Blockchain for IoT Security and Privacy: The Case Study of a Smart Home

Ali Dorri*, Salil S. Kanhere *, Raja Jurdak† and Praveen Gauravaram‡
*School of Computer Science and Engineering
The University of New South Wales
Sydney, Australia
Email:(ali.dorri,salil.kanhere)@unsw.edu.au
†CSIRO
Brisbane, Queensland, Australia.
Email: Raja.Jurdak@csiro.au
‡ Tata Consultancy Services, Australia.
Email: p.gauravaram@tcs.com

***Abstract*—Internet of Things (IoT) security and privacy remain a major challenge, mainly due to the massive scale and distributed nature of IoT networks. Blockchain-based approaches provide decentralized security and privacy, yet they involve significant energy, delay, and computational overhead that is not suitable for most resource-constrained IoT devices. In our previous work, we presented a lightweight instantiation of a BC particularly geared for use in IoT by eliminating the Proof of Work (POW) and the concept of coins. Our approach was exemplified in a smart home setting and consists of three main tiers namely: cloud storage, overlay, and smart home. In this paper we delve deeper and outline the various core components and functions of the smart home tier. Each smart home is equipped with an always online, high resource device, known as "miner" that is responsible for handling all communication within and external to the home. The miner also preserves a private and secure BC, used for controlling and auditing communications. We show that our proposed BC-based smart home framework is secure by thoroughly analysing its security with respect to the fundamental security goals of confidentiality, integrity, and availability. Finally, we present simulation results to highlight that the overheads (in terms of traffic, processing time and energy consumption) introduced by our approach are insignificant relative to its security and privacy gains.**

I. Introduction

Internet of Things (IoT) consists of devices that generate, process, and exchange vast amounts of security and safety-critical data as well as privacy-sensitive information, and hence are appealing targets of various cyber attacks [1]. Many new networkable devices, which constitute the IoT, are low energy and lightweight. These devices must devote most of their available energy and computation to executing core application functionality, making the task of affordably supporting security and privacy quite challenging. Traditional security methods tend to be expensive for IoT in terms of energy consumption and processing overhead. Moreover many of the state-of-the-art security frameworks are highly centralized and are thus not necessarily well-suited for IoT due to the difficulty of scale, many-to-one nature of the traffic, and single point of failure [2]. To protect user privacy, existing methods often either reveal noisy data or incomplete data, which may potentially hinder some IoT applications from offering personalised services [3]. Consequently, IoT demands a lightweight, scalable, and distributed security and privacy safeguard. The Blockchain (BC) technology that underpins Bitcoin the first cyptocurrency system [4], has the potential to overcome aforementioned challenges as a result of its distributed, secure, and private nature.

Bitcoin users that are known by a changeable Public Key (PK), generate and broadcast transactions to the network to transfer money. These transactions are pushed into a block by users. Once a block is full, the block is appended to the BC by performing a *mining* process. To mine a block, some specific nodes known as *miners* try to solve a resource consuming cryptographic puzzle named Proof of Work (POW) [5], and the node that solves the puzzle first mines the new block to the BC. In our previous work [6], we argued that adopting BC in the context of IoT is not straightforward and entails several significant challenges such as: *high resource* demand for solving the POW, *long latency* for transaction confirmation, and *low scalability* that is a result of broadcasting transactions and blocks to the whole network. We proposed a novel instantiation of BC by eliminating the concept of POW and the need for coins. Our proposed framework relies on hierarchical structure and distributed trust to maintain the BC security and privacy while making it more suitable for the specific requirement of IoT. We exemplified our ideas in the context of a smart home, but our framework is application agnostic and can be applied in other IoT contexts. The design consists of three core tiers that are: smart home, cloud storage, and overlay. Smart devices are located inside the smart home tier and are centrally managed by a miner. Smart homes constitute an overlay network along with Service Providers (SP), cloud storages, and users' smartphones or personal computers as illustrated in Figure 1. The overlay network is akin to the peer-to-peer network in Bitcoin and brings the distributed feature to our architecture. To decrease network overhead and delay, nodes in the overlay are grouped into clusters and each cluster elects a Cluster Head (CH). The overlay CHs maintain a public

978-1-5090-4338-5/17/$31.00 ©2017 IEEE

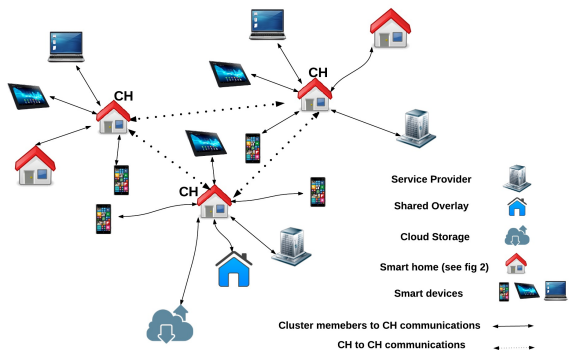


Fig. 1. Overview of the proposed BC-based architecture discussed in more details in [6].

BC in conjunction with two key lists. These key lists are: *requester* key lists that is the list of overlay users' PKs that are allowed to access data for the smart homes connected to this cluster; *requestee* key lists that is the list of PKs of smart homes connected to this cluster that are allowed to be accessed. Cloud storage is used by the smart home devices to store and share data. We discussed details of the overlay and the cloud storage in our previous work [6].

This paper's contribution is to give a comprehensive discussion on the details of the smart home tier in our design. We first outline how the IoT devices are initialised and then explain how transactions are processed. A local and private BC is employed for providing secure access control to the IoT devices and their data. Besides, the BC generates an immutable time-ordered history of transactions that is linkable to other tiers for giving specific services. The design security comes from diverse features including: (1) indirectly accessible devices; and (2) different transaction structures in the smart home and the overlay. To achieve a lightweight security, symmetric encryption is employed for smart home devices. We provide qualitative arguments to demonstrate that the smart home tier achieves confidentiality, integrity, and availability and also discuss how key security attacks such as linking attack [7] and Distributed Denial of Service (DDOS) are thwarted. Finally, we present quantitative results using simulations and show that the overheads induced by our framework are relatively small.

The rest of the paper is organized as follow: In Section II we present the main components of the design. The BC-based smart home is discussed in depth in Section III. Simulation results and security discussions are presented in Section IV. Section V summarizes related works, and finally Section VI concludes the paper.

II. Core Components

This section discusses the main smart home components as shown in Figure 2.

A. Transactions

Communications between local devices or overlay nodes are known as *transactions*. There are different transactions in the BC-based smart home each designed for a specific function. *Store* transaction is generated by devices to store data. An *access* transaction is generated by a SP or the home owner to access the cloud storage. A *monitor* transaction is generated by the home owner or SPs to periodically monitoring a device information. Adding a new device to the smart home is done via a *genesis* transaction and a device is removed via a *remove* transaction. All of the aforementioned transactions use a shared key to secure the communication. Lightweight hashing [8] is employed to detect any change in transactions' content during transmission. All transactions to or from the smart home are stored in a local private BlockChain (BC).

B. Local BC

In each smart home, there is a local private BC that keeps track of transactions and has a policy header to enforce users' policy for incoming and outgoing transactions. Starting from the genesis transaction, each device's transactions are chained together as an immutable ledger in the BC. Each block in the local BC contains two headers that are block header and policy header as shown at the top of Figure 2. The block header has the hash of the previous block to keep the BC immutable. The policy header is used for authorizing devices and enforcing owner's control policy over his home. As shown in the top right corner of Figure 2, the policy header has four parameters. The "Requester" parameter refers to the requester PK in the received overlay transaction. For local devices, this field is equal to the "Device ID" as shown in the fourth row of the proposed policy header in Figure 2. The second column in the policy header, indicates the requested action in the transaction, which can be: *store* to store data locally, *store cloud* to store data on the cloud storage, *access* to access stored data of a device, and *monitor* to access real-time data of a particular device. The third column in the policy header is the ID of a device inside the smart home, and finally, the last column indicates the action that should be done for the transaction that matches with the previous properties.

Besides the headers, each block contains a number of transactions. For each transaction five parameters are stored in the local BC as shown in the top left corner of the Figure 2. The first two parameters are used to chain transactions of the same device to each other and identify each transaction uniquely in the BC. The transaction's corresponding device ID is inserted on the third field. "Transaction type" refers to the type of transaction that can be genesis, access, store, or monitor transactions. The transaction is stored on the fifth field if it comes from the overlay network, otherwise, this filed is kept blank. The local BC is kept and managed by a local *miner*.

C. Home miner

Smart home miner is a device that centrally processes incoming and outgoing transactions to and from the smart home. The miner could integrate with the home's Internet gateway or a separate stand-alone device, e.g. F-secure [9], could be placed between the devices and the home gateway.

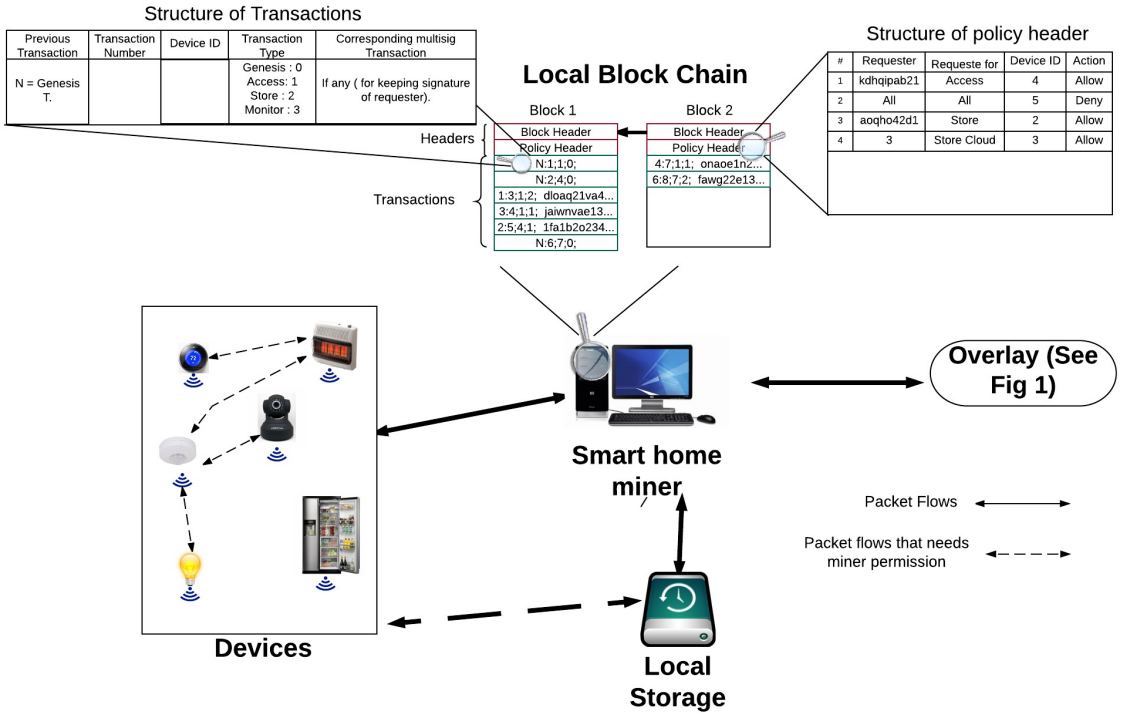


Fig. 2. Overview of the Smart home: The smart home consists of IoT devices, local storage (see section II.D), the miner (see section II.C), and the local BC (see section II.B).

Similar to existing central security devices, the miner authenticates, authorizes, and audits transactions. In addition the miner also accomplishes the following additional functions: generating genesis transactions, distributing and updating keys, changing the transactions structure, and forming and managing the cluster. The miner collects all transactions into a block and appends the full block to the BC. To provide additional capacity, the miner manages a *local storage*.

D. Local Storage

Local storage is a storing device e.g. backup drive that is used by devices to store data locally. This storage can be integrated with the miner or it can be a separate device. The storage uses a First-in-First-out (FIFO) method to store data and stores each devices' data as a ledger chained to the device's starting point.

III. The BC-based Smart Home

First, we discuss the initialization steps, transactions handling, and shared overlay.

A. Initialization

In this section, we describe the process of adding devices and policy header to the local BC. To add a device to the smart home, the miner generates a genesis transaction by sharing a key with the device using generalized Diffie-Hellman [10]. The shared key between the miner and the device is stored in the genesis transaction. As for defining policy header, the home owner generates its own policies according to our proposed policy structure in Figure 2 and adds the policy header to the first block. The miner uses the policy header in the latest block in BC; therefore, to update the policy the owner should update the latest block's policy header.

B. Transaction Handling

The smart devices may communicate directly with each other or with entities external to the smart home. Each device inside the home may request data from another internal device to offer certain services, e.g., the light bulb requests data from the motion sensor to turn on the lights automatically when someone enters the home. To achieve user control over smart home transactions, a shared key should be allocated by the miner to devices which need to directly communicate with each other. To allocate the key, the miner checks the policy header or asks for permission from the owner and then distributes a shared key between devices. After receiving the key, devices communicate directly as long as their key is valid. To deny the grant permission, the miner marks the distributed key as invalid by sending a control message to devices. The benefits of this method is twofold: on one hand, the miner (and so the owner) has a list of devices that share data, and on the other, the communications between devices are secured with a shared key.

Storing data on the local storage by devices is the other possible transaction flow inside the home. To store data locally,

each device needs to be authenticated to the storage that is done using a shared key. To grant the key, the device needs to send a request for the miner and if it has storing permission, the miner generates a shared key and sends the key for the device and the storage. By receiving the key, the local storage generates a starting point that contains the shared key. Having the shared key, the device can store data directly in the local storage.

The devices may demand to store data on the cloud storage that is known as *store transaction*. Storing data in the cloud is an anonymous process that is discussed in [6]. To store data the requester needs a starting point that contains a block-number and a hash used for anonymous authentication purpose. The cloud storage may be either owned and managed by the SP (e.g. Nest thermostat) or paid for and managed by the home owner (e.g. Dropbox). In the former instance, the miner requests for the starting point by generating a signed transaction with the device key. In the latter case, payment is done through Bitcoin. In either storage type, after receiving a request the storage creates a starting point and sends it to the miner. When a device needs to store data on the cloud storage, it sends data and the request to the miner. By receiving the request, the miner authorizes the device for storing data on the cloud storage. If the device has been authorized, the miner extracts the last block-number and hash from the local BC, and creates a store transaction and sends it along with the data to the storage. After storing data, the cloud storage returns the new block-number to the miner that is used for further storing transactions.

The other possible transactions are *access* and *monitor* transactions. These transactions are mainly generated by either the home owner to monitor the home when he is outside or by SPs to process devices' data for personalized services. By receiving an *access transaction* from nodes in the overlay, the miner checks whether the requested data is on the local or the cloud storage. If data is stored in the local storage, the miner requests data from the local storage and sends it to the requester. On the other hand, if the data is stored in the cloud, the miner either requests data from the cloud storage and sends it to the requester, or sends the last block-number and hash to the requester. The latter scenario empowers the requester to read entire data stored by the device in cloud storage and is suitable when the stored data are for a unique device. Otherwise, the user's privacy might be endangered as part of a linking attack which is discussed later in Section IV.

By receiving a *monitor transaction*, the miner sends current data of the requested device to the requester. If a requester is allowed to receive data for a period of time then the miner sends data periodically until the requester sends a close request to the miner and abolish the transaction. The monitor transaction enables home owners to watch cameras or other devices in which send periodic data. In order to avoid overhead or possible attacks, the owner should define a threshold in minutes for the periodic data. If the time in which the miner is sending data for the requester reaches to the threshold, then the connection is terminated by the miner.

C. Shared overlay

When an individual has more than one home, he needs separate miners and storage for each of the homes. To reduce the cost and managing overhead in this instance, a shared overlay is defined. The shared overlay consists of at least two smart homes that are managed centrally as a single home by a shared miner. The shared overlay is similar to the smart home, however, the structure of the shared BC is different to that of a smart home. In the shared BC each home has a genesis transaction and the genesis transaction of all devices are chained to their home's genesis transaction by the shared overlay miner. Another difference in the shared overlay is regarding the communications between the homes with the miner. Devices that are in the same home with the miner experience no change, while for devices in other homes a Virtual Private Network (VPN) connection is established between the Internet gateway in each home and the miner of the shared overlay that routes the packets to the shared miner.

IV. Evaluation and Analysis

This section provides a complete discussion on the security, privacy, and performance of the BC-based smart home.

A. Security Analysis

There are three main security requirements that need to be addressed by any security design, namely: Confidentiality, Integrity, and Availability, known as CIA [11]. Confidentiality makes sure that only the authorized user is able to read the message. Integrity makes sure that the sent message is received at the destination without any change, and availability means that each service or data is available to the user when it is needed. Employed methods to achieve the first two requirements are discussed in Section III. To increase smart home availability devices are protected from malicious requests. This is achieved by limiting the accepted transactions to those entities with which each device has established a shared key. Transactions received from the overlay are authorized by the miner before forwarding them on to the devices. Furthermore, it can be argued that our BC-based framework only introduces a marginal increase in the transaction processing delays as compared to existing smart home gateway products. There is also an additional one-time delay during initialization for generating and distributed shared keys. In summary, the additional delays are not significant and do not impact the availability of the smart home devices.

Table I summarizes how our framework achieves the aforementioned security requirements

Next we analyze the effectiveness of our solution to prevent two critical security attacks that are particularly relevant for smart homes. The first one is Distributed Denial of Service (DDOS) attack in which the attacker uses several infected IoT devices to overwhelm a particular target node. Several recent attacks [12] have come to light which have exploited IoT devices to launch massive DDoS attacks. The second is a linking attack in which the attacker establishes a link between multiple transactions or data ledgers with the same PK to find

the real world ID of an anonymous user. This attack endangers users privacy.

DDOS attack: Our design has a hierarchical defence against this attack. The first level of defence can be attributed to the fact that it would be impossible for an attacker to directly install malware on smart home devices since these devices are not directly accessible. All transactions have to be checked by the miner. Let us for a moment assume that the attacker somehow still manages to infect the devices. The second level of defence comes from the fact that all outgoing traffic has to be authorized by the miner by examining the policy header. Since the requests that constitute the DDoS attack traffic would not be authorized, they would be blocked from exiting the home. The next two defence layers are specially designed and managed by the target of DDOS attack that can be any user in the overlay. These defense layers, that are granting permission by using CH key lists and changing the PK in the CH key lists, are discussed in our previous paper [6] and are not in the scope of this paper.

Linking attack: To protect against this attack, each device's data is shared and stored by a unique key. The miner creates unique ledger of data in the cloud storage for each device using a different PK. From the overlay point of view, the miner should use a unique key for each transaction.

B. Performance Evaluation

BC-based architecture incurs computational and packet overhead on the smart home devices and the miner for providing improved security and privacy. To evaluate these overheads, we simulated a smart home scenario in Cooja simulator [13]. To compare the overhead of the BC-based architecture, we simulated another scenario that handles transactions without encryption, hashing, and BC. We refer to this baseline method as the "base method". We used IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN) as the underlying communication protocol in our simulation, since it is well-suited to the resource constraints for a smart home setting. We simulated three z1 mote sensors (that mimick smart home devices) which send data directly to the home miner (also simulated as a z1 mote) every 10 seconds. Each simulation lasted for 3 minutes and the results presented are averaged over this duration. A cloud storage is directly connected to the miner for storing data and returning the block-number. It is worth noting that the overlay delay and processing is not considered in our simulation. To provide a comprehensive evaluation we simulated store and access transactions.For the store transaction we simulated two different and realistic traffic flow patterns:

- Periodic: In this setting, devices periodically send their data to the cloud storage. This is fairly typical for various current smart home products such as Nest thermostat.
- Query-based: Herein, the device sends data on-demand and in response to a query received from the miner. This flow is equivalent to storing data to the cloud by the home owner.

We evaluated the following metrics:

- Packet overhead: Refers to the length of transmitted packets.
- Time overhead: Refers to the processing time for each transaction in the miner and is measured from when a transaction is received in the miner until the appropriate response is sent to the requester.
- Energy consumption: Refers to the energy consumed by the miner for handling transactions. The miner is the highest energy consuming device in the smart home since it handles all transactions and performs lots of hashing and encryption. The energy consumption of other devices is limited to encryption for their own transactions.

The discussion on the evaluation is as follows:

Packet overhead: Table II illustrates the simulation results for packet overhead. The table content applies to both access and store transactions since both have the same packet size. Using encryption and hashing increases the packets payload size; however, considering the lower layer headers (i.e. 6LowPAN), the increase in the data payload has relatively small effect.

Time overhead: Figure 3 shows the results for the time overhead. The BC-based design consumes more time to process packets compared to the base method which can be attributed to the additional encryption and hashing operations. In the worst case for the query-based store transaction the additional overhead introduced by our method is 20ms, which is still small.

Energy consumption: Figure 4 outlines the energy consumption results. As is evident, the BC method increases the energy consumption by 0.07 (mj). The table at the bottom of Figure 4 outlines the energy consumption for the 3 core tasks performed by the miner, namely: CPU, transmission (Tx), and listening (Lx). The energy consumption by CPU increased roughly 0.002(mj) in our design due to encryption and hashing. Transmitting longer data packets doubled the transmission energy consumption of our method in compare to the base method. It should be noted that we have assumed a 100% radio duty cycle in our evaluations (i.e. the radio is always on). If the

TABLE I
SECURITY REQUIREMENT EVALUATION.

Requirement	Employed Safeguard
Confidentiality	Achieved using symmetric encryption.
Integrity	Hashing is employed to achieve integrity.
Availability	Achieved by limiting acceptable transactions by devices and the miner.
User control	Achieved by logging transactions in local BC.
Authorization	Achieved by using a policy header and shared keys.

TABLE II
EVALUATION OF THE PACKET OVERHEAD

Packet Flow	Base (Bytes)	BC-based (Bytes)
From devices to the miner	5	16
From the miner to the cloud	5	36
From the cloud to the miner	5	16

radio is switched off intermittently to conserve energy, then the relative listening overhead incurred by our method would be higher. However, even assuming a very aggressive duty cycle of 1%, the relative increase in listening energy would still only be about 60%.

In summary, the low overheads introduced by our BC-based method significantly outweigh given the significant security and privacy benefits on offer.

V. Related Works

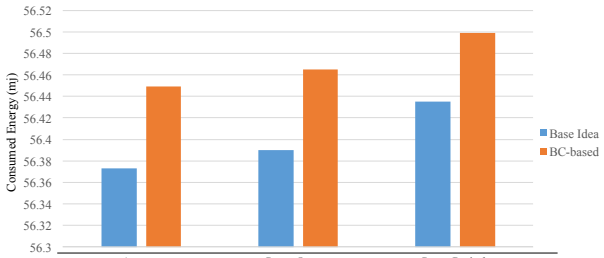
There exist different studies on security and privacy of IoT and smart home. Authors in [14] demonstrated that off-the-shelf IoT devices lack basic security safeguards by hacking into a variety of smart home device including a light bulb, switch and smoke alarm. Authors in [15] argued that the smart homes are vulnerable to attacks conducted by users' smartphones even if the home gateway controls the exchange of packets to and from the home.

Authors in [3] proposed a method with three modules to protect users' privacy in the smart home. The *data collector* module collects users' data from the smart home and sends them to *data receiver* module that stores data in two different datasets. The *result* module controls the user's access to data to protect the privacy. This method ensures that only the true user can access data. Besides, by using two datasets it is guaranteed that linking different data of a user to each other is impossible. However, the method does not provide privacy when the user needs to reveal his data to a service provider.

VI. Conclusion

IoT security is gaining a lot of attention these days from both academia and industry. Existing security solutions are not necessarily suited for IoT due to high energy consumption and processing overhead. We previously proposed a method that addresses these challenges by leveraging the Bitcoin BC, which is an immutable ledger of blocks. The idea was discussed using a smart home as a representative case-study. In this paper, we outlined the various core components of the smart home tier and discussed the various transactions and procedures associated with it. We also presented an all-inclusive analysis regarding its security and privacy. Our simulation results demonstrate that the overheads incurred by

Fig. 3. Evaluation of time overhead.



	Access			Store-Query			Store-Period		
	CPU	Tx	Lx	CPU	Tx	Lx	CPU	Tx	Lx
Base Idea	0.0103	0.0102	56.3489	0.0117	0.0357	56.3393	0.0121	0.0549	56.3645
BC-based	0.0127	0.0834	56.3496	0.0132	0.1106	56.3379	0.0133	0.1166	56.3657

Fig. 4. Evaluation of energy consumption in different traffic flows.

our method are low and manageable for low resource IoT devices. We argue that these overheads are worth their weight given the significant security and privacy benefits on offer.

To the best of our knowledge, this research is the first work that aims to optimize BC in the context of smart homes. In our future research, we will investigate the applications of our framework to other IoT domains.

References

[1] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," *Computer Networks*, vol. 76, pp. 146–164, 2015.

[2] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed internet of things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.

[3] A. Chakravorty, T. Wlodarczyk, and C. Rong, "Privacy preserving data analytics for smart homes," in *Security and Privacy Workshops (SPW), 2013 IEEE*. IEEE, 2013, pp. 23–27.

[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[5] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," *July 7th*, 2013.

[6] A. Dorri, S. S. Kanhere, and R. Jurdak, "Blockchain in internet of things: Challenges and solutions," *arXiv preprint arXiv:1608.05187*, 2016.

[7] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies*. Princeton University Pres, 2016.

[8] A. Bogdanov, M. Knežević, G. Leander, D. Toz, K. Varıcı, and I. Verbauwhede, *spongent: A Lightweight Hash Function*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 312–325.

[9] F.-S. sense, https://sense.f-secure.com/, [Online; accessed 19-November-2016].

[10] H. Delfs, H. Knebl, and H. Knebl, *Introduction to cryptography*. Springer, 2002, vol. 2.

[11] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014.

[12] wired, https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/, [Online; accessed 10-December-2016].

[13] Cooja, http://anrg.usc.edu/contiki/index.php/CoojaSimulator/, [Online; accessed 19-November-2016].

[14] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli, "An experimental study of security and privacy risks with emerging household appliances," in *Communications and Network Security (CNS), 2014 IEEE Conference on*. IEEE, 2014, pp. 79–84.

[15] V. Sivaraman, D. Chan, D. Earl, and R. Boreli, "Smart-phones attacking smart-homes," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 195–200.