16] The Winternitz one-time signature Scheme[13,14] used in the IOTA protocol not only makes IOTA a quantum-proof protocol but also makes it well-suited for adaptability in the IoT space because hash-based signing is faster than ECC.

4 | LINKING PUBLIC COMPUTING WITH IOTA POW

4.1 | Proof-of-work

A proof-of-work system is an effective measure of proving engagement in a significant amount of computational effort. It was initially proposed for preventing spam and DoS attacks on a network.[17] A proof-of-work protocol should require a lot of trial and error attempts on a random process so that the probability of generating a valid proof *on average* is low. Verifying the validity of the proof requires far less time and computational effort. One popular system of proof-of-work is *Hashcash*[7] that uses partial hash inversions for generating a valid proof. Figure 2 shows that a random proof string p is required, which when hashed after concatenation with a challenge string c results in a sequence of n zeros. Verification of the proof string p would also take far less time because finding $H(p+c)$ is not computationally expensive. The difficulty of finding the proof can be adjusted by tuning the required number of zeros. Similar to Hashcash,[7] the IOTA protocol also requires proof-of-work. Unlike the expensive proof-of-work employed in miner-based ledgers such as Bitcoin,[6] the IOTA proof-of-work is a short computational process.

4.2 | Transaction reattachments

When a new transaction is added to the IOTA ledger, there is a possibility that it would never get picked by newer transactions for validation. If a transaction is never validated, it would never be confirmed by the network. To overcome this issue, the IOTA protocol introduces the concept of transactions reattachments. Reattaching a transaction creates a copy of the original transaction and adds it to a part of the tangle graph where the probability of the transaction being validated and approved by newer transactions is high. This creates a double spend situation and the tangle ensures that only a single copy of the transaction is confirmed by the network. As transactions reattachment requires another entry to the ledger, proof-of-work needs to be performed again.

4.3 | Transaction promotions

Promoting a transaction on the IOTA ledger yields faster transaction confirmations. Promotion requires a spam[2] transaction that approves the original transaction. A series of promotions increases the probability of a transaction getting confirmed quickly. Unlike transactions reattachments, transactions promotions do not create a double spend situation because every promotion creates a fresh spam transaction in the IOTA ledger.

The IOTA official client wallet[18] Trinity automatically performs transactions reattachments and promotions whenever necessary and therefore automatically yields faster confirmation rates. However, auto promotion and reattachment in the Trinity wallet are only restricted to the transactions created or received in the wallet. This creates an unusual situation for the transactions created from other clients because their confirmation rates will be low in the absence of no reattachments and promotions. A similar situation will occur even for the transactions created from the Trinity wallet if the wallet goes inactive.

5 | PROPOSED SYSTEM AND ARCHITECTURE

This problem of low confirmation rates on the IOTA network because of less promotions and reattachments can be solved if the responsibility of reattaching and promoting transactions can be shifted from the client layer to the network layer (the Tangle). The network layer provides a global view of all the transactions on the IOTA ledger. We propose an architecture (as shown in Figure 3) where the transaction rates on the IOTA network can be improved by introducing a system that uses idle resources of devices for automatically promoting and reattaching the transactions happening on the network. In this system, devices can volunteer their idle resources by requesting a promotion/reattachment service for tasks. The system services, based on the free computational resources of the requested client, returns transactions for reattachments and promotions with a deadline. This service model can be further improved by incentivizing clients for their completed tasks. The service acts as a middleware between the Tangle and public devices making their idle resources available for use.

Our proposed system, inspired from the concept of public computing[12] and grid computing,[19] makes use of idle computing resources. In contrast to grid computing, our proposed system is based on a *pull* model in which volunteer devices continuously request tasks from a central service, rather than being based on a *push* model as in grid computing. The *pull* model implies that the volunteer devices do not necessarily need to accept incoming network requests and they can operate behind Network Address Translators or firewalls. Another important difference is the relationship between entities. Grid computing has a symmetric relationship between entities, that is, any entity can either *provide* or *use* idle computing resources in contrast to our proposed model in which volunteer devices only *provide* idle computing resources.

Next, we discuss the main components and services of our proposed system.

Core client: The core client is a process, which needs to be executed on the device that wants to automatically outsource its computational capabilities. The core client is responsible for maximizing the resource usage of the device. It is also responsible for communicating with the server services for requesting tasks and for reporting results within deadlines.

Tangle subscriber: The Tangle subscriber service communicates with the tangle network by using a publish-subscribe model. The Tangle publishes every new transaction that it receives and adds to the ledger and the subscriber service gets notified about the new transactions continuously. Upon receiving new transactions, the subscriber service adds all new *valid*[3] transactions to the database.

Database: The database component is responsible for storing every new transaction published by the Tangle. The *tangle subscriber* component acts as a communication layer between the database component and the tangle. The database

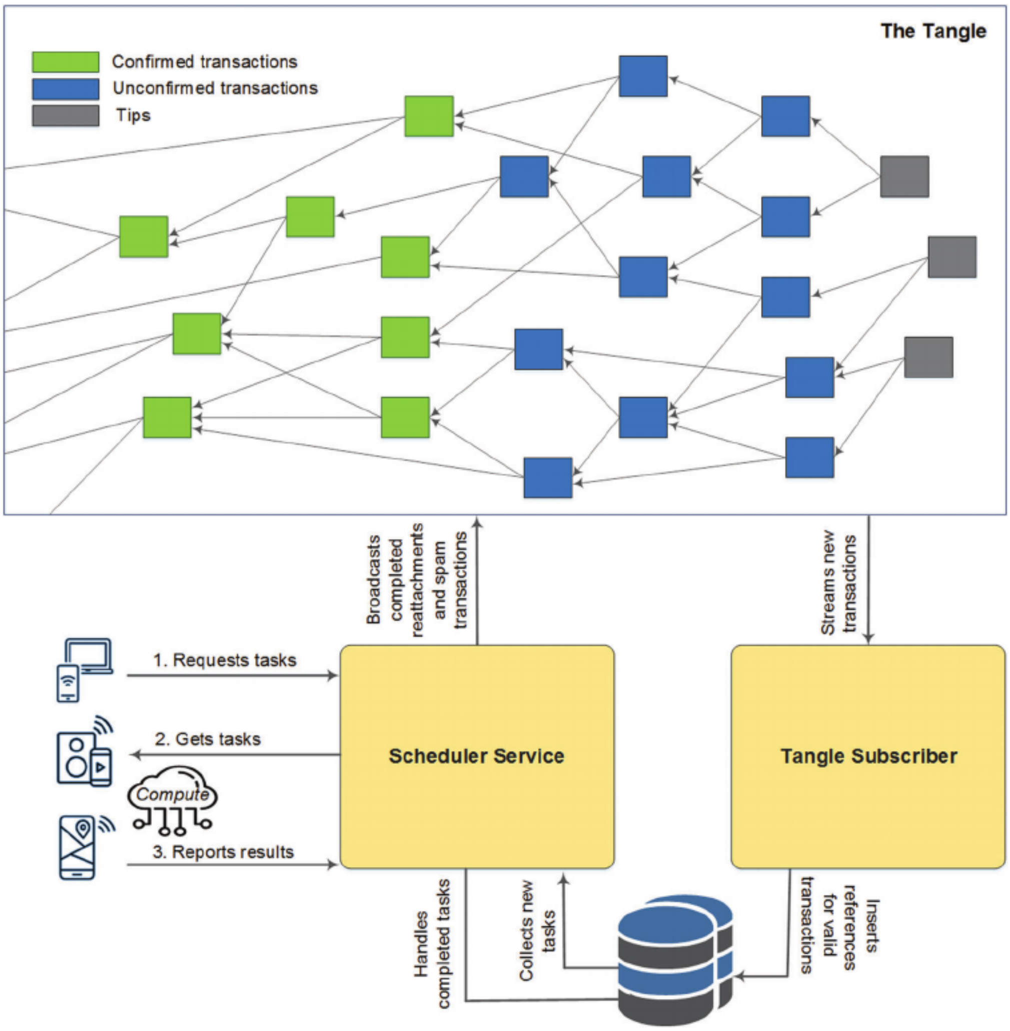

FIGURE 3 Overview of our proposed system architecture

also stores the additional information about the clients. In addition, it also stores the results and the amount of computational contribution by each client. An incentivized service (paying each client based on the contributions) can be employed in parallel with the other active services of the system by using the amount of work performed by each client.

Scheduler service: The scheduler service is responsible for handling the initiated requests and results from all the clients. It issues tasks and also handles reports for completed tasks from clients. Furthermore, the Scheduler service is also responsible for broadcasting the necessary completed results to the IOTA network.

6 | SYSTEM IMPLEMENTATION AND EVALUATION

We have developed a proof-of-concept implementation of our proposed system and we have evaluated it with the IOTA ledger's *devnet also known as testnet*.[4] In particular, we have implemented all the components and services of our proposed system (Section 5), which are sufficient to demonstrate the usability of our proposed solution as an application in a real world setting. In addition, we have separately evaluated the IOTA proof-of-work efficiency on different hardware architectures as shown in Table 1.

Our implementation uses *Node.js* version 8.15.0. For performing the proof-of-work, we have leveraged *entangled*, the official C++ implementation of IOTA ledger components and models. We have used *MongoDB* for the *database* component in our proposed solution. The *database* component allows read and write operations to the *Tangle subscriber* service for writing newly published transactions and to the *scheduler* service for assigning transactions to client requests. To minimize storage requirements for the *database* component, we have only stored a reference[5] to the original transaction. The *tangle subscriber* service (after establishing a connection with the *database* component) subscribes to the

TABLE 1 Time trials for proof-of-work with varying *difficulty* and *bundle size* (number of transactions) using different *processors* and *RAM* (available / total). Each proof-of-work was performed (*bundle size* * *trials*) times

Processor	RAM (GB)	Bundle size	Trials	Difficulty	Min. time (s)	Max. time (s)	Avg. time (s)
Intel i5-7360U @ 2.30	6.35 / 8	5	20	14	7.30	9.64	7.99
Intel i5-7360U @ 2.30	6.38 / 8	5	20	15	17.28	21.54	18.67
Intel i5-7360U @ 2.30	6.28 / 8	5	20	16	17.99	23.59	19.21
Intel i5-7360U @ 2.30	6.59 / 8	15	20	14	23.11	28.21	25.14
Intel i5-7360U @ 2.30	6.52 / 8	15	20	15	52.01	64.44	56.23
Intel i5-7360U @ 2.30	6.59 / 8	15	20	16	52.56	65.67	59.89
Intel M-5Y10c @ 0.80	1.64 / 3.91	5	20	14	14.99	15.87	15.51
Intel M-5Y10c @ 0.80	1.36 / 3.91	5	20	15	34.78	37.09	36.49
Intel M-5Y10c @ 0.80	1.57 / 3.91	5	20	16	35.07	37.58	37.03
Intel M-5Y10c @ 0.80	2.18 / 3.91	15	20	14	44.93	47.10	46.76
Intel M-5Y10c @ 0.80	2.51 / 3.91	15	20	15	93.48	111.74	93.48
Intel M-5Y10c @ 0.80	1.48 / 3.91	15	20	16	109.74	113.00	111.66
Intel i7-4600U @ 2.10	3.96 / 7.21	5	20	14	10.95	14.16	13.32
Intel i7-4600U @ 2.10	5.31 / 7.21	5	20	15	96.33	99.61	98.00
Intel i7-4600U @ 2.10	3.97 / 7.21	5	20	16	28.03	34.61	33.30
Intel i7-4600U @ 2.10	3.96 / 7.21	15	20	14	36.70	46.84	44.87
Intel i7-4600U @ 2.10	5.36 / 7.21	15	20	15	76.68	105.36	98.83
Intel i7-4600U @ 2.10	5.34 / 7.21	15	20	16	84.61	104.22	101.55
Intel i5-5257U @ 2.70	6.06 / 8	5	20	14	8.84	9.78	9.05
Intel i5-5257U @ 2.70	6.04 / 8	5	20	15	20.52	21.73	20.52
Intel i5-5257U @ 2.70	6.20 / 8	5	20	16	20.52	21.61	20.96
Intel i5-5257U @ 2.70	6.41 / 8	15	20	14	26.81	35.13	29.55
Intel i5-5257U @ 2.70	6.53 / 8	15	20	15	61.81	74.76	65.36
Intel i5-5257U @ 2.70	6.30 / 8	15	20	16	67.41	72.77	68.91
Samsung mt6757 @ 2.4	1.91 / 3.74	5	20	14	65.85	68.88	67.13
Samsung mt6757 @ 2.4	1.95 / 3.74	5	20	15	116.31	230.80	156.50
Samsung mt6757 @ 2.4	1.98 / 3.74	5	20	16	95.42	154.01	147.89
Samsung mt6757 @ 2.4	2.02 / 3.74	15	20	14	135.28	201.26	194.89
Samsung mt6757 @ 2.4	2.15 / 3.74	15	20	15	438.38	458.75	453.28
Samsung mt6757 @ 2.4	2.14 / 3.74	15	20	16	333.95	462.27	450.32
Huawei hi6250 @ 2.36	1.31 / 3.78	5	20	14	22.14	37.97	27.41
Huawei hi6250 @ 2.36	1.32 / 3.78	5	20	15	57.96	64.52	60.83
Huawei hi6250 @ 2.36	1.71 / 3.78	5	20	16	57.71	68.06	61.41
Huawei hi6250 @ 2.36	1.31 / 3.78	15	20	14	74.49	85.09	80.66
Huawei hi6250 @ 2.36	1.85 / 3.78	15	20	15	163.01	244.48	192.91
Huawei hi6250 @ 2.36	1.81 / 3.78	15	20	16	157.90	233.70	177.59

live feed of new transactions published on the IOTA ledger. In particular, the *tangle subscriber* service subscribes to the *tx*[6] topic from the live feed. After caching an in-memory batch of transaction references, the *tangle subscriber* service makes batched writes in the *database* component. The *scheduler* service is responsible for handling requests from volunteer clients.[7] It assigns tasks to the requesting clients and also interacts with the *database* component for keeping track of the amount of work performed by the clients. In general, the *scheduler* service acts as a middleware between the clients and the *database* component. The *core client* component is essentially a process that needs to run on the volunteer node. We evaluated our *core client* component on a Mac system with 8 GB 2133 MHz LPDDR3 RAM and 2.3 GHz Intel Core i5 processor. The *core client* initiates the contact with the *scheduler* service by providing it the hashed universally unique identifier (UUID) of the volunteer node. After a successful handshake, the *core client* periodically requests the *scheduler* service for tasks and uses the available computation capabilities of the volunteer node for executing the tasks.

Our proof-of-concept implementation efficiently integrates with the IOTA ledger and does not incur any memory overheads on the volunteer nodes. The implementation efficiently uses the available computation capabilities of the volunteer nodes thereby improving the overall throughput and yielding faster confirmation rates of the transactions.

7 | FUTURE CHALLENGES

7.1 | Single point of failure

Our proposed system model requires a centralized storage for maintaining a partial copy of the Tangle and also for managing information about volunteering clients. As with any centralized system, our proposed system also faces the challenge of single point of failure. This can be overcome if the volunteer devices directly use the IOTA ledger for getting tasks for computation. However, all the client devices would require to stay in sync with each other so that their computational efforts are not wasted by performing a task that is already performed by another client. The design challenges of constructing such a peer-to-peer system among volunteer devices still remain an open issue.

7.2 | Incentivizing computational contributions

A credit and accountability system, similar to Berkeley open infrastructure for network computing (BOINC)[20] can be employed for keeping track of the computational contributions of the volunteer devices. Such a system can attract more volunteers and can also motivate existing volunteers in the system for making more contributions to the network. However, a secure design of a monetary *reward and fine* system wherein contributions are paid-off for completed tasks and fined for cheating is another challenge that will need to be addressed in the future.

8 | CONCLUSION

Distributed ledger technology is about to change digital services and operations by offering an immutable data integrity and communication layer. It offers a trust-less collaboration service for smart devices and also handles interactions among people and devices thereby contributing toward the goals of the IoE concept. While many existing distributed ledgers face throughput and propagation latency issues, some new ledger designs (such as IOTA ledger) aim to solve these issues by being developed based on a highly scalable consensus design. To enable security and stability, most public distributed ledgers employ a technique that requires the requester to solve a computationally expensive puzzle. Not only the expensive puzzle often misuses resources, it also impedes the overall speed of the network and transaction confirmation rates. In this article, we have proposed and discussed a system architecture that consumes the idle resources of public volunteer devices for executing computationally expensive tasks for the IOTA ledger. Our proposed system works in conjunction with the network layer of the IOTA ledger and can help public volunteer devices to contribute computational resources to the IOTA network.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments, which helped us improve the content and presentation of this paper.

ENDNOTES

[1] Entities attempting to write to the ledger.

[2] A spam transaction is a zero-value transaction. As the transaction gets added to the ledger, proof-of-work is required.

[3] Transactions that are semantically invalid or are non-funded would never confirm on the IOTA network, and therefore they should be ignored.

[4] A dedicated network mimicking the IOTA protocol by allowing transactions with fake tokens.

[5] *Bundle hash* is used as a reference to the original transaction. By using this reference, clients can query the IOTA ledger to get the complete transaction details.

[6] IOTA ledger reference implementation streams allow streaming different feeds including the *tx* feed. The goal is to allow clients to subscribe to targeted stream events.

[7] A client running the *Core client* component.

ORCID

Umair Sarfraz https://orcid.org/0000-0002-1468-5018
Sherali Zeadally https://orcid.org/0000-0002-5982-8190

REFERENCES

1. D. Evans, The internet of everything: how more relevant and valuable connections will change the world. Cisco. 2012. [Online], Available: https://www.cisco.com/c/dam/global/enmy/assets/ciscoinnovate/pdfs/IoE.pdf. [Accessed November 1, 2018].
2. S. Popov, IOTA: the tangle. 2016. [Online]. Available: https://iota.org/IOTAWhitepaper.pdf. [Accessed November 20, 2018].
3. L. Baird, The swirld hashgraph consensus algorithm: fair, fast, byzantine fault tolerance. 2016. [Online]. Available: http://www.swirlds.com/downloads/SWIRLDS-TR-2016-01.pdf. [Accessed November 10, 2018].
4. Y. Gilad, R. Hemo, S. Micali, G. Vlachos and N. Zeldovich. Algorand: scaling byzantine agreements for cryptocurrencies. Paper presented at: 26th ACM Symposium Operating Systems Principles; 2017; pp. 51–68.
5. R. M. Needham. Denial of service. Paper presented at: 1st ACM Conference on Computer and Communications Security; 1993; pp. 151–153.
6. S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system. 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf. [Accessed November 15, 2018].
7. A. Back, Hashcash - A denial of service counter measure. 2002. [Online]. Available: http://www.hashcash.org/hashcash.pdf. [Accessed November 20, 2018].
8. Ismail L, Materwala H. Article a review of blockchain architecture and consensus protocols: use cases, challenges, and solutions. *Symmetry*. 2019;11(10):1198.
9. Digiconomist. Bitcoin energy consumption index. *Digiconomist*. 2018. [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption. [Accessed: Nov. 20, 2018].
10. B. Barney, Introduction to Parallel Computing, Lawrence Livermore National Laboratory. 2007. [E-book] Available: https://computing.llnl.gov/ tutorials/parallel comp/. [Accessed: Nov. 10, 2018].
11. Rauber T, Rünger G. *Parallel Programming for Multicore and Cluster Systems*. 2nd ed. Heidelberg: Springer Science and Business Media; 2013:1.
12. Anderson DP, Cobb J, Korpela E, Lebofsky M, Werthimer D. SETI@home: an experiment in public-resource computing. *Commun ACM*. 2002;45(11):56-61.
13. Bernstein DJ, Dahmen E, Buchmann J. Hash-based digital signature schemes. *Post-Quantum Cryptography*. Heidelberg: Springer; 2008:35-36.
14. S. Rohde, T. Eisenbarth, E. Dahmen, J. Buchmann and C. Paar, Fast hash-based signatures on constrained devices. Paper presented at: 8th Smart Card Research and Advanced Application, CARDIS; 2008; pp. 104–117.
15. Kapoor V, Sonny V, Singh R. Elliptic curve cryptography. *ACM Ubiquity*. 2008;8(20):1–8.
16. Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA). *Int J Inf Security*. 2001;1(1):36-63.
17. Dwork C, Naor M. Pricing via processing or combatting junk mail. In: Ernest FB, ed. *Advances in Cryptology—CRYPTO 92*. London, UK: Springer-Verlag; 1992:139–147.
18. IOTA Foundation, Trinity Wallet. IOTA Foundation. 2018. [Online]. Available: https://trinity.iota.org/. [Accessed Nov. 18, 2018].
19. Berman F, Fox G, Hey A. *Grid Computing: Making the Global Infrastructure a Reality*. New York: John Wiley & Sons, Inc.; 2003.
20. D. P. Anderson, A. Menezes and S. Vanstone, BOINC: a system for public-resource computing and storage. Paper presented at: Proceedings of Fifth IEEE/ACM International Workshop on Grid Computing, GRID'04; November 2004; pp. 4–10.

How to cite this article: Sarfraz U, Zeadally S, Alam M. Outsourcing IOTA proof-of-work to volunteer public devices. *Security and Privacy*. 2020;3:e98. https://doi.org/10.1002/spy2.98