VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Pham Huy Chinh

OVERVIEW OF IOT SECURITY CHALLENGES AUTHENTICATION, ENCRYPTION AND BLOCKCHAIN SOLUTION

Bachelor's Thesis

Information Technology and Communication

2019

ACKNOWLEDGEMENTS

Firstly, I wish to express my sincere gratitude to my supervisor Dr Ghodrat Moghadampour for guiding me through all the processes of the thesis, and Mr Jukka Matila, Senior Lecturer, for providing me the ideas for the topic of my thesis. Had there not been their help, I would never have completed this paper.

Secondly, I am also grateful to my family for supporting me during a hard time when I desperately looking for ideas for the thesis. Special thanks to my father, who continuously checked on my progress with the thesis from the beginning to the end and helped me to build the very first bricks of the work.

Finally, I would like to send my thanks to all my friends in Vaasa for providing precious information when I needed it. I am extremely thankful to my former house-mate Minh Nguyen for continuously sharing the thesis progress and his experience with me.

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Information Technology

ABSTRACT

Author	Pham Huy Chinh
Title	Overview of IoT Security Challenges, Authentication, Encryption and Blockchain Solution
Year	2019
Language	English
Pages	46
Name of Supervisor	Ghodrat Moghadampour

The modern world is experiencing a new industrial revolution, known as industry 4.0, where machines, storage systems and other production facilities are capable of operating, exchanging information and triggering actions autonomously and independently. The growth of IoT is playing a key role in industry 4.0. However, the rapid increase in the number of IoT devices also raises serious concern about its effect on cybersecurity and privacy as these devices are typically limited in processing power, storage and network capacity, and therefore vulnerable to attacks.

This thesis paper surveys and presents major challenges for IoT security and available methods to protect the IoT system from being hacked or compromised. The security risks were reviewed, existing attacks and threats were categorised based on what layer in the IoT architecture they may affect and their motives. For the solution, the best and widely accepted methods for authenticating and encrypting data were presented. A new approach in IoT security with emerging technology, blockchain, is discussed.

There is not any perfect solution to solve all security issues. Therefore, the thesis explains the pros and cons of each method that IoT developers need to put in consideration before choosing the right method for their own system. However, recommendations are also given based on the research results.

Keywords	Internet of Things, cyber security, authentication, encryption, blockchain

CONTENTS

TIIVISTELMÄ
ABSTRACT
1 INTRODUCTION 3
1.1 Definition of IoT 3
1.2 IoT Security Concerns 4
1.3 Objective 5
1.4 Methodology 5
1.5 Thesis Structure 6
2 SECURITY RISKS IN IOT SYSTEM 7
2.1 IoT Security Goals 7
2.2 IoT High-level Architecture and Challenges 9
2.2.1 Sensing/Perception Layer 10
2.2.2 Network Layer 13
2.2.3 Service/Support Layer 15
2.2.4 Interface/Application Layer 17
2.3 Security Threats 17
3 SECURITY SOLUTION 21
3.1 Authentication and Encryption 21
3.1.1 Symmetric and Asymmetric Encryption 21
3.1.2 Cryptographic Hash Function 23
3.1.3 Digital Signature 24
3.1.4 SSL/TLS 25
3.1.5 Lightweight cryptography 28
3.2 Blockchain Integration 31
3.2.1 Blockchain Background 31
3.2.2 Benefits 35
3.2.3 Challenges 36
3.2.4 Proposals 37
4 CONCLUSION 40
REFERENCES 42

LIST OF FIGURES

Figure 1. Security requirements in IoT 8
Figure 2. Symmetric-key encryption..... 22
Figure 3. Asymmetric-key encryption 23
Figure 4. Generation of digital signature 25
Figure 5. Digital signature verification 25
Figure 6. SSL/TLS standard handshake..... 27
Figure 7. TLS 1.3 handshake 27
Figure 8. Example of blockchain 34
Figure 9. Where the blockchain makes sense 38
Figure 10. Hybrid-IoT high-level architecture..... 39

LIST OF TABLES

Table 1. Threat types and their members 19
Table 2. System threats 20
Table 3. Lightweight cryptographic Algorithms and AES method comparison.. 29
Table 4. Hardware performance of CLEFIA compared with AES 30

ABBREVIATIONS

API	Application Programming Interface
BFT	Byzantine Fault Tolerance
CA	Certificate Authority
CIA	Confidentiality, Integrity, Availability
CRC	Cyclic Redundancy Check
DDOS	Distributed Denial-Of-Service
DNS	Domain Name System
IC	Integrated Circuit
IoT	Internet of Things
LWC	Lightweight Cryptography
MITM	Man-in-the-middle
P2P	Peer-to-peer
PKI	Public Key Infrastructure
PoS	Proof of Stake
PoW	Proof of Work
QR	Quick Response
RFID	Radio Frequency Identification
WSN	Wireless Sensor Network

1 INTRODUCTION

1.1 Definition of IoT

The phrase “Internet of Things” (IoT) was first mentioned by Kevin Ashton, a co-founder of the Auto-ID Center at Massachusetts Institute of Technology (MIT), in a presentation he made to Procter & Gamble (P&G) in 1999. IoT can be considered as the latest evolution of the Internet caused by the invention of radio-frequency identification (RFID), quick response code (QR-code), the rising number of sensors and actuators, the increase in device processing power and storage capacity, cheaper but faster broadband Internet connectivity. IoT has been adding two components to computers: sensors and networking. Sensors allow the computers to understand the world by itself. Networking helps computers to connect to sensors that are distributed all over the place and gather different kinds of information from them.

Since IoT can relate to multiple technologies, such as machine learning, sensory, communication, networking, real-time analytics, the exact definition is still subjective to different perspectives taken. Different researchers proposed different definition of IoT; however, they all tried to describe a vision where objects become part of the Internet: where each object has its identification and can access to the network, its location and status can be trackable, where services and intelligence are added to this expanded Internet, fading away the barrier between digital and physical world, impacting our everyday life /1/. The Institute of Electrical and Electronics Engineers (IEEE) listed standard definitions for IoT provided by some organisations /2/:

i. By IEEE: “A network of items – each embedded with sensors – which are connected to the internet.”
ii. By ITU: “A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on

existing and evolving interoperable information and communication technologies."

1.2 IoT Security Concerns

IoT has brought many benefits to businesses; some of these are: lowers the cost and increased productivity, reduction of human efforts, decision analytics and efficient resource utilisation. Ninety-two per cent of companies surveyed by Digicert said that IoT would be necessary to their business in 2020 [3]. Many researchers have reported a significant growth in the number of connected devices these years. It is estimated that over thirty billion devices will be connected to the internet, two-third of which will be related to the IoT /4/.

This rapid evolution of the traditional internet into IoT has opened the door for new business opportunities, allowing enterprises to benefit from new revenue streams developed by advanced business models and services and offers IoT users as customers of those services a "smart life". However, IoT also creates new challenges since it makes the society vulnerable to newer forms of threats and attacks. Symantec reported an average number of attacks against its IoT system at about fifty-two hundred per month, which yields up to nearly sixty thousand attacks only to a single company a year. In 2016, a wide-known cyberattack, namely distributed denial-of-service (DDOS), targeting systems operated by Dyn, a Domain Name System (DNS) provider, disrupted major internet platforms and services in Europe and North America. The cybercriminals that perform DDOS on Dyn infected millions of IoT devices and recruited them into the Mirai botnet, which then flooded Dyn's server with a large number of DNS lookup requests preventing traffic from reaching its actual customers. The Mirai DDOS worm was still the third most common IoT threat, which made up 15.9% of the attacks in 2018. This is the reason why eighty-two per cent of companies in the Digicert survey reported that security is their top concerns for IoT. Securing an IoT system is, however, still remained to be a difficult

challenge when less than half, forty-eight per cent, of businesses surveyed by Gemalto cannot detect whether their IoT devices have a breach, although spending on security grew by two per cent to thirteen per cent in 2018 from eleven per cent in the previous year, and almost all companies see a robust approach to IoT security as a critical competitive differentiator. At the same time, the strong growth of IoT concerns not only companies but also private users when seventy per cent of respondents to ARM's survey said that they wanted to see companies make a significant effort to improve data security and privacy in the future. Therefore, much more resources and efforts will likely to be spent on IoT security researches in the next few years, especially when 5G technology will be widespread, which is considered to be a condition for IoT to thrive even stronger. /3, 5, 6, 7/

1.3 Objective

This thesis aims to give an overview of IoT security including what challenges an IoT system is facing and available solutions to overcome the challenges. IoT security is extensive so that this paper only focuses on some most critical or innovative solutions.

1.4 Methodology

This thesis is based on literature analysis. The information presented in the thesis is obtained from various types of documents, such as books, journals articles, scientific report and surveys conducted by companies operating in cybersecurity field or related fields. Different opinions concerning IoT security, encryption and authentication method and blockchain are compared and selected before being presented on the paper to create a point of view.

1.5 Thesis Structure

The paper is divided into two main part: challenge and solution. The challenges are addressed first in Chapter 2. This chapter explains the goals of IoT security, presents what issues are threatening IoT security and how they affect these goals. The following chapter provides an overview of authentication and encryption in IoT by explaining and analysing the most vital solutions in this field, and analysing the blockchain approach solution.

2 SECURITY RISKS IN IOT SYSTEM

2.1 IoT Security Goals

IoT is also a form of the Internet; therefore, in general, security for IoT is similar to security for other elements of cyberspace. The primary security goals are to protect the confidentiality of data, preserve the integrity of data and promote the availability of data for authorised uses. These main goals form a well-known security model called the CIA triad, the basis of all security programs. A breach in any of these areas could cause severe problems to the system so they must be considered in any IoT project. However, the internet is an evolving entity, IoT is expanding in scale and related to more and more newer technologies, the strategies with the CIA Triad for achieving security will not be sufficient. Three more aspects are considered to be added to the CIA Triad to complete the security provision for IoT: Non-repudiation, authenticity and authorisation. The main security requirements are now addressed in six aspects, shown in Figure 1 /8, p. 30/ :

- Confidentiality: the ability to provide confidence to the users about the privacy of the sensitive information by using different mechanisms so that only permitted users have the right to access and its disclosure to the unauthorised party is prevented. Some mechanisms to achieve data confidentiality are data encryption, two-step verification, biometric verification, access control lists, Unix file permissions.
- Integrity: data is protected from being tampered by unauthorised entities and remains consistent, accurate and trustworthy over its entire life cycle. Conventional methods to ensure data integrity are Checksum and Cyclic Redundancy Check (CRC), version control.
- Availability: data or services of the system is ensured to be accessed immediately by the authorised party at all time. Availability is assured when authentication mechanisms, access channels and systems all work correctly

for the information they protect. In case of a system failure or various system conflictions, redundancy and failover backup should be implemented to provide a duplication of the system components.

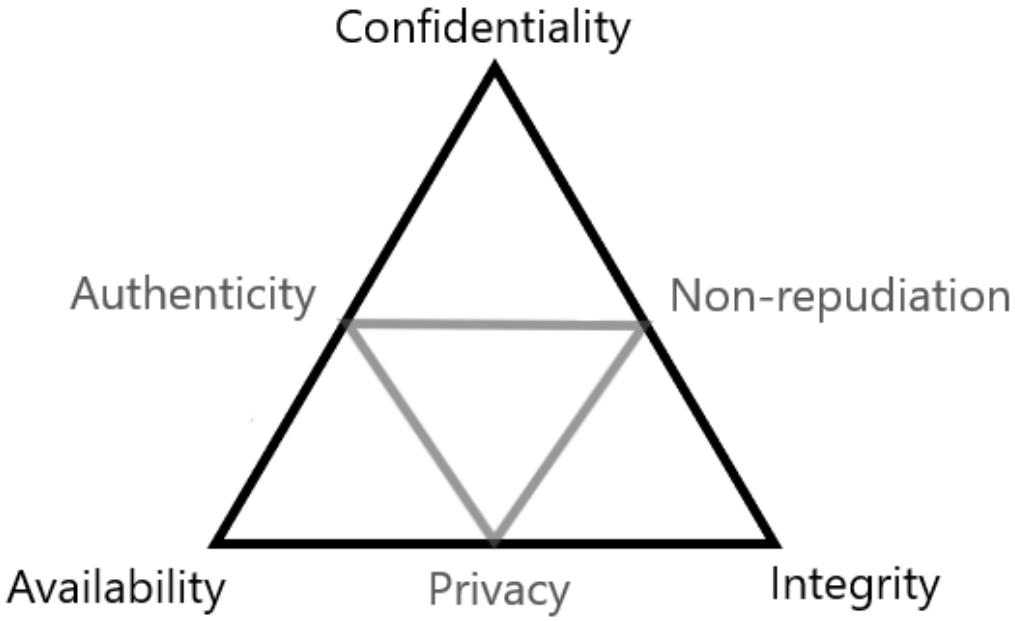

Figure 1. Security requirements in IoT

- Authenticity: Devices need to reliably identify themselves when they communicate with each other to prove that data, transactions, documents or other entities in the communication are genuine.
- Non-repudiation or traceability: the ability to uniquely trace actions to its causing entity, this serves as undeniable proof or the origin and validity of all data that is transmitted. Non-repudiation cannot be achieved without identities being verified, which means authentication needs to be performed first. However, having authenticity does not guarantee non-repudiation /9/. The digital signature or digital certificate method can provide non-repudiation for the date and origin of data.

- Authorisation or privacy: entities in or related to a system only do what they are authorised to do to protect the privacy of the users and prevent sensitive data from being collected by other parties, which are not the owners of the data, arbitrarily.

2.2 IoT High-level Architecture and Challenges

No well-defined IoT architecture has been established, yet. However, researchers have proposed different architectures for IoT, in which a three-layer high-level architecture is commonly accepted. The three layers are Application layer, Network layer and Perception layer. Later on, due to new technologies, such as cloud computing and Fog computing, some researchers believe that there should be one more layer, Support layer, in between network layer and perception layer /10/.

- The primary purpose of the Perception layer is to perceive the physical properties of things in the real world that are part of the IoT using sensors technology, such as RFID, Global positioning system (GPS), near-field communication (NFC).
- The responsibility of the Network layer is to connect smart things, devices and servers, and to transmit data obtained from the perception layer to any particular data processing system via the Internet, Mobile networks or any other reliable network technologies.
- The Application layer uses data processed in the previous layers to deliver specific services to the users. This layer consists of various practical applications of IoT in different kinds of industries, such as Smart Home, Smart Transportation, Smart city.

Although the three-layer architecture is accepted in defining the main idea of the IoT, many researchers found that it was not sufficient to describe all features and connotation of the IoT and suggested different architecture models, such as 5-layer

architecture /11/, 7-layer architecture /12/. However, this paper only focuses on security in the IoT system, the most basic model, 3-layer architecture with an extra support layer, is used to discuss further since it can cover most security issues in this paper.

2.2.1 Sensing/Perception Layer

The Perception layer is the lowest level of IoT construction consisting of intelligent tags and sensors network that can sense the surrounding environment and exchange data among devices automatically. Physical security of sensing devices and securing the collected information are two major security issues in the perception layer.

Physical security is vital since IoT devices can be placed anywhere in the environment. IoT devices need to be designed with anti-tampering and safety function so that it is challenging to extract sensitive information, such as personal data, credentials or cryptographic keys. Attacks targeting physical devices in the IoT system are divided into two groups: non-invasive and invasive attacks. To perform a non-invasive attack, the attackers need to be close enough to the target devices to examine the electrical characteristics; this allows the attackers to gather data flowing inside the devices or change the behavior of the devices without harming the devices physically during the attack. Non-invasive attacks do not require any preparations of the device under test and often scale well since the necessary equipment can be reproduced or updated at low cost. On the other hand, invasive attacks require removing the targeted chip's package to expose its surface so that it can be physically manipulated. Invasive attacks normally require the attackers to be well-equipped and knowledgeable to succeed. As the devices are getting more sophisticated, invasive attacks are becoming demanding and expensive. Therefore, the semi-invasive attack comes as a cheaper but repeatable alternative solution, mainly to learn the device functionality and test its circuit. Semi-invasive attacks still require depackaging the chip but do not make any contact with the internal chip layers /13, 14/. Some of the physical security threats are:

- Side-channel analysis is a passive non-invasive attack, allows the attackers to extract sensitive information from an Integrated Circuit (IC) by monitoring its electromagnetic radiation and analogue characteristics of supply and interface connection during normal operation.
- Tamper attack also known as microprobing, is an invasive attack where the attackers physically tamper the IC by using microprobes to gather information flowing through electric wires or try to override the current state of the IC. Microprobing and other invasive attacks require a special piece of equipment known as Microprobing Station that consists of necessary components for analysing the IC and making electrical contact to on-chip bus lines without damaging them.
- Fault injection attacks cause faulty behaviour in the system to take advantages of these faults to compromise the security. There are various ways to induce faults, for example, introducing glitches in the power supply, clock network or the reset network of an IC, changing the operating conditions of an IC by tampering the level of power supply or change the clock frequency, changing the operational environment, such as temperature. Fault injection is one type of active non-invasive attack, which involves actual interaction with the targeted devices rather than only observes IC signal and electromagnetic emission.

To secure the information collection, IoT devices must be able to prove their identity to maintain authenticity, sign and encrypt their data to maintain authenticity and also limit locally stored data to protect privacy. Perception layer collects and exchange data using different sensor technologies, such as RFID, which is mainly used as RFID tags for automated exchange of information without any manual involvement. However, RFID tags are exposed to many kinds of threats from outside due to the incorrect security status of the RFID technology. Most common types of threats are discussed below:

- Unauthorised access to the Tags: RFID systems that lack proper authentication mechanism put their tags in danger of being accessed by someone without authorisation. The attackers can not only read the data, trace the tags but also disable the tags, making them misbehave under the scan of a tag reader.
- Tag cloning: when the attackers successfully capture the identification information of the tags, replication of the tags is made possible in a way that the reader cannot distinguish between the original and the compromised tag.
- Eavesdropping: RFID readers always send radio signal requests to the tags to send back their identity information. This wireless characteristic of the RFID can be smooth for attackers to sniff out the confidential information, such as a password or any other data flowing from tag-to-reader or reader-to-tag. A reader needs to be at most 50cm from the tags so that it can power them. However, in the tag-to-reader eavesdropping attack, the adversarial reader can perform the attack at a greater range provided that the target tag has been powered. The reader-to-tag eavesdropping range can be much longer at a distance of kilometres away, depending on how strong the signal emitted from the reader is /15/.
- Replaying attacks: In this attack, the adversary intercepts the communication signal between the reader and the tag. The signal is recorded and then replayed when the reader sends a query later, thus faking the availability of the tag. For example, the adversary may broadcast an exact replay of the signal sent from a valid access card to the reader to get the unauthorised access to a restricted area. This attack can be mitigated by changing the message every time the tag responses to the reader based on timestamps, one-time password or cryptography using incremental sequence numbers.
- Jamming: This is one type of Denial of Service (DoS) attack that occupies the communication channel between the nodes, and stops them from communicating with each other. The transmission of the radio signal is used to interfere with radio frequencies used by the sensor network.

2.2.2 Network Layer

In the IoT system, the Network layer acts as a communication channel to transfer data collected in the perception layer to other connected devices. The network layer is implemented by using different wired or wireless communication technologies, such as Wi-Fi, Bluetooth, Zigbee, Lora, Z-Wave, cellular network, to allow data flow between devices in the same network. Security in the network layer mainly focuses on authentication/authorisation, routing security and data privacy. Proper authentication and point-to-point encryption minimise the chance of illegal access to the sensor nodes. After authentication is verified, routing protocols are used to ensure the privacy of data exchange between the sensor nodes and the processing systems. Finally, the data received needs to be checked to make sure it matches with the original data thus protects the integrity requirement /16/. Some security issues of the Network layer are:

- Sybil attack: In this kind of attack, the adversary replicates a single node and then present it with multiple identities to other nodes. Sybil attack may compromise a considerable part of the system, which results in false information about the redundancy. Generally, Sybil attacks are classified into two types: direct and indirect attack. In a direct attack, Sybil nodes communicate directly with honest nodes and try to influence their decisions. Indirect Sybil attacks use a middle node, which is already under the malicious influence of Sybil nodes, to affect the normal operation of other honest nodes instead of performing direct communications between Sybil nodes and targeted nodes. By using a large number of Sybil nodes, a Sybil attack can manipulate the reputation of a system; for instance, changing the result of a voting system, disseminate spam, advertisements and malware and violate the user's privacy on the online social network /17/.
- DoS attack: The main purpose of a DoS attack is to make the targeted system inaccessible to its intended users. There are many ways to perform a DoS attack; the attacker can either flood the system with traffic or trigger a crash or prevent the data from reaching its destination. The DDOS attack,

as introduced earlier, is a type of DoS attack; it exploits the rising number of unsecured IoT devices to flood its target system with useless traffic to exhaust its resources. The distributed characteristic of DDOS gives the attacker multiple advantages compared to other DoS type: attack location is difficult to identify due to the random distribution of IoT devices, finding true attacking party becomes challenging as they are disguised behind many systems, the attack can scale up to greater number of machines to execute a seriously disruptive attack. Due to this unique characteristic, DDOS can be considered the most dangerous DoS attack, and it is also causing high concern to organisations that may be the target of such attack.

- Man-in-the-middle (MITM) attack: MITM is active eavesdropping, in which the attacker, a third party, secretly monitor and possibly control the private communications between the two parties. The attacker can even fake the identity of the victim and regularly communicate to gain more information. For example, there are two users; user A wants to send user B an invoice, and he wishes to communicate through a "safe" line by encrypting his message. Firstly, user A sends "Hello" message and ask user B for her public key. However, the MITM attacker has successfully interrupted the connection and redirected all message from A to his computer. When the attacker receives user A's request, he forwards it to user B and follows the reaction of B. If B sends back her public key, the attacker keeps it and sends A his own public key. Then, user A believes that he has received B's key and use it to encrypt the invoice. At this point, the attacker has full control of the communication between A and B as he has all the keys he needs to encrypt and decrypt the messages.
- Sinkhole attack: The attacker compromises a node and makes it look attractive to the nearby nodes, for example, by claiming and displaying to have the shortest possible path. Therefore the data flow from any particular node is diverted towards the compromised node resulting in packets drop.
- Wormhole attack: By using a low-latency link, the adversarial node build a wormhole tunnel between itself and one or more other nodes faking a route

that is shorter than the original one within the network. This can confuse routing mechanisms which rely on the knowledge about the distance between nodes.

- Selective forwarding attack: A compromised node refuses to forward some of the packets in its buffer, such as control information or data packets to cut off the packets' propagation. This attack may not prevent the data from reaching the destination but instead introducing a delay. Selective forwarding attacks have many forms. The compromised nodes can either selectively drop the packet coming from a specific node or refuse to forward every packet. The attacker can also drop packets randomly or delay the packets passing through them his nodes. Moreover, messages can also be forwarded to the wrong path, which creates false routing information in the network /18/.
- Hello flood attack: adversarial node broadcast Hello packets with very high power to a large number of nodes in a large area of the network; these nodes are convinced the attacker node is their neighbour and waste their resource on responding to the hello message leaving the network in the state of confusion /19/.
- *Malicious code injection:* the attacker compromises a node to inject malicious code into the system. Depending on the intention of the attacker, the network may be shut down or controlled.

2.2.3 Service/Support Layer

The Service layer consists of information processing systems which can act automatically be based on the results of processed data and links the system with the database which provides storage capabilities to the collected data /16/. This layer offers IoT a cost-effective platform where the hardware and software platforms can be reused. The key technologies in the Service layer are Cloud Computing and Big

Data Processing. By providing more significant space to store the significantly increased amount of data generated by IoT devices, and providing an extreme processing power to analyse the data and make meaning from it, Cloud Computing helps to increase scalability and performance of many IoT system and reduce the need to buy expensive hardware with more storage and processing power. The Cloud is playing an essential role in the development of IoT; therefore, securing the Cloud is also a major concern in this layer.

- Denial of Service: As discussed earlier, DoS attack can be performed by overwhelming or flooding a targeted machine with requests until it runs out of resources to process.
- Insecure Application programming interfaces (API): APIs have been playing a critical role in the modern technology world as they underpin almost every daily activity, from banking to shopping to controlling indoor heating system. In IoT development, the cloud is the main processing power and data storage in many IoT system; IoT applications that need to access this stored data must communicate with cloud application via APIs. Exploiting a cloud API may give the adversary the chance to cause significant damage to the system by deleting the existing data or changing the right to access the related services. API can be breached by submitting unexpected parameter to exploit the application weaknesses; an SQL injection attack is one example of an attack that uses this method. A MITM attack is also another way to intercept the traffic between an API and an application/user, especially those APIs that are not using strong encryption method such as SSL/TLS.
- Malicious insiders: This kind of attack is performed by an authorised user by accessing the information of other users and tampering the data for personal or third party's benefit. Unauthorised access through misuse of employee credentials and improper access controls is one of the biggest threat to cloud security in 2019 /20/.

2.2.4 Interface/Application Layer

The Interface layer is the topmost layer, which provides the services that customers requests consisting of business logic, formulas and user interfaces to end-user. It involves a variety of application of IoT in different kinds of industries, such as Smart Home, Smart transportation, Smart City. Due to the diversity of the Application layer, it is the most complicated layer in the IoT architecture, and there has not been any universal standard to the construction of the application layer yet /21/. Security at the Application layer includes authentication, intrusion detection, risk assessment and data security /16/. Authentication is the first move in securing the application layer; verifying users' identities is necessary to prevent any unauthorised access. Intrusion detection techniques continuously monitor the activities of the system and generate warning or alarm whenever a suspicious activity occurs. Risk assessment identifies the information assets that could be targeted by attackers and detect threats that could affect those assets. Integrity and encryption mechanisms are vital to prevent data-stealing threats or capture threats. Moreover, firewalls are also a recommended method for application layer to monitor incoming and outgoing network traffic, reducing the risks of intruders. Some general threats, that the application layer has, are:

- Phishing attack: Attacker masquerades as a trusted party to lure the victim into opening an email, text message or instant message, through which the attacker gains access to the credentials of the victim.
- Virus, spyware and worms: Attacker can inject malicious software into the system to pilfer some kinds of data from the user, causing data corruption or Denial-of-Service.

2.3 Security Threats

In computer security, vulnerabilities are weaknesses in some aspect or feature of a system which could be exploited by a hacker; and threats are potentials for those

vulnerabilities to turn into attacks on the system. Based on the intended motives of possible attacks, threats can be classified into three categories: Capture threat, Disruptive threat and Manipulation threat /22, p. 25/. Table 1 shows the list of threats mentioned in the previous parts together with their categories. Table 2 illustrates what Misra et al. concluded about how each type of threat affects the security requirements /22, p. 29/.

Capture threats are threats of exposing data, which allows attackers to gain intelligence advantages to control affected parts in the system, perform attacks or get access to more information stored in the system. Although capture threat does not cause direct harm to the functioning of the system, it seriously violates the privacy aspect in the security requirements since private information in the system is shared to an unauthorised party.

Disruption threats directly cause harm to the system by disrupting its normal functioning in various levels from slowing down the performance of the targeted system, deny its service to wreck it. Disruption attacks target mainly on the availability of data and services, but they also violate any other aspects of IoT security.

Manipulation threat influence the decision-making abilities of an IoT system. Manipulation threats usually are corrupted data during generation or transmission; they can be as simple as giving false input data to sensors by manipulating the environmental condition at the place the sensors are installed, or they can be as complicated as faking the identities and modifying messages during transmission. Therefore, manipulation attacks seriously affect the integrity and authenticity of the data; other security aspects may also be violated depending on some instances.

Table 1. Threat types and their members

Threat types	Threats
Capture threat	Side-channel analysis
	Eavesdropping: Sniffing/Snooping
	Phishing
Disruption	Jamming
	DOS/DDOS
	Hello flood
Manipulation	Tamper
	Fault injection
	Tag cloning
	Replaying
	Man-in-the-middle
	Sinkhole
	Wormhole
	Selective forwarding

Table 2. System threats

Key Security Elements / *Threat types*	**Availability**	**Confidentiality**	**Integrity**	**Authenticity**	**Authorisation**	**Non-repudiation**
Capture threat	**X**	**O**	**O**	**O**	**O**	**O**
Disruption threat	**O**	**O**	**O**	**O**	**O**	**O**
Manipulation threat	**O**	**O**	**O**	**O**	**O**	**O**

X	Never compromised
O	Maybe compromised
O	Always compromised

3 SECURITY SOLUTION

3.1 Authentication and Encryption

Encryption and authentication are two of the essential parts of any security solution. The primary purpose of encryption is to prevent eavesdropping on transmissions by converting information or data into code using an algorithm, such as Rivest–Shamir–Adleman (RSA), Advanced Encryption Standard (AES). However, encryption itself does not have the ability to verify who is participating in communications. Therefore, authentication comes in place to exchange verifiable credentials and validate them. Strong IoT device authentication is needed to make sure that all connected devices on the IoT can be trusted to be what they claim to be.

3.1.1 Symmetric and Asymmetric Encryption

Cryptographic authentication is divided into two major categories: Symmetric and Asymmetric. The difference between the two categories is described in Figure 2 and 3 and discussed below.

Symmetrical encryption: is the most straightforward kind of encryption that involves one secret key to encrypt and decrypt data. This key can be a number or a string of letters. In order to read the message, the sender must exchange the key he uses to cypher the information with the receiver so that the message can be deciphered accurately. Exchanging the key is also a disadvantage of the symmetric-key encryption since the key is at risk of being exposed during the process. Some examples of the symmetric-key algorithm are AES, International Data Encryption Algorithm (IDEA), Data Encryption Standard (DES), Blowfish, RC4, RC5 and RC6.

Asymmetrical encryption uses two different keys, public and private, for encryption and decryption instead of one same key. The two keys are generated in pairs and are related to each other mathematically in such a way that a message encrypted by a public key can only be decrypted by its partner private key, while also, a message can be decrypted using a public key when a private key encrypts it. The public key is expected to be freely disclosed so that anyone can use it to send a message. The private key must be known only by one principal. An asymmetric-key is better in ensuring the security of data transmitted during communication but require more computational power and usually much slower compared to symmetric-key. Some common asymmetric-key algorithms are RSA, Diffie-Hellman (DH), Elliptic Curve Cryptography (ECC) and ElGamal.

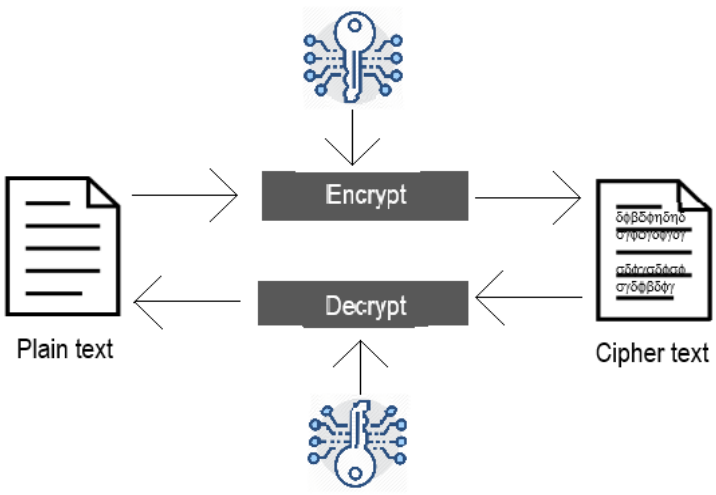

Figure 2. Symmetric-key encryption

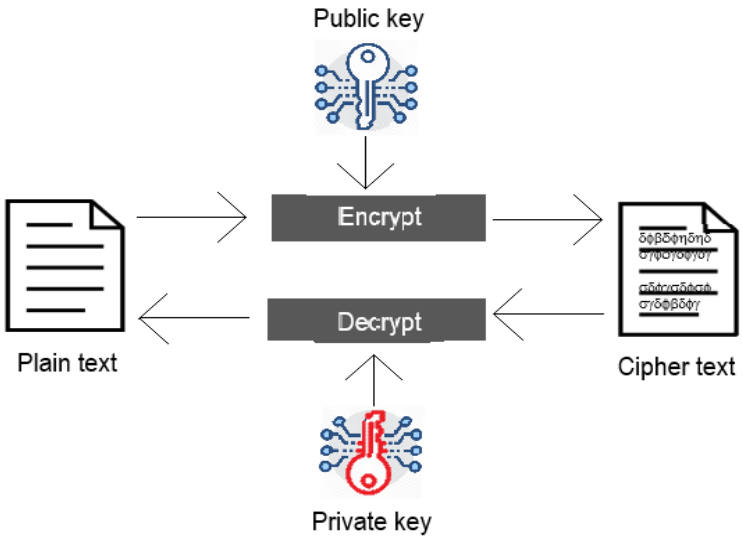

Figure 3. Asymmetric-key encryption

3.1.2 Cryptographic Hash Function

A cryptographic hash function is an algorithm used to map input of any size, such as an individual file or a string, into a fixed number of bits, called hash value. Cryptographic hash functions are one-way functions, whose primary purpose is to verify the authenticity of a piece of data. For example, two files are assumed to be identical only if the hash values, generated from each file by the same hash function, are identical; the hash values can also be used to locate the files but cannot be reversed into the original files. Some of the most common cryptographic hash functions are Secure Hashing Algorithm (SHA) version 2 and 3, Message-Digest Algorithm 5 (MD5), RACE Integrity Primitives Evaluation Message Digest (RIPEMD) and BLAKE2. All hash functions need to have four properties discussed by Daniel in order to become a useful one and be used widely: Computationally efficient, Deterministic, Pre-image resistant and Collision resistant /23/.

Cryptographic hash functions work in different ways compared to encryption. While a hash function is a one-way function as discussed, encryption is a two-way process where encrypted data can be reversed into the original one by using the key to decrypt it. In most cases, hashing only serves as a tool to compare a large amount of data or to index data, such as in hash tables.

3.1.3 Digital Signature

A digital signature is like an electronic, encrypted stamp of authentication on digital messages or documents. A valid digital signature confirms that the message was originated from the signer and has not been altered in transit. Digital signatures assure the authentication, integrity and non-repudiation requirements in cybersecurity /24/.

Digital signatures are often a combination of Public Key Infrastructure (PKI), which are based on asymmetric-key cryptography discussed earlier, and cryptographic hash function. When digitally signing a document, the signature is generated by encrypting the hash value of the document by using the private key of the sender (Figure 4). Here, the cryptographic hash is a unique pointer to the document, which is much smaller in size than the actual document, and the code received from encrypting the hash value using the sender's private key is a unique pointer to the sender. The time that the document was signed is also included when creating the signature. If any detail of the document changes after signing, the hash value changes and the digital signature is invalidated. Moreover, digital certificates, an electronic document, are issued by a Certificate Authority (CA) to bind a digital signature to an entity to verify the trustworthiness of the sender. The process of verifying a signature or the integrity of the document is described in Figure 5. The received signature is decrypted using the sender's public key to obtain the hash value of the document. This decrypted hash value is then compared with the hash value of the document received. The document is trusted to be unharmed only when the two hash values are identical.

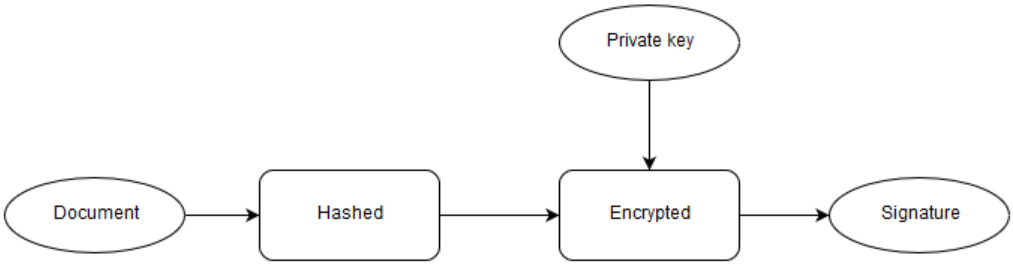

Figure 4. Generation of digital signature

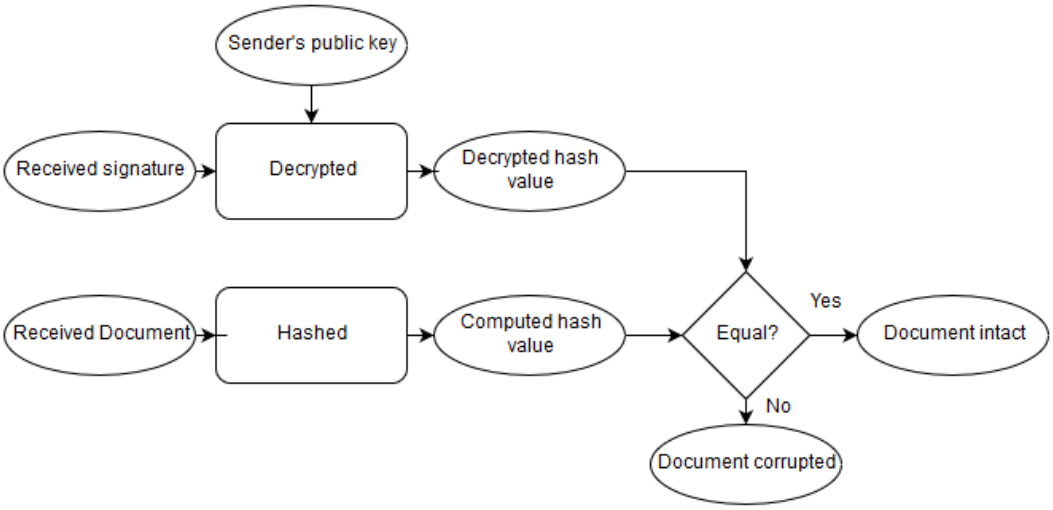

Figure 5. Digital signature verification

3.1.4 SSL/TLS

The Secure Sockets Layer, SSL, is the standard cryptographic security protocol for keeping an internet connection secure and The Transport Layer Security is an updated, more secure version of SSL. SSL/TLS prevents any sensitive data that is being transmitted between two systems, such as server and client or server to server, from leaking and being modified. SSL/TLS encryption is ideally suited to safeguard

the traffic in the IoT network layer /8, p. 32/ and is impossible to crack as it requires an enormous resource that beyond human capacity /25/.

At the beginning of the handshake, the client initiates the communication by sending the information that will be required by the server, such as SSL version number, cypher settings, session information. Then, the server responds with the configuration selected from the Client Hello, along with its digital certificate. If the server needs the client's certificate, it will send an additional certificate request in Server Hello. After the client successfully verifies the server's digital certificate, it creates a random byte string, known as a pre-master secret. The pre-master secret is sent to the server after it is encrypted with the server's public key. The client also sends its digital certificate if required by the server in the previous step. In the next step, the server verifies the client's certificate and uses its private key to decrypt the pre-master secret. A secret key is generated based on the agreed cypher between the client and the server. At this point, symmetric-key cryptography is used to encrypt data transferred between the server and the client as the computational cost is much lower compared to asymmetric encryption. Both the client and the server exchange a Finished message together with a "Change Cipher Spec" message to indicate that part of the handshake is complete, and they are now switching to symmetric encryption. Finished messages and all other application data after them are symmetrically encrypted with the negotiated secret key. Figure 6 illustrates the SSL/TLS standard handshake between a client and a server. /26, 27/

In the newest version TLS 1.3, a major improvement was made over its previous version handshake. TLS 1.3 handshake shown in Figure 7 cut the number of the round trip by half, resulting in a much lighter and faster protocol. Client Hello now includes a list of key-agreement protocol groups, called ECDHE, which the client supports and a key-share for some or all of the groups. In reply to Client Hello, Server Hello sends back selected protocol, its key share and digital certificate, followed by a Server Finished. Then, the client verifies the server certificate, generates a secret key and sends back the Client Finished to complete the handshake. /28/

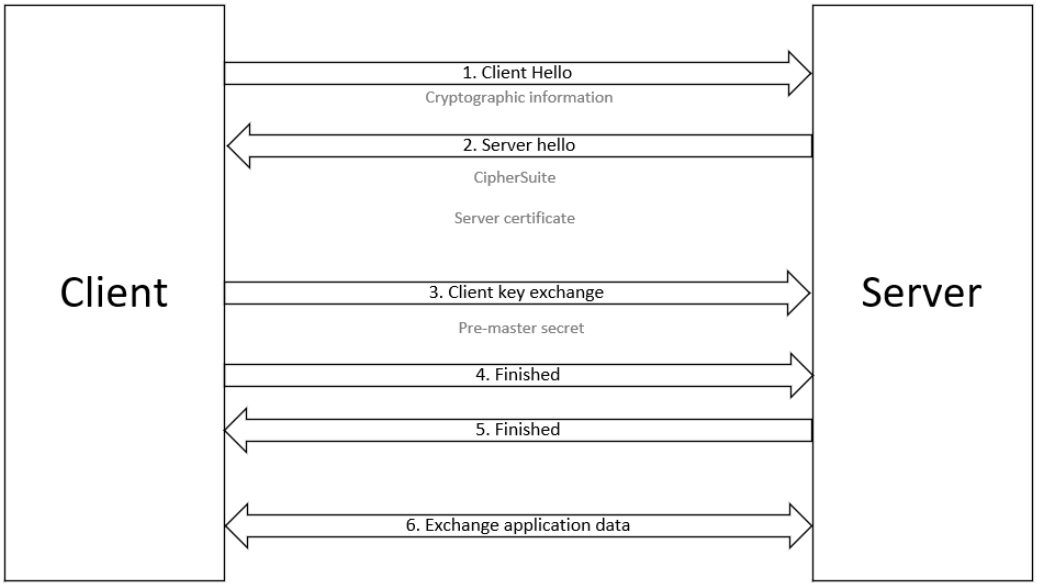

Figure 6. SSL/TLS standard handshake

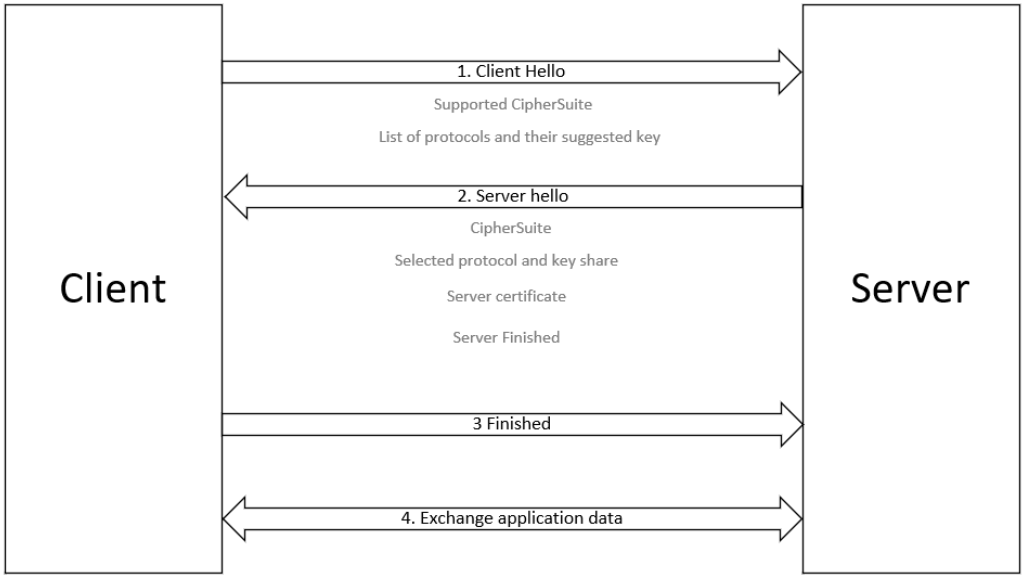

Figure 7. TLS 1.3 handshake

3.1.5 Lightweight cryptography

Although the conventional cryptography methods discussed previously are very secure and hard to crack, they usually require significant computing power and memory capabilities that IoT devices with constrained resources, such as sensor networks, RFID tags, or embedded systems, cannot afford. Therefore, lightweight cryptography (LWC) methods are proposed to be substitutes for conventional ones to provide end-to-end security solutions that are smaller and require less power supply and less memory for resources-limited devices /29/.

ISO/IEC 29192 is a standardisation project that describes the properties of LWC and specifies suitable algorithms. ISO/IEC 29192 part 1 defines the security, classification and implementation requirements. The minimum security strength for LWC is 80-bit security. LWC implementation on hardware considers chip area and energy consumption to be important metrics. In the software implementation case, code size and random access memory (RAM) consumption are preferable metrics for lightweight application /29/. In part 2, ISO/IEC 29192 specifies two block cyphers: PRESENT with 64 bits in block size and 80 or 128 bits in key size, CLEFIA with blocksize 128 bits and key size 128, 192 or 256 bits. ISO/IEC 29192 part 3 specifies two stream cyphers: Enocoro with 80 or 128 bits in key size and Trivium with a key size of 80 bits. Block cyphers and stream cyphers are listed as symmetric-key based lightweight mechanisms. Three lightweight mechanisms using asymmetric techniques specified in part 4 are a unilateral authentication mechanisms based on discrete logarithms on elliptic curves, also known as Elliptic light (ELLI), an authenticated lightweight key exchange (ALIKE) mechanism for unilateral authentication and establishment of session key and an identity-based signature mechanism, namely cryptoGPS, for wireless sensor network (WSN). Part 5 specifies three hash functions that are suitable for applications implementing lightweight cryptography: PHOTON, SPONGENT and Lesamnta-LW. However, LWC hash functions are still immature to adopt, but it is recommended to combine LWC block cyphers and hash functions /29, 30/.

Gunathilake et al. compared the performance of some of the most common light-weight block cypher with the classical AES method, and the result is shown in Table 3 /30/.

Table 3. Lightweight cryptographic Algorithms and AES method comparison

Cypher	Block size	Key size	Security	Target
AES	128	128	0.7	SW, HW
Fantomas	128	128	NA	SW
HIGHT	64	128	0.69	HW
LBlock	64	80	0.72	SW, HW
LED	64	80	NA	SW, HW
Piccolo	64	80	0.56	HW
PRESENT	64	80	0.84	HW
PRINCE	64	128	0.83	HW
RC5	64	128	0.9	SW
Robin	128	128	NA	SW
Simon	64	96	0.67	SW, HW
Speck	64	96	0.58	SW, HW
TWINE	64	80	0.64	SW, HW
*Note: NA = Not Applicable, SW = Software, HW = Hardware				

Gunathilake et al. conclude that the algorithms that have the highest security level tend to target either software or hardware attacks. The most secure algorithm is RC5 with a security level of 0.9. However, RC5 has a key size of 128, which is a bit outbound the requirement for LWC, and only targets software attacks. The second most secure algorithm is PRESENT, which is recommended in ISO/IEC 29192, has the smallest block size and key size but still reach a security level of 0.84. Other algorithms that handle both software and hardware attacks, such as LBlock, Simon, Speck, TWINE, seem to have lower strength varying from 0.58 to 0.72.

Table 4. Hardware performance of CLEFIA compared with AES

Cypher	Key length	Enc/Dec (cycles)	Key Setup (cy-cles)	Area (gates)	Freq. (MHz)	Speed (Mbps)	Speed/Area (Kbps/gate)
CLEFIA	128	18	12	5,979	225.83	1,605.94	268.63
		36	24	4,950	201.28	715.69	144.59
	192	22	20	8,536	206.56	1,201.85	140.81
	256	26	20	8,482	206.56	1,016295	119.89
AES	128	11	N/A	12,454	145.35	1,691.35	135.81
		54	N/A	5,389	131.24	311.09	57.63

Moreover, the other standard block cypher in ISO/IEC 29192, CLEFIA, supports block size and key size of 128, 192 and 256 bits, which is the same size as AES. However, CLEFIA is still considered to be lightweight due to its low cost and high efficiency in hardware implementation in comparison with AES. The result of the

hardware performance of CLEFIA in Table 4 provided by Sony Corporation has proved this conclusion /31/. With the same key length of 128 bits, the efficiency of CLEFIA is nearly double that of AES with 268.63 Kbps/gate compared to only 135.81 Kbps/gate.

3.2 Blockchain Integration

3.2.1 Blockchain Background

Blockchain was introduced in October 2008 by a pseudonymous developer Satoshi Nakamoto as a public transaction ledger of Bitcoin, a peer-to-peer (P2P) electronic cash system that would allow online payments to be sent directly from one party to another without the involvement of a central financial authority. Blockchain helps to solve the problem of trust in the information system when dealing with sensitive information like in economic transaction.

Blockchain is created with five fundamental principles /32/, namely,

- Distributed database: Each node on the blockchain have the right to access to the entire database and its modification history. All transaction record of any user of the blockchain can be verified without an intermediary.
- P2P transmission, in which peers communicate directly with each other rather than through a central node.
- Transparency with pseudonymity: Every transaction is visible to anyone who has access to the blockchain system. Each node or user is assigned with a unique 30-plus-character alphanumeric address. It is up to the users to stay anonymous or provide proof of identity to others. All transactions occur between those address.

- Irreversibility of records: Each record is cryptographically secured and linked with previous ones so that it cannot be altered once entered into the database.
- Computational logic: users can trigger transaction automatically between nodes using their own algorithms and rules.

There are two major types of blockchain: permissioned and permissionless blockchain. Popular cryptocurrencies on the market today, such as Bitcoin, Etherium and LiteCoin, are operated based on the permissionless blockchain, which can also be called public blockchain. In this type, the blockchain is open to the public; anyone can join the blockchain network without restrictions and becomes a full-node even when he has malicious intents if he verifies all the rules of the blockchain system. A full-node has the most up-to-date copy of the blockchain and participates in the process of verifying transactions and creating new blocks. On the other hand, permissioned blockchain, also called private blockchain, has certain restrictions placed on who is allowed to participate in the blockchain network and what transactions. Private blockchain ensures that there are no malicious participants or nodes; it is also an option for developing blockchain projects and applications that require internal data being kept in secret. Hyperledger Fabric and R3 Corda are two common examples of private blockchain platforms. In comparison with public blockchains, the private ones are much better in performance (high throughput) and scalability /33/, but are more centralised, relies on the credibility of the authorised nodes and are more vulnerable to hacks and data manipulation due to fewer nodes available.

Moreover, there are two other types of blockchain that are worth mentioning: Consortium and Hybrid blockchain. The Consortium blockchain, also known as Federated blockchain, is permissioned blockchain but instead of giving only a person or an organisation creating the blockchain full control, the power is shared among a group of approved people. This type of blockchain provides many of the same benefits of a private blockchain, such as efficiency, privacy and scalability, and is suitable for organisation collaboration. The Hybrid blockchain is an attempt to create a blockchain that uses the best part of both private and public blockchain solution.

Hybrid blockchain consists of a public blockchain that anybody can participate and a private network that restricts participation to those selected by a centralised body. This type of blockchain offers flexible control over what data is kept private versus shared on a public ledger.

Blockchain is basically a sequence of blocks of transactions in a particular order. Data can be stored as text files or in the form of a simple database. The structure of blockchain is similar to a link list, where each block has specific data and links to other blocks by using a pointer. Figure 8 shows an example of a blockchain, where each block has a header and a body. The block header includes a hash value that points to the previous block and a Merkle tree root hash, an authenticated data structure. The Merkel tree is designed to sign the content in a large body of data and check the signatures rapidly with a minimal amount of memory required /34/. The body of the block includes a transaction counter and transactions. The maximum number of the transactions in each block depends on the size of the transactions and the block size. Different blockchains may include other headers and body components that are suitable for their use cases. For instance, in Hyperledger Fabric version 1.0, block header includes three fields, a block number, Hash of the previous block's header and the hash of the data segment of the current block, block body consists of block data, such as version, timestamp, epoch, and block metadata /35/. On the other hand, in Ethereum, a block header includes many fields, such as parent hash, uncle hash, root hash, difficulty, gas limit and nonce but the body is kept simple with two fields transactions and uncles.

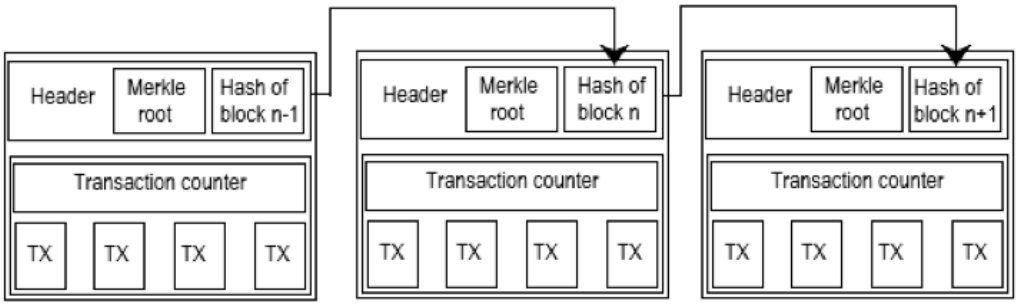

Figure 8. Example of blockchain

In a blockchain network, creating a new block and linking it to the chain requires performing specific steps to ensure security and consent. Blockchain makes use of asymmetric cryptography to safeguard the authentication, integrity and non-repudiation of the network. When making a transaction, all nodes prove their identity by signing it with their private key before transferring their asset to someone else in the same network. The nodes receiving the asset make sure that the incoming transaction is valid before moving on to the next step; Invalid transactions are discarded. For example, in Bitcoin, the verifying nodes need to identify the senders through their public key and check if they have a sufficient balance of cryptocurrency for the transaction. When all requirements are satisfied, the transactions are broadcasted throughout the network. Then, those transactions are ordered and packaged into a block. However, since any node can collect transactions to create a block and broadcast it to the rest of the network, the system may confuse about which block should be the next one in the blockchain. Therefore, every node participating in the process of creating a new block must go through a distributed consensus mechanism to decide which should be the next block. Baliga analysed and compared some most famous consensus mechanism, namely, Proof of Work (PoW), Proof of Stake (PoS), Proof of Elapsed Time (PoET), Byzantine Fault Tolerance (BFT), and Federated BFT /36/. Permissioned blockchains mainly use PoW and

PoS as their consensus mechanism while permissioned blockchains prefer BFT protocols more. Finally, only one node satisfying the consensus mechanism has the right to add its block to the chain.

Blockchain is usually considered to be immutable, or tamperproof because of its architectural design and operation. It is protected against attacks by a combination of different security methods, such as cryptographic hash and digital signature, a consensus mechanism, and a decentralised network. However, there is no perfect solution for security. Depending on each blockchain and the area it is applied to, there will be vulnerabilities that can be exploited by a hacker. For example, PoW consensus model has a risk of "51% attack", the case that the cybercriminal has control over more than haft of the nodes; poorly written smart contracts, self-executing software programs that specify and enforce contracts among two or more parties on the blockchain, may allow a hacker to infiltrate to steal, alter or divert wealth. Blockchain is still an emerging technology and improving day by day. Most vulnerabilities are patched up quickly by soft forks, or in extreme cases, a hard fork to upgrade the blockchain to a newer version. In general, blockchain is still a better security solution to storing and exchanging digital assets than any that comes before it.

3.2.2 Benefits

Blockchain is becoming an attractive technology for addressing many security and privacy challenges in IoT security. The following part will discuss some intrinsic features of the blockchain that can be useful for securing the IoT.

- Decentralisation: Distributed database and P2P communication can lower the need for a centralised entity. The number of IoT devices is proliferating at CAGR of nine-teen per cent between 2017 and 2023; by 2023, it will reach 20 billion devices /4/. The amount of communication between devices that need to be handled will increase costs exponentially. In a blockchain

network, the cost for handling massive traffic flow is eliminated since devices can communicate securely with each other and execute actions through smart contracts automatically. When the network does not rely on central control anymore, it also decreases delays and overcomes a single point of failure vulnerability, which is being exploited by DDOS attacks.

- Data integrity and authentication: As explained in [3.2.1], Data transmitted between peers in a blockchain network is always cryptographically proofed and signed by the real owner, which can be identified by its public key published on the network, ensuring authentication and integrity of the data. Transaction history recorded on the blockchain distributed ledger is immutable and can be tracked securely. By this design, blockchain is a reliable shield defending the IoT system from possible MITM attacks.
- Authorisation and privacy: Smart contracts allow blockchain users to set up access rules to connected IoT devices effectively and much easier compared to conventional authorisation protocols, such as OAuth 2.0, OpenID, Role Base Access Management (RBAC) and LWM2M /37/. The smart contract automatically executes on the system according to the rules and terms approved by all involved parties. No one can change the rules after the smart contract is executed; hence, ensure the privacy of the data.

3.2.3 Challenges

The main barrier in defining a robust security mechanism for IoT is its resource-constrained architecture. Since blockchain ledger is irreversible and cannot be modified, its size expands consistently during operation. For instance, Bitcoin blockchain has grown from only one gigabyte (GB) in 2010 to 233GB at the beginning of August 2019; if Bitcoin continues to grow at this speed, it is expected to reach 500GB in the next two years. This size seems to be all right for regular PC to become a full node, but it is a considerable amount for small IoT devices that are not

equipped with such large memory capacity. Not only memory size, but the processing power of IoT devices is also a big challenge when integrated with blockchain. Blockchain relies on consensus algorithm to ensure its immutability characteristic. The two most common algorithms that are employed widely in many blockchain systems, PoW and PoS, require significant computational resources that are beyond the limit of most IoT devices.

Another challenge is Scalability. When implementing a blockchain, all blocks have to be broadcasted over the network and verified by all nodes. The more devices/nodes in the IoT network, the more traffic flow when they broadcast data; this may cause a severe issue to low power IoT devices with limited bandwidth connections. Moreover, IoT devices need to exchange data with each other and with the cloud server frequently and quickly; the time interval may be counted in milliseconds. Therefore, the number of transactions per second that the blockchain can handle plays an essential role in the normal functioning of the system. In this case, public blockchain, such as Bitcoin is incompatible with IoT as it can only perform, in the best scenario, seven transactions per second /38/.

3.2.4 Proposals

Blockchain is a distributed database created to help multiple mutually mistrusting entity to interact directly with each other under a shared system instead of having to agree on a trusted third party. However, blockchain requires data to be shared and devices to have enough processing power, storage and network capacity that make it challenging to integrate blockchain into every system. To avoid redundant investment on the blockchain, Karl Wust and Arthur Gervais discussed some requirements that the developers need to consider to avoid redundant investment on the blockchain; they summarised the step to determine whether a blockchain is the appropriate technical solution in a flow chart as shown in Figure 9 /39/. Depending on the aims of each IoT project, developers have to choose which type of blockchain they should use. However, for most IoT system, private or consortium blockchain

is recommended due to its greater scalability, efficiency and the flexibility in managing access right and permission level of its member nodes. Being able to manage the access-list is critical for IoT devices that interact with customers and gather sensitive information. For example, information recorded by a smart grid about how much power is consumed in a specific time in the day must be kept in secret as it may help thieves to know exactly what time the owner may come home.

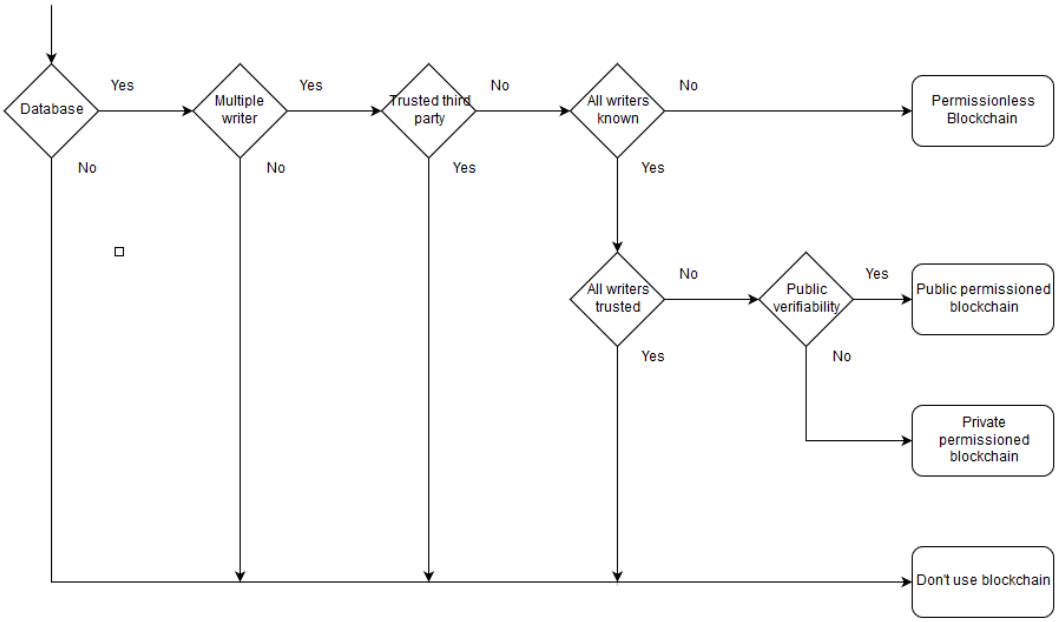

Figure 9. Where the blockchain makes sense

Besides pure private and consortium blockchain, a hybrid architecture was recently introduced for IoT in 2018 in /40/. The Hybrid blockchain is expected to get the best out of both public and private blockchain by exploiting both PoW and BFT protocols. Sagirlar et al. introduced a Sweet-spot Guideline, which is the base of the architecture, after performing numerous experiments to evaluate PoW performance with different block sizes and block generation intervals, device location and the number of IoT devices. The Guideline suggests that in order to achieve the highest performance, a PoW blockchain should contain a few hundreds of IoT devices

that are geographically close and frequently communicating with each other, block sizes should proactively be set to no more than 1MB, and block generation interval should be as short as possible to prevent security vulnerabilities. The Hybrid-IoT architecture is visualised in Figure 10. The blockchain is divided into multiple sub-blockchains. Sub-blockchains follow the Sweet-spot Guideline for PoW blockchain and are connected with each other by a BFT inter-connector framework. In a sub-blockchain, only IoT devices that have enough processing power and capacity to perform complex operations, such as Raspberry Pi 3, are allowed to become a full peer node and to participate in the consensus process. Less powerful devices, such as Arduino Uno, take a Light peer role that only performs simple tasks, such as sending transactions to a transaction pool and the full peers. Other devices that have minimal capability, such as sensors, cannot become a peer but can only connect to full peers for data fusion. Unlike Bitcoin, the difficulty of PoW puzzles in Hybrid-IoT is set according to the hardware constraints of IoT devices so that they can still perform their application tasks while mining blocks. Security experiment results in the publication paper of Sagirlar et al. show that sub-blockchains that follow the sweet-spot Guideline are able to have more or comparable total work even with low difficulty puzzles than those with high difficulty PoW puzzles but are not generated according to the Guideline. /40/

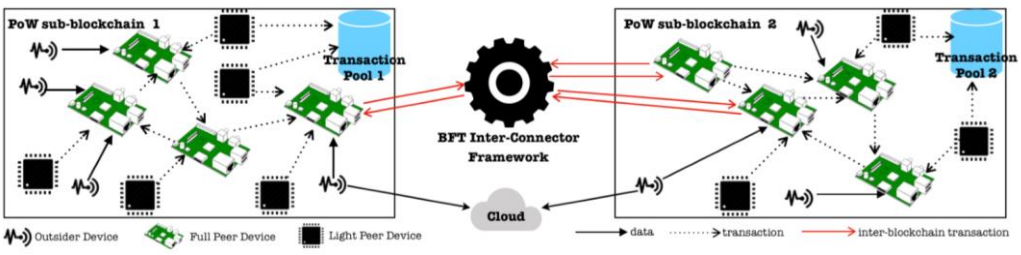

Figure 10. Hybrid-IoT high-level architecture

4 CONCLUSION

Many researchers have been forecasting the booming number of IoT devices in the next few years, raising the concerns about securing the enormous amount of data generated by those devices. This paper surveys the main challenges that IoT systems have to face. The security issues are analysed in each IoT architecture layers, and the threats are sorted into three categories based on their intended motives: Capture, Disruptive and Manipulation.

For the security solution, the paper gives an overview of one standard method, authentication and encryption, and a new method using an emerging technology blockchain. Digital signature and SSL/TLS are well studied and are currently implemented widely not only in IoT but also in different IT systems. SSL/TLS is considered to be very secure and ideal for safeguarding the network layer but requires devices to have enough computing power to process sophisticated algorithm and enough memory capacity to store the certificate. When dealing with resource-constrained devices, lightweight cryptography is more suitable to do the job. There are many different LWC algorithms available; however, it is recommended to refer to ISO/IEC 29192 standard when considering suitable LWC algorithms for an IoT system. Blockchain is known as the immutable ledger as it is a very secure way to store data. Blockchain is created with five fundamental principles: distributed, peer-to-peer, transparency, irreversibility and computational logic, which ensure the integrity, availability and non-repudiation of the data on the blockchain either at rest or during transmission. However, blockchain requires devices to have adequate processing capability and memory capacity to perform complex consensus algorithms. Scalability is also an issue that blockchain needs to overcome to works effectively in an IoT system. Hybrid-IoT is recently proposed as a possible architecture that allows an IoT system to have security features from both public and private blockchain with satisfactory performance.

In conclusion, no single solution is perfect for overcoming all security challenges but rather suitable for a specific situation. The safest methods, such as SSL/TLS or blockchain, tend to consume more energy and memory capacity. By lowering the complexity of the algorithm, light-weight alternatives can be applied to resource-constrained devices but with a lower security level. Therefore, it is recommended to combine different methods when securing any information technology system to achieve the best results.

As technology is evolving, IoT devices are becoming cheaper and more powerful; old resource-constrained devices will be gradually replaced with new one with more capacity to handle more advanced algorithm. However, the need for light-weight security methods will still be tremendous. Cryptography will still need much more research so that conventional algorithms become lighter and light-weight algorithms catch up with conventional one in security level. Moreover, blockchain is a very potential solution and it will be widely applied in many IoT system once its scalability issue is handled.

REFERENCES

/1/ Coetzee, L.,Eksteen, J. 2011. The Internet of Things - Promise for the Future? An Introduction. IST-Africa 2011 Conference Proceedings. Gaborone.

/2/ Minerva, R., Biru, A., Rotondi, D. 2015 .Toward a definition of the Internet of Things (IoT). Accessed 4.4.2019. https://iot.ieee.org/definition.html

/3/ Digicert. 2018 .State of IoT security survey. Accessed 26.4.2019. https://safenet.gemalto.com/iot-2018/

/4/ Ericsson. 2017 .Erricson Mobility Report. Accessed 26.4.2019. https://bit.ly/2sIITrg

/5/ Symantec. 2019. Internet Security Threat Report. https://www.symantec.com/security-center/threat-report

/6/ Gemalto. 2018. The state of IoT security. Accessed 26.4.2019. https://safenet.gemalto.com/iot-2018/

/7/ ARM. 2019. Technology survey & Predictions.

/8/ Li, S., Xu, L. D. 2017. Securing the Internet of Things. Todd Green.

/9/ Finjan. 2017. What is Non-Repudiation? Principles, Techniques and Best Practices. Accessed 26.4.2019. https://blog.finjan.com/what-is-non-repudiation/

/10/ Bilal, M. 2017. A Review of Internet of Things Architecture, Technologies and Analysis Smartphone-based Attacks Against 3D printers.

/11/ Rahul, M., Muntjir, M., Alhumyani, H. 2017. An Analysis of Internet of Things(IoT): Novel Architectures, Modern Applications, Security Aspects

and Future Scope with Lastest Case Studies. Building Services Engineering Research and Technology. 6.

/12/ Rashmi. 2018. IoT Concept and Improve Layered Architecture. International Journal of Engineering Development and Research 2018. 6, 2.

/13/ Skorobogatov, S. 2012. Chapter 7 - Physical Attacks and Tamper Resistance. Introduction to Hardware Security and Trust. New York. Springer.

/14/ Bhunia, S., Tehranipoor, M. 2019. Chapter 10 - Physical Attacks and Countermeasures. Hardware Security, 245-290.

/15/ Juels, A. 2006. RFID security and privacy: a research survey. IEEE Journal on Selected Areas in Communications. 24, 2, 381-394.

/16/ Farooq, M.,Waseem, M., Khairi, A., Mazhar, S. 2015. A Critical Analysis on the Security Concerns of Internet of Things. International Journal of Computer Applications. 111, 7.

/17/ Zhang, K., Liang, X., Lu, R., Shen, X. 2014. Sybil Attacks and Their Defenses in the Internet of Things. IEEE Internet of Things Journal. 1, 5, 372-383.

/18/ Khan, W.Z., Xiang, Y., Aalsalem, M. Y., Arshad, Q. 2012. The Selective Forwarding Attack in Sensor Networks: Detections and Countermeasures. International Journal of Wireless and Microwave Technologies. 2, 33-44.

/19/ Dubey, A., Meena, D., Gaur, S. 2014. A survey in Hello Flood Attack in Wireless Sensor Networks. International Journal of engineering Research & Technology (IJERT). 3, 1.

/20/ Cybersecurity Insiders. 2019. Cloud Security Report. Accessed 26.6.2019. https://go.deltarisk.com/2019-cloud-security-report

/21/ Rizvi, S., Pfeffer, J., Kurtz, A., Rizvi, M. 2018. Securing the Internet of Things (IoT): A Security Taxonomy for IoT. 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 163-168. New York. IEEE.

/22/ Misra, S., Maheswaran, M., Hashmi, S. 2017. Security Challenges and Approaches in Internet of Things. SpringerBriefts in Electrical And Computer Engineering.

/23/ Daniel. 2018. Cryptographic Hash Functions Explained: A Beginner's Guide. Accessed 10.7.2019. https://komodoplatform.com/cryptographic-hash-function/

/24/ AET. 2017. The difference between a digital signature and digital certificate. Accessed 10.7.2019. https://bit.ly/2MVBKz0

/25/ Martins, F. 2014. Cracking SSL Encryption is Beyond Human Capacity. Digicert. Accessed 11.7.2019. https://www.digicert.com/blog/cost-crack-256-bit-ssl-encryption/

/26/ Rescorla E., Dierks, T. 2008. The Transport Layer Security (TLS) Protocol Version 1.2. Accessed 12.7.2019. https://tools.ietf.org/html/rfc5246.

/27/ Freier, A., Karlton P., Kocher, P. 2011. The Secure Sockets Layer (SSL) Protocol Version 3.0. Accessed 12.7.2019. https://tools.ietf.org/html/rfc6101#section-5.5

/28/ Rescorla, E. 2018. The Transport Layer Security (TLS) Protocol Version 1.3 Accessed 12.7.2019. https://tools.ietf.org/html/draft-ietf-tls-tls13-28.

/29/ Katagi M., Moriai, S. 2008. Lightweight Cryptography for the Internet of Things. Sony Corporation.

/30/ Gunathilake, N., Buchanan, W., Asif, R. 2019. Next Generation Lightweight Cryptography for Smart IoT Devices: Implementation, Challenges and Applications. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT). Doi: 10.1109/WF-IoT.2019.8767250

/31/ Sony Corporation. 2007. The 128-bit Blockcipher CLEFIA Security and Performance Evaluations. Accessed 18.7.2019. https://bit.ly/31w8VNZ

/32/ Iansitim M., Lakhani, K. R. 2017. The Truth About Blockchain. Havard Bussiness Review. Accessed 24.7.2019. https://hbr.org/2017/01/the-truth-about-blockchain

/33/ Jaeger, G. L. 2018. Public versus private: What to know before getting started with blockchain. Accessed 25.7.2019. https://ibm.co/2BZ8P7J.

/34/ Merkel, R. C. 1988. A Digital Signature Based on a Conventional Encryption Function. Pomerance C. (eds) Advances in Cryptology — CRYPTO ’87. CRYPTO 1987. Lecture Notes in Computer Science, 293. Springer, Berlin, Heidelberg.

/35/ Thakkar, P., Nathan, S., Viswanathan, B. 2018. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 264-276. Milwaukee. IEEE.

/36/ Baliga, A. 2017. Understanding Blockchain Consensus Models. Persistent System Ltd.

/37/ Khan, M. A., Salah, K. 2018. IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems. 82, 395-411.

/38/ Vukolié, M. 2015. The Quest for Scalable Blockchain Fabric:Proof-of-Work vs. BFT Replication. Inter-national Workshop on Open Problems in Network Security (iNetSec), 112-125. Zurich. HAL-Inria.

/39/ Wüst, K., Gervais, A. 2018. Do you Need a Blockchain?. 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), 45-54. Zug. IEEE.

/40/ Sagirlar, G., Carminati, B., Ferrari, E., Sheehan, J. D., Ragnoli, E. 2018. Hybrid-IoT:HybridBlockchainArchitectureforInternetofThings-PoW Sub-blockchains. arXiv:1804.03903v3.