Towards an Optimized BlockChain for IoT

Conference Paper · April 2017

DOI: 10.1145/3054977.3055003

3 authors:

Ali Dorri
Queensland University of Technology
82 PUBLICATIONS **6,922** CITATIONS
SEE PROFILE

Salil S. Kanhere
UNSW Sydney
460 PUBLICATIONS **19,401** CITATIONS
SEE PROFILE

Raja Jurdak
Queensland University of Technology
431 PUBLICATIONS **13,771** CITATIONS
SEE PROFILE

Towards an Optimized BlockChain for IoT

Ali Dorri
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
ali.dorri@unsw.edu.au

Salil S. Kanhere
School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
salil.kanhere@unsw.edu.au

Raja Jurdak
CSIRO DATA61, Brisbane, Australia
Raja.Jurdak@csiro.au

ABSTRACT

There has been increasing interest in adopting BlockChain (BC), that underpins the crypto-currency Bitcoin, in Internet of Things (IoT) for security and privacy. However, BCs are computationally expensive and involve high bandwidth overhead and delays, which are not suitable for most IoT devices. This paper proposes a lightweight BC-based architecture for IoT that virtually eliminates the overheads of classic BC, while maintaining most of its security and privacy benefits. IoT devices benefit from a private immutable ledger, that acts similar to BC but is managed centrally, to optimize energy consumption. High resource devices create an overlay network to implement a publicly accessible distributed BC that ensures end-to-end security and privacy. The proposed architecture uses distributed trust to reduce the block validation processing time. We explore our approach in a smart home setting as a representative case study for broader IoT applications. Qualitative evaluation of the architecture under common threat models highlights its effectiveness in providing security and privacy for IoT applications. Simulations demonstrate that our method decreases packet and processing overhead significantly compared to the BC implementation used in Bitcoin.

CCS CONCEPTS

•**Computer systems organization** →**Embedded systems;** *Redundancy;* Robotics; •**Networks** →Network reliability;

KEYWORDS

Internet of Things, Security, Privacy, BlockChain

ACM Reference format:
Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2017. Towards an Optimized BlockChain for IoT. In *Proceedings of The 2nd ACM/IEEE International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA USA, April 2017 (IoTDI 2017),* 6 pages.
DOI: http://dx.doi.org/10.1145/3054977.3055003

1 INTRODUCTION

Internet of Things (IoT) has a broad range of applications including smart grids, smart cities [5], and health management [7]. However, the increasingly invisible, dense and pervasive collection, processing and dissemination of data in the midst of people's private lives gives rise to serious security and privacy concerns. Several intrinsic features of IoT amplify its security and privacy challenges including: lack of central control, heterogeneity in device resources, multiple attack surfaces, context specific risks, and scale.

In this paper, we argue that BlockChain (BC) technology that underpins Bitcoin, the first cryptocurrency system launched in 2008 [10], can provide an effective solution to IoT privacy and security. BC security mainly comes from a cryptographic puzzle known as Proof of Work (POW) used for appending (mining) new blocks into the BC. BC also offers a high level of privacy by using a changeable Public Key (PK) as the users' identity. BC has been adopted for a number of non-monetary applications, e.g. proof of location [2], distributed storage systems [12], and health care data [13]. These salient features of BC make it attractive for delivering distributed privacy and security in IoT. However, applying BC to IoT is not straightforward. Several key challenges need to be addressed including: (i) *high resource* requirements due to the use of POW; (ii) *scalability* issues that originate from the need to achieve consensus among miners; (iii) *high delays* attributed to POW and mechanisms to prevent double spending (which may be important for cryptocurrency yet not for IoT).

The main contribution of this paper is to introduce a new type of BC that is optimized for IoT. To exemplify our idea, we use the scenario of a smart home in the rest of the paper. However, the architecture is application-agnostic for diverse IoT use cases. Our lightweight instantiation of BC retains the underlying privacy and security benefits, while eliminating the aforementioned issues. We adopt a hierarchical structure to optimize resource consumption and increase network scalability. Our framework consists of three tiers which are: smart home, overlay network, and cloud storage. IoT devices in the smart home benefit from a private Immutable Ledger (IL), that acts similar to BC but is managed centrally, and symmetric encryption to reduce the processing overhead, while higher resource devices jointly create a distributed overlay that instantiates a public BC. Communications among entities in different tiers are known as transactions that are grouped into blocks. Blocks are appended to the BC without solving the POW, which decreases the appending overhead significantly. Verified signed transactions are available for the entire network immediately. This significantly reduces the delay of IoT transactions, such as data access or queries. A distributed trust method is employed in the overlay to decrease the processing overhead in validating new blocks. We qualitatively discuss the robustness of the proposed method against attacks, and we evaluate the packet and processing overhead quantitatively through simulations.

IoTDI 2017, Pittsburgh, PA USA

DOI: http://dx.doi.org/10.1145/3054977.3055003

The rest of the paper is organized as follows. Section 2 reviews IoT privacy and security and introduces the basic concepts of Bitcoin BC. Section 3 discusses the proposed architecture. Details of transaction handling are discussed in Section 4. Analysis and evaluation is presented in Section 5, while Section 6 concludes the paper.

2 LITERATURE REVIEW

2.1 Privacy and security in IoT

Security in IoT is challenging due to low resource capabilities of the vast majority of devices, immense scale, heterogeneity among the devices, and lack of standardization. Moreover, many of these IoT devices collect and share large amounts of data from our personal spaces, thus opening up significant privacy concerns. To protect user's privacy, the authors in [1] defined different privacy zones for diverse types of data. Each zone has an associated context based policy checking method, that is checked by a Home Security Hub prior to accepting join or re-join requests to protect user data against unauthorized data sharing. However, the possibility of accessing smart devices directly bypassing the hub is not considered.

Authors in [11] demonstrated that a wide variety of off-the-shelf IoT devices lack fundamental security considerations. The authors proposed a Security Management Provider that is responsible for controlling access to data and devices by using fixed or dynamic content-based policies. However, protecting user privacy while revealing personal data is not addressed. A comprehensive study on IoT security appears in [8]. In the context of smart homes, the authors discussed the security implications involved in sensor distribution, data capture, and forwarding data to the gateway.

As proposed in [3] by using safe or aggregated answers the user can send as little data as possible to the service provider. In some instances, their proposed method may even add noise to the data to ensure privacy. Although these methods improve data privacy, for some applications, expressly in a smart home setting, the noisy version of the data may lead to inaccurate services.

In summary, despite recent proposals for providing security and privacy in IoT, three challenges still need to be addressed:

- *Resource optimization:* Resource constrained devices in IoT are not suited for high level, complex security methods.
- *Privacy:* Protecting user privacy while revealing diverse types of data.
- *Centralization:* Centralized methods tend to be inappropriate for IoT and bring the challenges of single point of failure, many-to-one traffic, and reduced scalability.

2.2 Bitcoin BC

BC is an immutable ledger of blocks that underpins Bitcoin and maintains network transactions. Network participants, that are known by a changeable Public Key (PK), manage the BC in a distributed manner. There are certain nodes that are responsible for appending new blocks to the BC. These nodes are called miners and the appending process is called mining. Bitcoin mining involves solving a resource consuming cryptographic puzzle known as Proof of Work (POW). BC is an attractive technology for addressing the mentioned security and privacy challenges in IoT as a result of its key features including decentralization, anonymity and security.

Authors in [6] proposed a BC-based multi-tier method to share IoT users' data with organizations and people. However, they assume that IoT devices have sufficient resources for solving the POW which may not always be true. In fact, solving the POW for Bitcoin requires very sophisticated hardware and cannot be achieved even using off-the-shelf computers. Adopting BC in IoT is not straightforward and will require addressing the critical challenges of high resource, latency, bandwidth utilization and low scalability. This paper takes a step in this direction.

3 BC-BASED SMART HOME ARCHITECTURE

The proposed architecture, shown in Figure 1, includes three tiers, namely the smart home, the overlay, and the cloud storage. In each tier, entities use transactions to communicate with each other. In the following we briefly introduce three tiers.

3.1 Smart home

The smart home is comprised of IoT devices, local IL, and a local storage as shown in the bottom left of Figure 1. Each home has a local private IL that is similar to a BC but is managed centrally by the smart home manager (SHM). The SHM processes all incoming and outgoing transactions and uses a shared key for local communications with IoT devices and local storage. The local IL maintains a policy header defined by the home owner to authorize the received transactions. Local devices inside the home or overlay nodes might generate transactions in order to share, request, or store data. Our previous work [4] discusses the smart home tier in greater detail.

3.2 Overlay

The overlay is a peer-to-peer network that brings the distributed feature to our architecture. The constituent nodes, known as overlay nodes, could be SHMs, other high resource devices in the home, or the user's smartphone or personal computer. To decrease network overhead and delay, nodes in the overlay are grouped in clusters and each cluster elects a Cluster Head (CH) using methods such as in [9]. Each CH has a unique PK, known by other CHs in the overlay, used for generating new blocks so that other CHs could authorize the block generator. Each node is free to change its cluster if it experiences excessive delays. Moreover, nodes in the cluster can elect a new CH at any time. In this paper, it is assumed that the aforementioned steps are performed at start-up. Each CH maintains the following lists:

PK of requesters: the list of PKs that are allowed to access data for the SHMs connected to this cluster. An example of which might be a SP that provides certain services for the smart home devices.

PK of requestees: the list of PKs of SHMs connected to this cluster that are allowed to be accessed.

The overlay CHs maintain a public BC, which has a ledger for each overlay node that shows the history of transactions sent by the overlay user and is used to gain reputation. The transactions are generated by users or devices to request or share data with others. The overlay has multisig transactions, meaning they need to be signed by two entities - requester and requestee to be treated as a valid transaction. Additionally, each transaction has two outputs

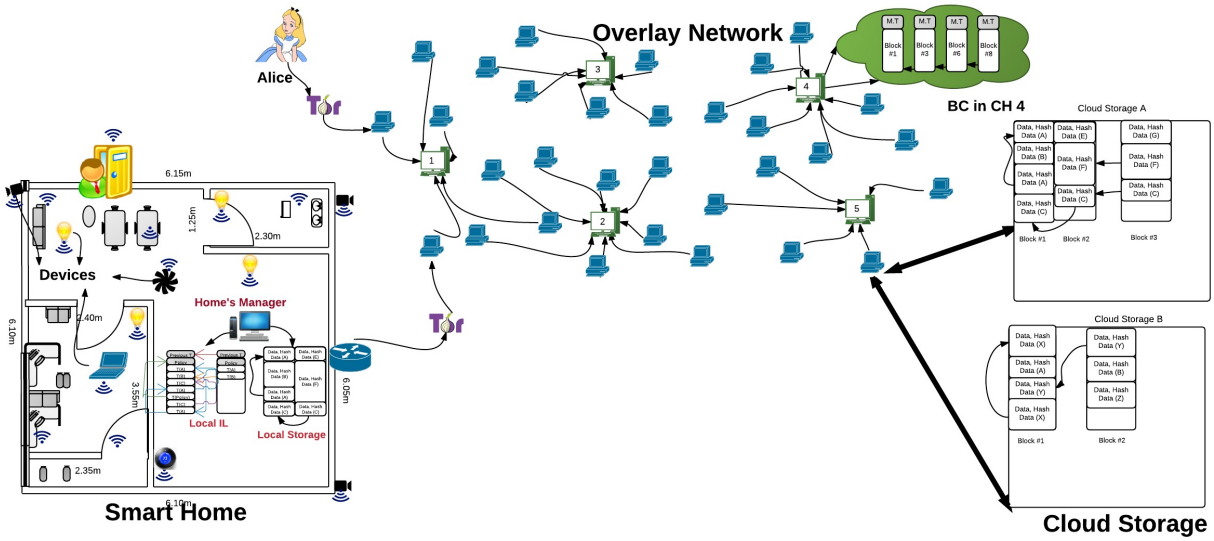

Figure 1: BC-based smart home.

which indicates the total number of accepted or rejected transactions, by the requestees, that are created by the generator of the current transaction.

Distributed Trust and POW elimination: We use distributed trust to ensure that the received blocks are valid and to decrease the overhead for verifying blocks compared to Bitcoin. This works as follows: A user that initially has no transaction history is suspected to be malicious and all his transactions are verified. To verify a transaction the first step is to confirm that the requester has the right to append transactions to the specified ledger which is done by comparing the hash of the current transaction PK with the output PK of the previous transaction. Following this, the requester signature is verified using his PK in the transaction. Next, the verifier controls that only one of the outputs of the current transaction, i.e. the number of successful transactions or the number of rejected transactions, is increased only by one. If the steps passed successfully, the transaction is verified.

In the overlay network each CH maintains a trust rating for other CHs based on direct and indirect evidence. CH A has direct evidence about CH B if it verified a block generated by B. If A receives a block generated by B and A has no trust in B, but one or more CHs signed the block as a valid block, then A has indirect evidence about B. When a CH generates a new block it has to create a multisig transaction which is used for evaluating trust. The CH then sends both the block and multisig transaction to its neighboring CHs. Neighbor CHs check the multisig transaction for direct or indirect evidence. If the neighbor CH has direct evidence with the block generator or other CHs who signed the multisig transaction, then it randomly verifies a portion of the transactions in the block by checking their signatures. The portion of transactions that need to be verified change as a function of the number of succesfully verified blocks for the corresponding CH. Figure 2 shows an example of such a "trust table". Note that, a certain portion of transactions must be verified even when a CH is deemed to be trustworthy to protect the network against CHs which may have been recently compromised. If a CH has no direct evidence with the block generator or those who signed it, then it checks all transactions in the received block.

If a single block is generated by more than one CH, then other CHs would accept the one that is appended by the CH which has the highest trust to decrease the validation processing overhead due to the act of distributed trust. This may result in a forked BC. However, all forked blocks are assumed to be valid. The CHs append newly received blocks to the longest chain of blocks which ensures BC consistency.

3.3 Cloud storage

The cloud storage groups user's data in identical blocks associated with a unique block-number. The block-number is used by the SHM for authentication along with the hash of stored data. If the storage can successfully locate data with the received block-number and hash from the SHM, then the user is authenticated. Received data packets from users are stored in a First-In-First-Out order in blocks along with the hash of stored data as shown in the bottom right of Figure 1. It is worth noting that each home owner can either create different ledgers of data in storage for each of its devices or a single common ledger for all of its devices. The former is

Number of Successfully verfied blocks	10	20	30	40	50
Percentage of transactions should be verfied	80%	60%	40%	30%	20%

Figure 2: Direct evidence trust table.

particularly useful if the owner wishes to provide access to all data of a particular device to a SP.

4 TRANSACTION HANDLING

In this section, we focus on how transactions are handled in our framework.

4.1 Storing

Let's assume that Alice has created an account in a cloud storage facility and set up permissions for her thermostat to upload data to this facility. During the bootstrapping process, the cloud storage returns a pointer to the first block of data. When the smart thermostat needs to store data in the cloud storage, it sends its data to the SHM. After checking permissions and extracting the previous block-number and hash from the local IL, the SHM creates a random ID and sends data to the storage with this ID. It is assumed that at any given time, two nodes cannot have the same ID. The storage checks the validity of the transaction by locating data using given parameters and also confirms that there is space available in the cloud storage. If so, it calculates a hash of received data packets and compares it with the received hash in the transaction. If the two hashes match, then data packets are stored in the storage and the new block-number is encrypted with the SHM PK, to make sure that only the true SHM can read the new block-number, and sent to the SHM. Next, the signed hash of data is signed by the storage and sent to the overlay network to be stored in the overlay BC. This ensures that any further changes in the user's data are visible to all.

In each smart home there is a private secure storage managed by the SHM. The SHM distributes a shared key between authorized IoT devices and the local storage. IoT devices use this key for generating store transaction. Local storage is managed by the owner of the home and is trusted. Therefore, there is no additional overhead for accounting.

4.2 Accessing

To access stored data of a device, the SP creates and signs the requester part of a multisig transaction. The SP then sends this to its own CH. The CH checks both lists of PKs defined in Section 3.2. If either the multisig transaction's requester is in the CH requester's key list, or the transaction's requestee is in its requestee's key list, then it broadcasts the transaction to its own cluster. Otherwise, the transaction is broadcast to other CHs. When the requestee (SHM) receives the multisig transaction, it authorizes the SP by checking the local policy in the local IL. If so, the SHM requests data from the storage, encrypts them with requester's PK, and sends them to the requester (the SP). After sending data for the requester, the SHM should store the multisig transaction in the local IL to keep the history of transactions. In addition, the SHM sends the multisig transaction to its CH to be stored in the overlay BC as a history of requester transactions.

4.3 Monitoring

A monitor transaction is launched by overlay nodes to monitor real-time data of a device. Monitor transaction processing is similar to access transaction. The only difference is that the SHM sends real-time data of requested device. A monitor transaction can also be used to set up continuous stream of real-time data from a particular device.

5 EVALUATION

In this section, we first discuss the differences between the proposed IL, overlay BC, and the Bitcoin BC. Then, we present a qualitative discussion on how our proposed solution addresses common security and privacy attacks. In addition, we evaluate the associated overheads.

Recall that our framework optimizes the Bitcoin BC for IoT by presenting different tiers of BC. Each tier has unique features that makes it differ with other tiers and Bitcoin BC. We summarize the key differences between our BCs and Bitcoin BC in Table 1.

5.1 Security and privacy analysis

It is assumed that the adversary can be the CH, a device in the home, a node in the overlay, or the cloud storage. Adversaries are able to sniff communications, discard transactions, create false transactions and blocks, change or delete data in storage, link a user's transactions to each other, and sign fake transactions to legitimize colluding nodes. However, they are not able to break the encryption. The main classes of threats are:

Accessibility threats that prevents the legitimate user from getting access to her data or services.

Anonymity threats that finds the real world identity of a user by analyzing the anonymous transactions and other publicly available information.

Authentication and access control threats in which the adversary tries to authenticate herself as a legitimate user in order to gain access to data.

We consider the following attacks that **threaten accessibility:**

Denial of Service (DOS) Attack: In this attack, the attacker sends a large number of transactions to a target to break its availability. The use of requester and requestee PK lists in the CHs in our architecture diminishes the effect of this attack as a packet would not be relayed to a SHM unless its key is on these two key lists. Moreover, if a CH receives several unsuccessful access requests from a particular PK, it can block that PK by removing the corresponding key from the CH key lists. However, the adversary can succeed in a DOS attack if it uses different PKs for the attack.

Modification Attack: In this attack the adversary may seek to change or delete stored data for a particular user. To launch this attack, the adversary would have to compromise the cloud storage security. However, the target user would be able to detect any change in his stored data by comparing hash of the data in the cloud with stored hash in its local IL as discussed in Section 3.2.

Dropping Attack: To launch this attack, the adversary should have control over a CH(s) and then drop all received transactions and blocks. Such an attack would be detected since nodes that belong to the constituent clusters would not receive any transactions or service from the overlay. In this instance, they would elect a new CH.

Appending Attack: To launch this attack, the adversary must control multiple CHs that work cooperatively. To increase the indirect evidence rating, the malicious CHs sign the multisig transaction along with the block claiming that they have verified the block.

Table 1: Comparison of the Bitcoin BC, local IL, and public BC employed in our proposed architecture.

#	Feature	BC in Bitcoin	Local IL	Overlay BC
1	BC Visibility	Public	Secure/ Private	Public
2	Transaction chaining	Input / Output	Previous T of the same D	T are chained to each other/ Output.
3	Transaction mining	All Ts	All Ts	Arbitrary Ts
4	Mining requirement	POW	None	None
5	Forking	Not allowed	Allowed	Allowed
6	Double Spending	Prohibited	Not applicable	Not applicable
7	Transaction verification	Signature	No verification	Signatures
8	Transaction parameters	input, output, coins.	Block-number, hash of data, time, output, PK, policy rules.	Output, PKs.
9	Transaction dissemination	Broadcast	Unicast	Unicast/ multicast
10	Deference in block header	puzzle	policies	Not applicable
11	New block verification	Blocks and Ts in blocks	No verification	Blocks and Ts in blocks
12	BC control	No one	Owner	No one
13	Miner checks	No one	No one	Other CHs and nodes.
14	Miner trust	Miners are all the same.	SHMs are all the same.	Different levels of trust are defined.
15	Miner joining overhead	download all blocks in BC.	download all blocks in IL.	Download all blocks in BC
16	Miner selection	Self-selection	Owner chooses the SHM.	Nodes in cluster choose one node in the cluster as CH.
17	Miner rewards	Coins	Nothing	Not defined
18	Pool mining	allowed	Cannot be defined.	Cannot be defined.
19	Malicious miner	Allowed to join	not possible	Allowed to join
20	Effects of 51% attack	double spending	not possible.	Increases the possibility of appending false blocks
21	Encryption method	Public/ private keys	No need	Public/private keys, shared key

T stands for transaction. D stands for device. H stands for home

The fake block might be a block with either one or more than one false transaction. In the proposed trust method a random portion of transactions in a received block are always validated based on the trust level between the CH and the block generator. Therefore, even if more than 51% of CHs in the overlay signs the current block as valid, there is still possibility that the honest CH detects the false block (this attack is called 51% attack in Bitcoin). Our framework exhibits graceful degradation in that the resilience degrades as the number of compromised nodes increases. The probability of detecting the fake block is a function of the number of total CHs, which presents an interesting trade-off for the system designer.

To break user **anonymity**, an attacker may try to deanonymize a user by linking different data associated with the same anonmyous identity. To protect against such linking attacks, overlay nodes can use different PKs for their communications with overlay nodes or cloud storage, so that each single transaction would have a unique ID and it would not be possible to link them together.

The next class of threats is against **authentication and access control**. It has been shown in recent research [11] that it is possible for an attacker to take control of a smart home device or introduce a fake device to a home network. Our design employs a hierarchical defence against these attacks. First, there is a central SHM that controls all incoming and outgoing packets and prevents smart home devices from being directly accessed from the Internet. If the SHM detects a packet that does not adhere to the policies defined by the owner, the packet is dropped. The second defence is that all devices in the home are required to have a genesis transaction in IL that allows them to initiate communication with the SHM and other devices. A device without a corresponding genesis transaction is isolated from the network. This prevents an attacker from introducing unauthorized devices to the network.

5.2 Performance evaluation

In this section, we first quantitatively evaluate various overheads, then, we discuss simulation results. Table 2 illustrates average performance metrics for the key transactions in our proposed architecture. These are expressed as a function of various design parameters such as packet, memory and computation overhead and delay.

To evaluate the performance of the proposed framework, we conduct simulations using NS3 simulator, focusing on the overlay network. Note that the smart home performnace was evaluated in our previous work [4]. We simulate a network of 50 nodes, of which 13 are CHs, to study traffic and processing overhead of our design. We run the simulation for 60 seconds during which a total of 960 transactions are created. The given results are the average of 10 runs of the simulation.

Table 2: Overhead evaluation.

	Appending & Trust	Transactions	CH joining
Packet overhead	O(N)	O ((N*S)/2)	O (BS)
Delay	O (N/TL)	O ((N*S)/2)	O (B*T)
Computation overhead	O (N/TL)	O (N)	O (B*T)
Memory overhead	O (BS)	O(1)	O(BS)

N: The number of Clusters , B: Blocks in each BC , BS: Block Size , TL: Trust Level , S: Hops between source and storage , T: Transactions in each block

Traffic overhead: This refers to the amount of bytes transmitted in the overlay for managing the public BC. To compare our results, a Bitcoin network with 50 nodes is simulated as the baseline. Simulation results show that our method generated roughly 37MB data while Bitcoin generated 138MB data. The main reason for this difference is due to clustering and the fact that the BC packets are only broadcast between CHs while in Bitcoin packets are broadcast to all nodes.

Processing overhead: This metric refers to the time consumed by CHs to verify new blocks. As a benchmark, we consider an overlay network with 50 nodes but which does not implement distributed trust. Simulation results for evaluating the processing overhead are presented in Figure 3. Initially, the processing time is equal for both methods as there is no trust between CHs. As more blocks are generated by the CHs, our distributed trust strategy kicks in whereby only a portion of the transactions in new blocks need to be validated. Consequently, the processing overhead with our method is lower than the method without trust. In our method, once 50 blocks are generated, the trust rating between CHs reaches the highest level. From here on the number of transactions that need to be verified for each new block remains fixed and thus the processing time remains steady. Overall, our method decreases the processing time roughly by 50%. It should be noted that the distributed trust strategy presented in this paper could also be employed in other BC based systems provided it does not compromise network security.

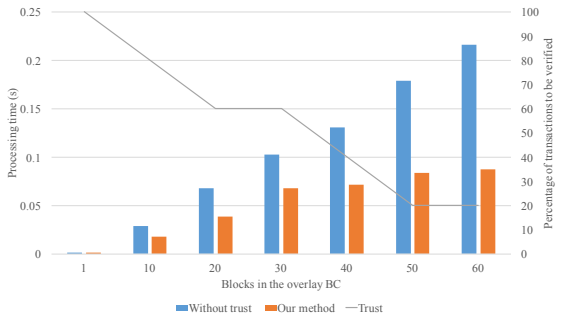

Figure 3: Processing overhead evaluation.

6 CONCLUSION

BlockChain (BC) holds promise for privacy and security in IoT. However, applying BC in IoT is not straightforward due to various associated challenges including: high resource consumption, scalability, and processing time. In this paper, we proposed an optimized BC that eliminates the overhead associated with the classic BC while retaining its security and privacy benefits. The proposed BC requires no mining and thus incurs no additional delays in processing generated transactions. It employs a hierarchical architecture that uses a centralized private Immutable Ledger (IL) at the local IoT network level to reduce overhead, and a decentralized public BC at higher end devices for stronger trust. A distributed trust method is employed to decrease new block processing overhead. Security and privacy of the design is evaluated that showing the robustness of the new architecture against several attacks. Simulation results demonstrate that the method has low packet and processing overhead. Future work includes more extensive evaluation of the impact of design choices on security and overhead in the overlay, and a comprehensive analysis on consensus and security of our framework.[1]

REFERENCES

[1] Abdullahi Arabo, Ian Brown, and Fadi El-Moussa. 2012. Privacy in the age of mobility and smart devices in smart homes. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*. IEEE, 819–826.

[2] Giacomo Brambilla, Michele Amoretti, and Francesco Zanichelli. 2016. Using Block Chain for Peer-to-Peer Proof-of-Location. *arXiv preprint arXiv:1607.00174* (2016).

[3] Yves-Alexandre de Montjoye, Erez Shmueli, Samuel S Wang, and Alex Sandy Pentland. 2014. openpds: Protecting the privacy of metadata through safeanswers. *PloS one* 9, 7 (2014), e98790.

[4] Ali Dorri, Salil S Kanhere, Raja Jurdak, and Praveen Gauravaram. 2017. Blockchain for IoT Security and Privacy: The Case Study of a Smart Home. In *IEEE Percom workshop on security privacy and trust in the internet of thing*. IEEE.

[5] Aditya Gaur, Bryan Scotney, Gerard Parr, and Sally McClean. 2015. Smart city architecture and its applications based on IoT. *Procedia Computer Science* 52 (2015), 1089–1094.

[6] Sayed Hadi Hashemi, Faraz Faghri, Paul Rausch, and Roy H Campbell. 2016. World of Empowered IoT Users. In *2016 IEEE First International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 13–24.

[7] Moeen Hassanalieragh, Alex Page, Tolga Soyata, Gaurav Sharma, Mehmet Aktas, Gonzalo Mateos, Burak Kantarci, and Silvana Andreescu. 2015. Health monitoring and management using Internet-of-Things (IoT) sensing with cloud-based processing: Opportunities and challenges. In *Services Computing (SCC), 2015 IEEE International Conference on*. IEEE, 285–292.

[8] Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. 2014. Security of the internet of things: Perspectives and challenges. *Wireless Networks* 20, 8 (2014), 2481–2501.

[9] Apostolos Kousaridas, Stefanos Falangitis, Panagis Magdalinos, Nancy Alonistioti, and Markus Dillinger. 2015. SYSTAS: Density-based algorithm for clusters discovery in wireless networks. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2015 IEEE 26th Annual International Symposium on*. IEEE, 2126–2131.

[10] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).

[11] Vijay Sivaraman, Hassan Habibi Gharakheili, Arun Vishwanath, Roksana Boreli, and Olivier Mehani. 2015. Network-level security and privacy control for smart-home IoT devices. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*. IEEE, 163–167.

[12] Shawn Wilkinson, Jim Lowry, and Tome Boshevski. 2014. *Metadisk a blockchain-based decentralized file storage application*. Technical Report. Technical Report. http://metadisk. org/metadisk. pdf.

[13] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. 2016. Healthcare Data Gateways: Found Healthcare Intelligence on Blockchain with Novel Privacy Risk Control. *Journal of medical systems* 40, 10 (2016), 218.

[1]This paper is supported by Data61, CSIRO, Australia.