International Journal of Internet of Things 2019, 8(1): 17-26
DOI: 10.5923/j.ijit.20190801.03

SDIoBoT: A Software-Defined Internet of Blockchains of Things Model

Navid Rajabi*, Jihad Qaddour

School of Information Technology, Illinois State University, Normal, IL, USA

Abstract Internet of things (IoT) has been attracting the technology world as an integral part for implementation of smart buildings, smart cities and actually, smart world (which is destination of the current trend with the aim of connecting everything to everything). However, some major problems like security issues, lack of sufficient processing power, etc. will prevent researchers and major tech companies to reach a consensus to finalize a highly-secure, efficient architecture for smart cities. In this paper, we've proposed a new model to tackle well-known IoT implementation challenges by using software-defined networking (SDN) architecture as an integral part for acting as the brain of the model and one of the significant components of the future 5G telecommunication networks. In addition to that, we've taken advantages of the Blockchain technology to enhance the security of IoT networks which will be an integral part of the future smart world paradigm. The security of the model is provided by leveraging Elliptic Curve Digital Signature Algorithm (ECDSA) that is being used in Bitcoin for integrity.

Keywords Internet of Things (IoT), Smart City, Smart World, Software-Defined Networking (SDN), Blockchain, ECDSA, Bitcoin

1. Introduction

After the appearance of the Internet, communication among people has changed a lot. There is another revolution in the technology realm, known as "Internet of Things (IoT)" that tries to create connections with one another, connections to the internet, and therefore, takes advantage of this schema for a transition from smart homes to a smart world! Internet of Things (IoT) has been proposed by Kevin Ashton in 1990 and people call him the "Father of Internet of Things" [1]. He described a new world in which each thing has its own digital identity, which allows computers to manage and control them. Although this concept was proposed more than one decade ago (1990s), the progress of this technology has not kept up with other companies, because of serious obstacles that this technology needs to overcome. Through these obstacles, we can exemplify lack of development infrastructure, low computational power, low storage capacity and then, lack of sufficient security. As we already know, security is one of the most challenging parts of any new technology. So, in such a vast environment that wants to connect billions of things and devices to the internet, security issues will become much more critical, since many of these devices and things are going to deal with human's lives directly and store their everyday data by using various sensors everywhere which could be classified as following:

- Wearable Technologies: Entertainment, Fitness, Smart Watches, Location & Tracking
- Buildings & Constructions: Access Control, Light Control, Temperature Control, Energy Optimization, Predictive Maintenance, Connected Appliances.
- Smart Cities: Residential Electronic-Meters, Smart Street Lights, Pipeline Leak Detection, Traffic Control, Surveillance Cameras, Centralized & Integrated System Control

Figure 1 shows the interest over time for the "Internet of things" in the past decade, measures by Google trend (which is the Internet's best search engine).

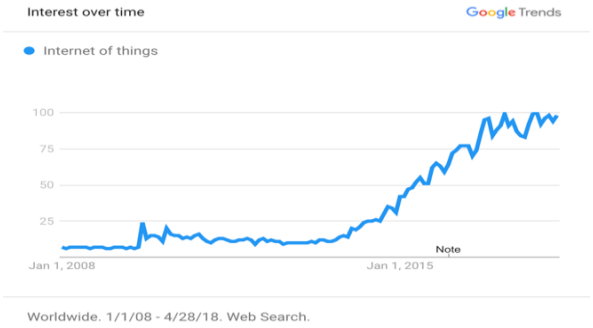

Figure 1. Internet of Things Interest (2008-2018)

* Corresponding author:
nrajabi@ilstu.edu (Navid Rajabi)
Published online at http://journal.sapub.org/ijit
Copyright © 2019 The Author(s). Published by Scientific & Academic Publishing
This work is licensed under the Creative Commons Attribution International License (CC BY). http://creativecommons.org/licenses/by/4.0/

Therefore, a secure architecture needs to be proposed to tackle potential critical issues and be compliant with CIA classical security model which stands for Confidentiality, Integrity, Availability.

To realize the significance of the security of IoT, we should look at the systems that are trying to meet the requirements of the CIA model. Every day, we are witness of security breaches and information leakages and numerous cyber-attacks like Distributed Denial of Service (DDoS) on powerful servers (with decent amount of computational resources for computing & storage) and crash them even in the situation that existing security mechanisms like Firewalls, Load Balancers, Host-based Intrusion Detection Systems (H-IDS), Network-based Intrusion Detection System (N-IDS) etc. are deployed at different parts of the Internet environment. This could be a serious alarm for smart city researchers who want to implement smart world, provide security for that and convince the people to use that technology which will deal with every single data of their everyday life. Since the above-mentioned protections are not sufficient for the future vast, borderless and fearful architecture of the Internet.

Figure 2 shows the interest over time for the "IoT Security" in the past decade, measures by Google trend.

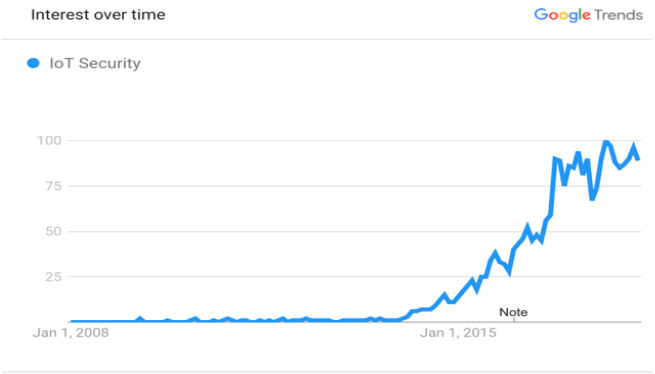

Figure 2. IoT Security Interest (2008-2018)

Figure 3 shows the interest over time for the "internet of things cyber security" in the past decade, measures by Google trend.

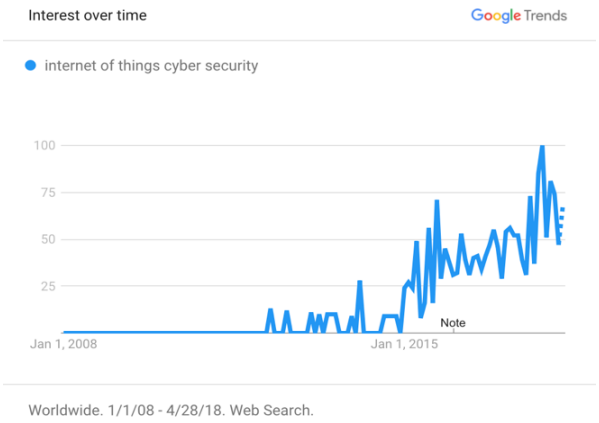

Figure 3. IoT Cybersecurity Interest (2008-2018)

In addition to the mentioned facts about the significance of the security for IoT, due to the demands for higher-speed networking and appearance of the 5G cellular communication networks, IoT is included inside the scope of 5G.

Generally speaking, 5G will be generated as a result combination of four major modern networking technologies called 4G cellular communication networks (LTE), Software-Defined Networking (SDN), Network Function Virtualization (NFV) and Internet of Things (IoT). Therefore, any robust model for the future IoT networks have to be compatible with other three networking technologies.

1.1. IoT Security Risks

Major security risks threatening IoT could be categorized as following:

- Physical-Layer Attacks
- Wireless Attacks
- Low Defensive Ability
- Excessive Reliance on Technology
- Low-Power Protocols
- Sensor's Energy
- Standardization
- Heterogeneous Devices/Heterogeneous Interfaces

1.2. Internet of Things (IoT) Architecture

Figure 4 depicts the stack of IoT communication layers in the way that blue boxes are layer names, yellow boxes are services, technologies and devices that are working inside each specific layer. The red boxes are showing the security services of each layer.

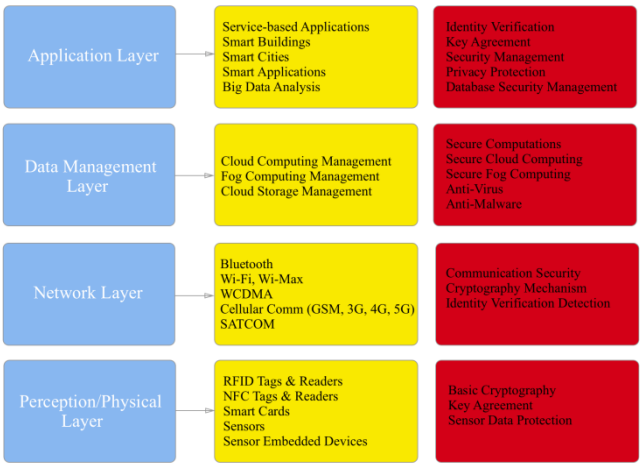

Figure 4. Internet of Things General Architecture Model

In terms of communication protocol, we are going to use the RFC 7252 Constrained Application Protocol (CoAP) for this model which is a RESTful based web transfer protocol to be used for IoT nodes with processing power constraints and make them able to interact with the server using GET, POST, PUT and DELETE requests [14]. In addition to that, according to RFC 7228, the CoAP has been created to be compatible with microcontrollers with as low as 10 KB of

RAM and 100 KB of storage, since most of the IoT nodes have to be cheap and inexpensive with lowest level of computing power and storage capacity [14].

2. Literature Review

2.1. Symmetric Cryptography Algorithms

TABLE 1. shows the already implemented symmetric cryptography algorithms (including information about them).

Table 1. Symmetric Cryptography Algorithms [4]

Algorithm	Data Size	Key Size	No of Rounds	Structure	Possible Attacks
AES	128 Bits	128/192/256	10/12/14	Feistel	Not any
DES	64 Bits	56	16	Feistel	Brute force
Triple DES	64 Bits	168	48	Feistel	Meet in middle
Blowfish	64 Bits	128-448	16	Feistel	Second order differential
IDEA	64 Bits	128	8	Substitution-Permutation	Related key
TEA	64 Bits	128	64	Feistel	Related key

2.2. Lightweight Symmetric Cryptography Algorithms

TABLE 2. shows the already implemented lightweight symmetric cryptography algorithms (including information about them).

Table 2. Lightweight Symmetric Cryptography [4]

Symmetric Algorithm	Code length	Structure	Number of rounds	Key Size	Block Size	Possible Attacks
AES	2606	SPN	10	128	128	Man-in-middle attack
Hight	5672	GFS	32	128	64	Saturation attack
TEA	1140	Feistel	32	128	64	Related Key Attack
PRESENT	936	SPN	32	80	64	Differential attack
RC5	Not foxed	ARX	20	16	32	Differential attack

2.3. Proprietary Cryptography Algorithms

Table 3. Proprietary/legacy Cryptography [5]

Name	Intended platform	Key	IS	IV	Att. time
A5/1	Cell phones	64	64	22	2^{24}
A5/2		64	81	22	2^{16}
CMEA †		64	16–48	–	2^{32}
ORYX		96	96	–	2^{16}
A5-GMR-1	Satellite phones	64	82	19	$2^{38.1}$
A5-GMR-2		64	68	22	2^{28}
DSC	Cordless phones	64	80	35	2^{34}
SecureMem.	Atmel chips	64	109	128	$2^{29.8}$
CryptoMem.		64	117	128	2^{50}
Hitag2	Car key/ immobilizer	48	48	64	2^{35}
Megamos		96	57	56	2^{48}
Keeloq †		64	32	–	$2^{44.5}$
DST40 †		40	40	–	2^{40}
iClass	Smart cards	64	40	–	2^{40}
Crypto-1		48	48	96	2^{32}
CSS	DVD players	40	42	–	2^{40}
Cryptomeria †		56	64	–	2^{48}
CSA-BC †	Digital televisions	64	64	–	2^{64}
CSA-SC		64	103	64	$2^{45.7}$
PC-1	Amazon Kindle	128	152	–	2^{31}
SecurID ‡	Secure token	64	64	–	2^{44}
E0	Bluetooth devices	128	128	–	2^{27}

TABLE 3. shows the already implemented proprietary/legacy lightweight symmetric cryptography algorithms (including information about them) [5].

2.4. Gap of the Existing Solutions

Figure 5 shows the unstable fluctuating status of the "Lightweight Cryptography" in the past 14 years.

Figure 5. Lightweight Cryptography interest from 2004 to 2018

To summarize what we've mentioned for the cryptography algorithms (symmetric, lightweight symmetric & proprietary/ legacy) we can say that each of the solutions could be effective under specific circumstances. These circumstances could be the computational power of that specific device, computational power of the other nodes in the network etc.

Proprietary cryptography algorithms are highly sensitive, because they have to pass National Institute of Science and Technology (NIST) standardization process. And even if the NIST approve proprietary algorithms, it is hard to trust this kind of algorithms, since they are compliant with the "Open Design" principle, since there is a secrecy in their design.

Symmetric cryptography algorithms are useful for a portion of IoT devices with higher level of computational power, but they don't provide enough confidentiality and integrity, since higher security could be achieved only with large number of rounds with large key sizes.

Asymmetric cryptography algorithms like Diffie-Hellman, ECC or ElGamal are not useful as well, since they are too heavy for major portions of IoT devices due to the use of both public key & private key in addition to the key exchange process (which requires high level of computing power). However, even by implementing an asymmetric cryptography algorithm with short key length, there is no guarantee to reach the highest level of security and performance, since there is a linear behavior associated with the key length and the achieved security level in the way that the longer key will provide higher security.

3. Overview of Blockchain and SDN

Therefore, we came up with a solution to combine SDN with the Blockchain technology. In this way, we can provide

confidentiality and integrity using hash function (instead of using heavy cryptography algorithms) and propose a model based on the SDN. By performing this integration, not only we can increase flexibility and performance in comparison to traditional networks, our model will also be compatible with the 5G networks.

3.1. Blockchain

Blockchain concept has come into the realm of technology by appearance of "Bitcoin: A Peer-to-Peer Electronic Cash System" paper introducing "Bitcoin" in 2008 [1]. In this paper, Satoshi Nakomto has been introduced the general architecture of this system and after attracting a large number of scientists and technology enthusiasts, he proposed the first version of Bitcoin at January 2009. Then after a while, Bitcoin network started.

Although the Bitcoin concept is the first word that comes to the mind after mentioning the Blockchain (and these two concepts are highly interdependent), the concept of Blockchain is not limited to the cryptocurrencies. This concept had a huge progress during the time and found many practical use cases in a broad range of industries like:

- Accounting
- Settlement
- Finance
- Cryptocurrencies
- Internet of Things (IoT)
- Electronic Government (Digital Voting etc.)
- Data Security & Integrity
- Digital Identity Management
- Smart Contracts

In fact, the existential philosophy of Blockchain is to eliminate exclusive ownership. In other word, no one could have ownership claim on whatever data that is existing on the Blockchain. Rather, this ownership has been fully distributed on the Blockchain and will be available to them after some specific procedures. We will dive into the details and those specific procedures later in this paper.

Before talking about the details of Blockchain technology, we've imported the result of Google Trend for "Blockchain for IoT" term in Figure. 6, which indicates a dramatic increase from 2015.

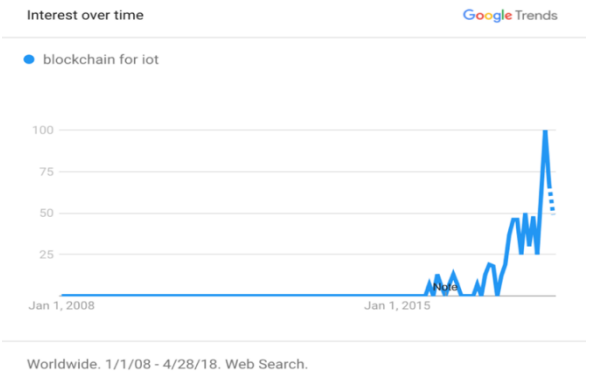

Figure 6. Blockchain for IoT trend from 2008 to 2018

Here, we will talk about the Blockchain technology in detail and answer the question that "How Blockchain can provide security?"

Basically, Blockchain can be defined as a distributed data storage (which is totally different to traditional databases from structure perspective) with the capability of providing an environment for secure transactions. In this way, data will be stored in a continuous sequence of blocks and the mechanism which keeps this sequence securely is called "Hash Pointing". A Hash Pointer is a data structure that consists of two parts. The first part is a pointer to where data is stored and the second part is a cryptographic hash of that data. So, by having a hash pointer, it is possible to retrieve the data and also, verify that data has not been altered. So, the Blockchain technology is relying on hash pointers for tamper-evident log among blocks:

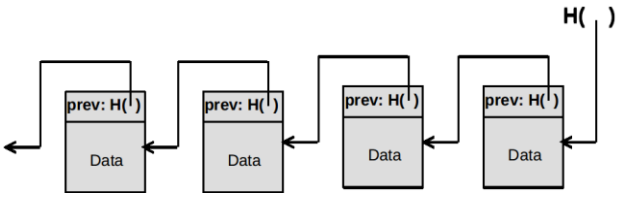

Figure 7. Hash Pointer Chain [9]

Therefore, if an adversary attempts to alter the data in any part of the chain, it will be detected very easily and so fast, because of the nature of the hash function. The hash function takes an input data and generates a fixed-length output as a hash result. So, if any part of the original data has been altered by anyone, the inconsistency can be easily recognized since the hash collision-free algorithm generates another fixed-length output which is not compliant with the original one. In this way, each block has two parts, header and data. So, the address of the previous block and also the hash value of that will be stored in the header of each block. Therefore, even if an adversary attempts to change all the hash pointers and data of the whole Blockchain, he won't be successful in this kind of attack, since the "Genesis block", the first block of the chain, is immutable. Then, the data integrity could be guaranteed in Blockchain by implementation of hash pointers. Another structure that has been implemented in Blockchain technology is called Merkle Tree, proposed by Ralph Merkle as a digital signature algorithm [2]. This storage structure is being implemented to be used in each block of the Blockchain to facilitate storing the transactions of that block. Merkle tree is an infinite binary tree of one-time signature which stores data in leaves and each internal node will contain a hash value of its children.

By using this structure, it is easy and fast to verify the integrity of each transaction of each block. It only needs to traverse through a binary tree from root to that leaf which is called "Proof of Membership". The time and space complexity would be as follows:

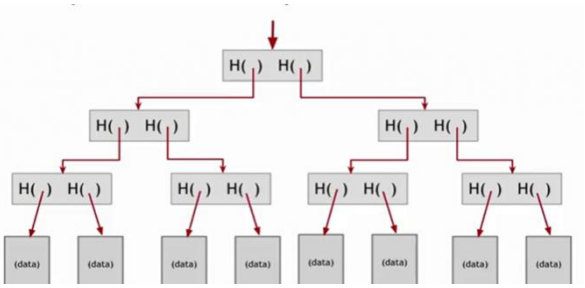

Figure 8. Merkle Tree using Hash Function [10]

	Average	Worst
Space	$O(n)$	$O(n)$
Search	$O(\log_2(n))$	$O(\log_k(n))$
Traversal	$^*O(n)$	$^*O(n)$
Insert	$O(\log_2(n))$	$O(\log_k(n))$
Delete	$O(\log_2(n))$	$O(\log_k(n))$

Figure 9. Merkle Tree Complexity Orders [11]

Therefore, each block of the Blockchain will have a Merkle tree to store all of its transactions securely, since it is storing the transaction data using a hash algorithm that is highly secure for data integrity because of irreversibility.

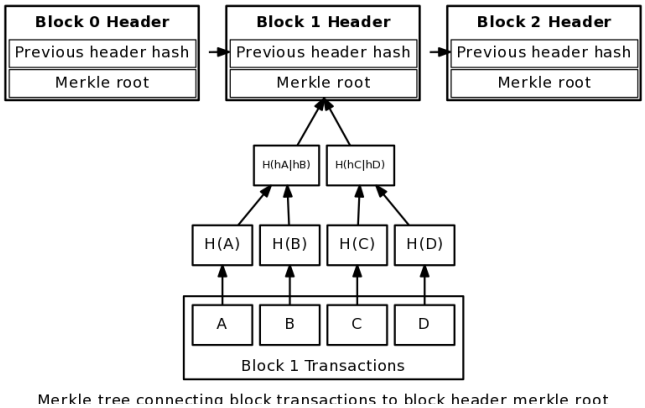

Figure 10. Merkle Tree [8]

3.2. Software-defined Networking (SDN) Architecture

SDN has emerged as a modern networking technology to enhance the flexibility of the traditional networks, accelerate the innovation in future networks, reduce costs of initialization of maintenance of the networks, decrease the dependency of proprietary networking devices etc. SDN general architecture consists of three major layers and two major APIs for proper, secure communication between these three layers. The middle layer called "Control Plane" known as "Network Brain" which includes the SDN controller, acting as an operating system which will be programmed by the top layer using "Northbound API" or "RESTful API". The top layer called "Application Plane" includes applications to manage & monitor the network for operations and security purposes. The bottom layer called "Data Plane" includes packet-forwarding devices (similar to switches and routers of traditional networks) and communicates with the network controller using "Southbound API" or "OpenFlow Protocol".

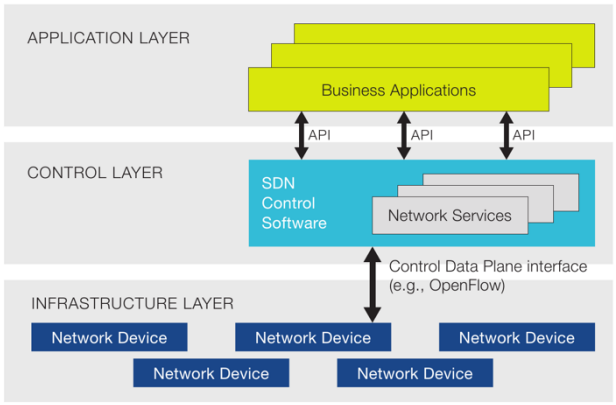

Figure 11. SDN Architecture [12]

The reason behind choosing SDN as the core part of our architecture is that SDN technology (which includes Open vSwitch, Northbound & Southbound APIs etc.) provides us the opportunity to implement our model in the SDN controller. Moreover, to define all the communications with highly flexible APIs like OpenFlow and RESTful APIs that we couldn't do by using traditional network technologies.

3.3. Fog vs. Cloud Computing

According to the differences between fog computing and cloud computing characteristics, we've decided to choose fog computing for our architecture, due to its great advantages and compatibility with our purpose like bringing the process location to the edge (to enhance the security by reducing the risk of data manipulation in the long-distance communications and minimize the delay). Advantages of that include low latency, capability of mobility function (which is one of the integral part of modern communication networks), and other positive points which makes fog computing the best fit for our architecture.

The Cloud and the Fog

	Key Use Cases	Areas of application	Software frameworks	Key requirements	Actors (end users and providers)
Cloud Computing at the Core	Large Batch Jobs	Deep data mining, search, financial algorithms	MR/Hadoop	Scalability, security, availability	Enterprises, individuals
	DC Bursting				
	Web Hosting	E-commerce	Web applications		
	Storage				
Fog Computing	Mobile Content Delivery	Video streaming, gaming	rendering, transcoding	Low latency	end subscribers, service providers
	Geo-distributed Sensor/Actuator Networks	Smart cities, environmental monitoring	Fusion/aggregation, MR	Geo-distribution, security, privacy	Utilities, federal/state agencies, car manufacturers
	Large Scale Distributed Controlled Systems	SCV, Smart Grid, Int. Transportation	Control loops	Partitioning/ quasi-autonomy, security	

Figure 12. Cloud Computing vs. Fog Computing [13]

4. Proposed Model

(Integration of Blockchain, IoT & SDN)

Considering all of the information mentioned-above, we have to propose a solution, which will solve three main issue of IoT:

- Low bandwidth communications
- Low processing/computing power
- Low memory capacity

Based on these major problems, the solution must use as low bandwidth as possible for communication while having the lowest possible overhead. In addition, there should be a substitute for cryptography protocols for implementation of confidentiality, integrity, and availability. For the last issue, there should be as low stuffs as possible for each node to store.

The main concept of our idea is to take advantage of distributed Blockchain technology to make IoT nodes secure and reliable. To be more specific, we are grouping all the IoT nodes in number of Blockchain based on their proximity and their processing power level to make it possible for nodes to take advantage of the maximum possible security level and solve the issue of scarifying the security just due to heterogeneity of IoT devices (which is one of the huge issues in 5th generation cellular networks and also smart cities). In this way, in each Local Area Network (LAN), which means Wireless Access Points (WAP) or 5G Small Cells etc. we are going to have several Communication Interfaces (CI). These CIs will be represented like different interfaces of that WAP or Small Cell and will have different behaviors, different processing, and memory capabilities. The reason for this kind of design is that we have to overcome one of the most important issues in IoT which is the security of the lightweight devices. In this way, we are going to put all digital physical assets (which includes all the sensors, machines, actuators etc.) in different Blockchain based-on their computational power. (Figure 13 shows the big picture of SDIoBoT architecture in large scale communications with other major components in smart city concept).

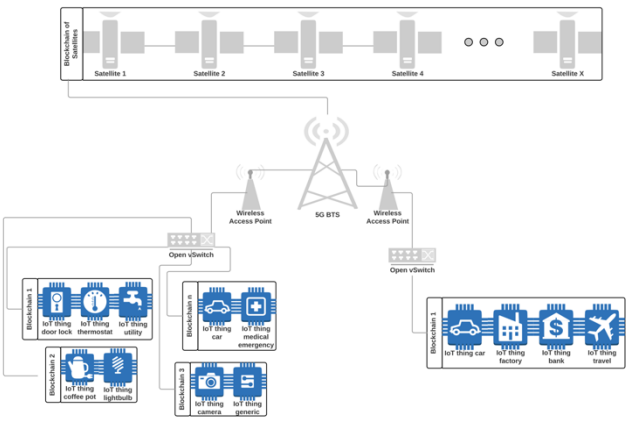

Figure 13. SDIoBoT Large Scale Architecture

To be more specific, we can demonstrate the big picture of our idea by depicting Figure 14, which is primarily indicating a software-defined network architecture in which each port of the Open vSwitch has been replaced with a Blockchain. In fact, these Blockchains are responsible of connecting proportional IoT devices to each other securely with the highest possible performance.

As shown in Figure 14, our model is more specifically dealing with the Data Plane of the SDN architecture and will be implemented there. Generally speaking, in this model, we are generating multiple Blockchains and connect each of them directly to each port of the Open vSwitch. To be more specific, these Blockchains will contain a dynamic number of IoT devices. So, we would have multiple Blockchains of IoT nodes and each IoT node will act as block in that specific Blockchain. The reason behind this architecture is that since the environment of IoT devices are completely heterogeneous, we are splitting them based on their computational power and overall capacity and put them in different Blockchains to enhance the performance of the system and reduce the latency.

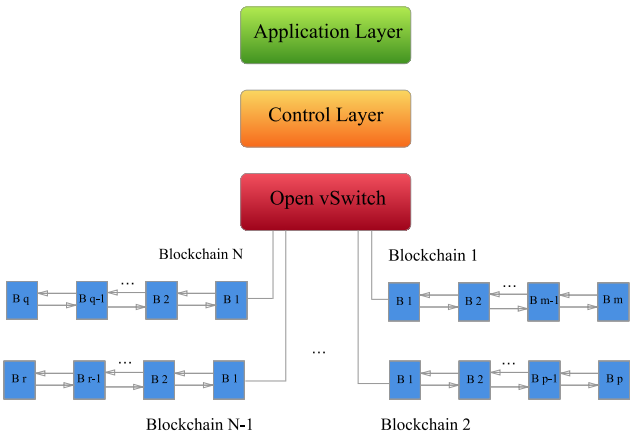

Figure 14. SDIoBoT Main Architecture

For example, a self-driving car and a light sensor cannot interact with other and shouldn't be in the same Blockchain. According to this architecture, each port of the Open vSwitch will be determined as a Genesis block of that Blockchain, since it will be connected to its following blocks and it is going to be the first block of that chain. Therefore, it has the Previous hash value of "Null" and will be determined as Genesis.

As we mentioned above, we have multiple blocks with different ranges of computational power. Therefore, they could be difference in major characteristics like hash function algorithm, data that will be stored on each block, smart contract, type of Blockchain platform (public or private), the number of Miners etc.

Another feature of our model is that it will rank the computational power of each block and save it on the Distributed Ledger of each block. This will be help the system to pick the more powerful blocks based on the ranking and assign them as the Miners of that Blockchain. This feature will make the system to use the most powerful blocks of each Blockchain to audit transactions and reduce the chance of error and failure during the auditing.

All of the mentioned features could be implemented as modules inside the SDN controller (which is the brain of the whole network) and then will be implemented by the Open vSwitch automatically (That's the logic behind integrating the model with SDN).

In this way, the implementation would be very transparent to the Application Layer, but all the data analysis and monitoring could be done inside that layer.

4.1. SDIoBoT's Sequence Diagram

Here we can see a general view of the communications which is going to be happened from the moment that IoT node is going to send a join request to the Open vSwitch (which includes the Gateway as well) and with SDN Controller for interacting with more high-level components of the network for faster and more accurate actions (Fig. 15).

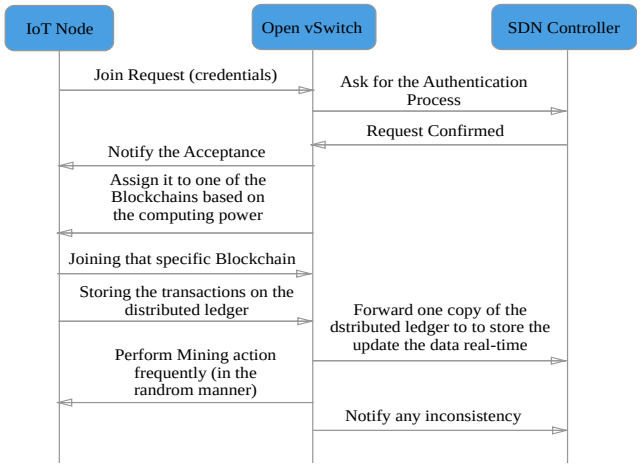

Figure 15. SDIoBoT Sequence Diagram

4.2. SDIoBoT's Algorithm Big Picture

1) Ports of the Open vSwitch are not connected to any IoT device and there is no Blockchain associated with any of the ports.
2) An IoT device asks the Open vSwitch to get connected.
3) The Open vSwitch will perform the authentication.
4) After the authentication, the Open vSwitch will compare the computational power of that IoT device with the predefined threshold to figure out which port should be assigned as Genesis for that Blockchain and generate the Blockchain.
5) The IoT device will be added to the generated Blockchain and all the required attributes will be assigned proportional to the measured computational power. (like Block ID, Timestamp, hash, previous, data, smart contract etc.)
6) For additional IoT devices, the process would be similar to 1 to 5, except the creation of the Blockchain for the situations that it is already existing.

In order to ensure authentication, non-repudiation and integrity of the communications between the blocks, a powerful digital signature algorithm must be used. In addition, that digital signature algorithm should be compatible with constrained enviroments and low-power devices in IoT and smart cities. Based on these assumptions, we've chosen Elliptic Curve Digital Signature Algorithm (ECDSA). This algorithm was invented in 1992 by Scott Vanstone with the purpose of getting similar level of security as DSA, but with using smaller keys [25]. ECDSA is also being used in Bitcoin and has three major phases, generating the key, generating the signature and verifying the signature. We are going to follow these steps and customize this algorithm for our model while preserving its structure unaltered [24]. In the first phase, IoT_Node_A generates a key using the following algorithm [24]:

1) $d = random\ integer\ from\ [1, n-1]\ interval$
2) $Q = d.g$
3) $IoT\_Node\_A_{PrivateKey} = d$
4) $IoT\_Node\_A_{PublicKey} = (E, g, n, Q)$

After the key generation phase, IoT_Node_A has to sign the message/transaction A_Message and make it ready for transmission using the following algorithm [24]:

5) $k = random\ integer\ from\ [1, n-1]\ interval$
6) $kg = (x_1, y_1)$
7) $r = x_1\ mod\ n \quad //(if\ r = 0\ return\ to\ step\ 5)$
8) $h = H(A\_Message) \quad //(H = SHA256)$
9) $s = k^{-1}(h + dr)\ mod\ n \quad //(if\ s = 0\ go\ to\ step\ 5)$
10) $IoT\_Node\_A_{\ Signature}\ (A\_Message) = (r, s)$

The IoT_Node_B can verify the signature of IoT_Node_A for A_Message using the following algorithm [24]:

11) $Getting\ IoT\_Node\_A_{PublicKey} = (E, g, n, Q)$
12) $Verify \quad 1 \le r \le n-1 \quad \&\& \quad 1 \le s \le n-1$
13) $w = s^{-1}\ mod\ n$
14) $h = H(A\_Message) \quad //(H = SHA256)$
15) $u1 = hw\ mod\ n$
16) $u2 = rw\ mod\ n$
17) $u1g + u2Q = (x_0, y_0)$
18) $v = x_0\ mod\ n$
19) $if\ v == r\ ,\ signature\ for\ A_{Message}\ is\ verified$

By using the above-mentioned algorithm, each IoT node can make sure that it is communicating with a legitimate entity and send/receive unaltered data. However, there is another point of failure that we have to talk about, the Distributed Ledger in Blockchain. Distributed Ledger is a manifestation of a real-time up-to-dated list of all successful transactions between all the nodes on the Blockchain. To be more specific, when a transaction is approved, Distributed Ledger will be updated and broadcasted to all the nodes on the Blockchain to make sure that everyone has an up-to-dated copy of that. However, we cannot implement this mechanism in the exact same way in our model, since it would have conflict with the fact that we are talking about an environment that has memory space constraint. Therefore,

we came up with hashing function solution to overcome this problem. In this way, we are going to use SHA256 hashing algorithm for storing the list of transactions on each node of the Blockchain, since it can accept an unlimited amount of text data and generate a 256 fixed-length irreversible string. Therefore, whenever a transaction is done, it will be concatenated with the previous hashed distributed ledger and will be hashed again. After that, a 256-character string will be broadcasted throughout the Blockchain to the others to keep the Distributed Ledger. However, before generating hash value of the transactions, the original data will be sent to the IoT gateway representative in that specific Blockchain to keep the actual data and giving it to the higher-level entities like SDN controller and application higher levels to store the original version of data. In this way, we can easily verify the original data with the hashed version of it by applying the same hash function (SHA256) to the data.

In this part, we have included an example of the process to clarify our proposed model. Assuming that we have a smart home with a number of IoT nodes with different power capabilities. So, we define two IoT chains for low-power and high-power devices called *IoT_Chain_Low* and *IoT_Chain_High* respectively. We add three IoT devices to the low-power chain:

```
Adding the IoT_Node_L_1 to the IoT_Chain_Low
Hash Value is : 5dc43b98b1632da4405473cc8a4bb8c9e15ffc329adbcfa65bf61166b993d5ab

Adding the IoT_Node_L_2 to the IoT_Chain_Low
Hash Value is : 7a329d27f0616876a83aa00929a41d6c1d2c93868b5b1d326de899237e0aa266

Adding the IoT_Node_L_3 to the IoT_Chain_Low
Hash Value is : 9c7fe18ab6a8b897f32030117f5f72d289795fec08bb694e969b1abcb6f7c7cc
```

Figure 16. Adding IoT nodes to the low-power chain

And three devices to the high-power chain:

```
Adding the IoT_Node_H_1 to the IoT_Chain_High
Hash Value is : b7f573ca4b3f0b89fe9ae2a3d66413d6fcfd64cbd9023ddc7fa230c06fbb9cfd

Adding the IoT_Node_H_2 to the IoT_Chain_High
Hash Value is : e2dfed52edce1ae38e51091872e8b1364349afbf0bbc2e6b73fceeb41dfb8116

Adding the IoT_Node_H_3 to the IoT_Chain_High
Hash Value is : e43ff8f8eebec8198aa2b2c778282205bbe9769a766cae668d58c286478b9893
```

Figure 17. Adding IoT nodes to the high-power chain

Then, we simulate some sample traffics throughout the chains and the results after the mining process would be like the following (similar steps can be taken for the high-power chain as well):


```
"index": 1,
"timestamp": "2019-02-02 02:01:02.029915",
"data": [
  "proof-of-work": 18,
  "Transmitted_Data": [
    {
      "FROM": "IoT_Node_L_1",
      "TO": "Gateway",
      "DATA": 101010111111
    },
    {
      "FROM": "IoT_Node_L_2",
      "TO": "Gateway",
      "DATA": 101110
    },
    {
      "FROM": "IoT_Node_L_3",
      "TO": "Gateway",
      "DATA": 101110101010101
    },
    {
      "From": "Server",
      "To": "q3nf394hjg-random-miner-address-34nf3i4nflkn3oi",
      "Data": 1
    }
  ]
},
"hash": "3c2812432a7a713e6732f73f1b58300a2eea0c66d528168d2d3a67bfab7bd2d3"
```


Figure 18. Secure communication between IoT nodes and the Gateway

The proposed architecture will provide us with three major advantages. The first one is the efficiency in proof-of-work and computational complexity. One of the most critical challenges in Blockchain applications is the high mining time due the length of the chain. By separating IoT devices into different chains with different power levels, we can increase the efficiency of the proof-of-work and mining process, which is going to be managed and controlled by the SDN Controller.

The second advantage is that after each packet transmission, the sender node will broadcast "FROM", "TO" and "DATA" of the transmitted packet to all the other nodes of that chain (in addition to the regular transmission). In this way, all the nodes on that chain have an immutable, up-to-dated distributed ledger of the transmissioned data (except the protocol overheads). Therefore, this feature will enhance data integrity throughout the network.

Finally, the third advantage is the identity management of the IoT devices on the network. Using this architecture, all the nodes are cryptographically connected to each other using a hash algorithm. The hash function has multiple inputs like timestamp (in which the node is added to the network), previous hash (the previous block), an index and an arbitrary data. In this way, the identity of the IoT devices will be under control and manageable. If any of the hash values will be changed, the miner of the chain can detect it. So, we can mitigate those types of attacks in which the attacker pretends to be a legitimate node.

4.3. Proof of the Hardware Capability

As we discussed earlier, we've chosen the CoAP protocol which has the ability of working with low-power microcontrollers (even with 10 KB of RAM and 100 KB of memory) [14]. In this way, we won't be worried about the processing power of our devices. On the other hand, Bitcoin hardware wallets like Opendime [18], Ledger Nano S [16] or Digital Bitbox [15] are in size of a small flash memory and are able to act as a block of a Blockchain system and interact with others. Because a block of a Blockchain only needs to sign a transaction and keep a fixed-length string of already happened transactions for the sake of integrity. In addition to that, we've assigned the Open vSwitch ports to do the "Mining" process (which needs more computing power) to prevent any failure or crash in IoT nodes.

5. Conclusions

In this paper, we have proposed and illustrated a new model to overcome low security level of IoT networks which is stem from low computing power. The big picture of this model is to generate multiple blockchains, in which each blockchain has a different limitation for computing power of its blocks. In this wasy, we are considering each IoT node as a block and connect each of the blocks to a IoT gateway and then connect the blockchain to the Open vSwitch. The

security approach we performed in this model is Elliptic Curve Digital Signature Algorithm (ECDSA) which is being used in Bitcoin to provide integrity and provides considerable security with lower key size and computational complexity than major existing algorithms. In addition to that, We've leveraged Software-Defined Networking (SDN) architecture to make the compatible with the 5G concept.

However, since there is no open source simulation environment that supports both IoT & Blockchain simultaneously, we had to present our model without simulation results. However, we are currently working on Cooja software of Contiki OS [22] (which is an open source network simulator operating system for IoT nodes) to improve it and implement Blockchain concept in it to provide the complete simulation report for the next paper. In addition, for another research paper we are going to use physical IoT devices with heterogeneous computing powers for an actual implementation.

6. Future Works

There are some significant points that we have to consider for the further improvements of this model as follows:

- Designing a dashboard including a graphical user interface (GUI) for the clients of the system to monitor whole the transactions of the Blockchain real-time. In addition to that, some alerts could be defined to pop up visually when something goes wrong. This web application can be also synchronized with the clients' smartphones in the next step (just in case for the situations that the client is not at home or wants to use his/her smartphone to see the status of the Blockchain)
- Optimizing the existing hash functions in order to invent a new customized hash function with lower timing and space complexities. This will help the method to become faster in mining processes, handover, Merkle tree operations which improves the total performance of the system.
- Implementation of AI techniques in order to increase the intelligence of the system for more accurate service management and anomaly detections.
- Increasing the security of transaction among the Blockchain with higher computing power by assigning more "Miners" randomly in order to prevent fake transactions and double-spending problems.
- Designing a powerful Open vSwitch with higher port capacities to have more Blockchain with more options for privacy levels (public, private & hybrid). In this way, customers are able to choose their preferred type of Blockchain proportional to their needs.
- Compression of the collected data before giving it to the hash function as the input, since it will help us to increase the performance of our IoT devices and decrease the delay.
- Big Data analysis in the application plane to recognize and classify suspicious activities for real-time anomaly detection (which makes the mining process easier and faster).

REFERENCES

[1] http://www.rfidjournal.com/articles/view?4986.

[2] Bhardwaj, Isha, Ajay Kumar, and Manu Bansal. "A review on lightweight cryptography algorithms for data security and authentication in IoTs." In *Signal Processing, Computing and Control (ISPCC), 2017 4th International Conference on*, pp. 504-509. IEEE, 2017.

[3] Laput, Gierad, Yang Zhang, and Chris Harrison. "Synthetic sensors: Towards general-purpose sensing." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 3986-3999. ACM, 2017.

[4] Luhach, Ashish Kr. "Analysis of lightweight cryptographic solutions for Internet of Things." *Indian Journal of Science and Technology* 9, no. 28 (2016).

[5] Biryukov, Alex, and Leo Paul Perrin. "State of the Art in Lightweight Symmetric Cryptography." (2017).

[6] Ross Anderson. A5 (Was: HACKING DIGITAL PHONES). uk.telecom (Usenet), https://groups.google.com/forum/?msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ#!msg/uk.telecom/TkdaytoeU4/Mroy719hdroJ, June 1994.

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

[8] https://yos.io/2016/05/19/merkle-trees-in-elixir/.

[9] https://blog.imaginea.com/from-bitcoin-to-blockchain-to-ethereum-part-2/.

[10] http://learningspot.altervista.org/hash-pointers-and-data-structures/.

[11] https://medium.com/@skj48817/merkle-trees-introduction-to-blockchain-c80c0247046.

[12] https://www.sdxcentral.com/sdn/definitions/inside-sdn-architecture/.

[13] https://www.slideshare.net/JiangZhu/web-perf2013.

[14] http://coap.technology/.

[15] https://bitcoin.org/en/wallets/hardware/digitalbitbox/.

[16] https://bitcoin.org/en/wallets/hardware/ledgernanos/.

[17] https://myhardwarewallet.co.uk/shop/digital-bitbox-hardware-wallet/.

[18] https://opendime.com/.

[19] https://myhardwarewallet.co.uk/shop/opendime-uk/.

[20] https://myhardwarewallet.co.uk/shop/ledger-nano-s/.

[21] https://myhardwarewallet.co.uk/shop/digital-bitbox-hardware-wallet/.

[22] http://www.contiki-os.org/index.html.

[23] Abidi, Abdessalem, Belgacem Bouallegue, and Fatma Kahri. "Implementation of elliptic curve digital signature algorithm (ECDSA)." In *Computer & Information Technology (GSCIT), 2014 Global Summit on*, pp. 1-6. IEEE, 2014.

[24] Sarath, Greeshma, Devesh C. Jinwala, and Sankita Patel. "A Survey on Elliptic Curve Digital Signature Algorithm and Its Variants." In *Second International Conference on Computational Science and Engineering (CSE-2014) Dubai, UAE*, pp. 121-136. 2014.

[25] Malvik, Arnt Gunnar, and Bendik Witzoee. "Elliptic Curve Digital Signature Algorithm and its Applications in Bitcoin." (2015).