Vietnam National University, Hanoi
University of Engineering and Technology

Advanced Institute of Engineering and Technology

Technical Report

UET-AVITECH-2020001 November, 2020

Analyze and identify potential cyber-security risks in Industry 4.0

Bui Minh Tuan, Tran Viet Khoa, Nguyen Linh Trung,
Dinh Thai Hoang, Diep N. Nguyen, Nguyen Viet Ha,
Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Analyze and identify potential cyber-security risks in Industry 4.0

Bui Minh Tuan[1], Tran Viet Khoa[1], Nguyen Linh Trung[1], Dinh Thai Hoang[2], Diep N. Nguyen[2], Nguyen Viet Ha[1], Eryk Dutkiewicz[2]

[1] AVITECH, VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[2] School of Electrical and Data Engineering, University of Technology Sydney, Australia

November, 2020

Abstract

In light of Industry 4.0, the new technologies have brought huge benefit to a wide range of industries. In this work, we introduce the common vulnerabilities and potential cyber-security risks and focus on manufacturing system. Based on its architecture and operating principle, we analyse the security threats that directly affect on the system. This will be fundamental for analysing and building risk assessment tools.

Index Terms

Cybersecurity, IoT, Industry 4.0, industrial control systems, operational Technology, cyberattack detection, intrusion detection.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project on Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. The architecture and operating principle of manufacturing systems in Industry 4.0

A. Architecture of Manufacturing Systems in Industry 4.0

Industry 4.0 bases on the technical integration of Cyber-Physical Systems (CPS) in production and the application of Internet of Things (IoT) services in industrial processes (Fig. 1) [1]. Its basic principle is that a manufacturer can create “intelligent” networks with significantly reduced intervention by operators by connecting machines that communicate and control each other autonomously.

1) Cyber–Physical Systems Architecture: The structure and methodology of CPS as guidelines for its implementation in Industry 4.0 [2] is illustrated in Figure 2, 3, and 4.

2) Internet of things-IoT: The Internet of Things involves adding sensors and networking technologies to the devices and systems that we use every day in the physical world. In [3], the concept of IoT was defined as “Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications. This is achieved by seamless ubiquitous sensing, data analytics and information representation with Cloud computing as the unifying framework”.

B. Operation Principle of Manufacturing Systems in Industry 4.0

The transition to the Industry 4.0 is facilitated through the advances of the suportive technologies related to adaptive robotics, embedded systems, cloud technologies, virtualization technologies, augmented reality, data analytics and artificial intelligence, information and communication technologies (e.g. Cloud and networking), smart sensors and others [4] (Fig. 4).

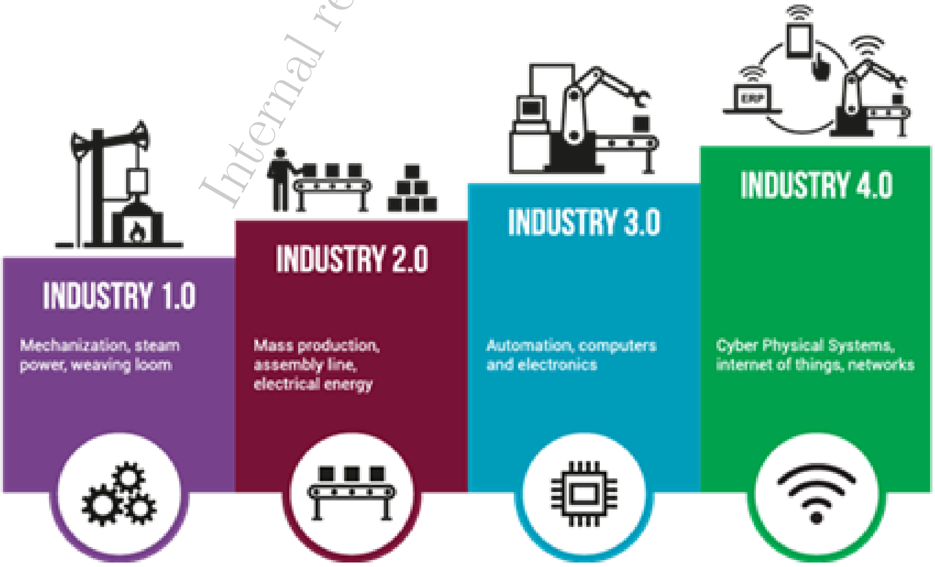

INDUSTRY 1.0
Mechanization, steam power, weaving loom
INDUSTRY 2.0
Mass production, assembly line, electrical energy
INDUSTRY 3.0
Automation, computers and electronics
ERP
INDUSTRY 4.0
Cyber Physical Systems, internet of things, networks

Fig. 1: Industry Revolution [1]

CPS are open, linked-up systems that operate flexibly, cooperatively (system-to-system cooperation), and interactively (human-to-system cooperation). They link the physical world seamlessly with the virtual world of information technology and software [2], and to do so they use various types of available data, digital communication facilities, and services. CPS are closely related to the Internet of Things, where seamless communication among physical objects is achieved through embedded systems and communication networks.

A step forward is the integration of the digital tools related to product development into Cloud platforms, in order to enhance collaboration among the various actors at this stage of the product lifecycle providing ubiquitous access to information [2]. It can be concluded that

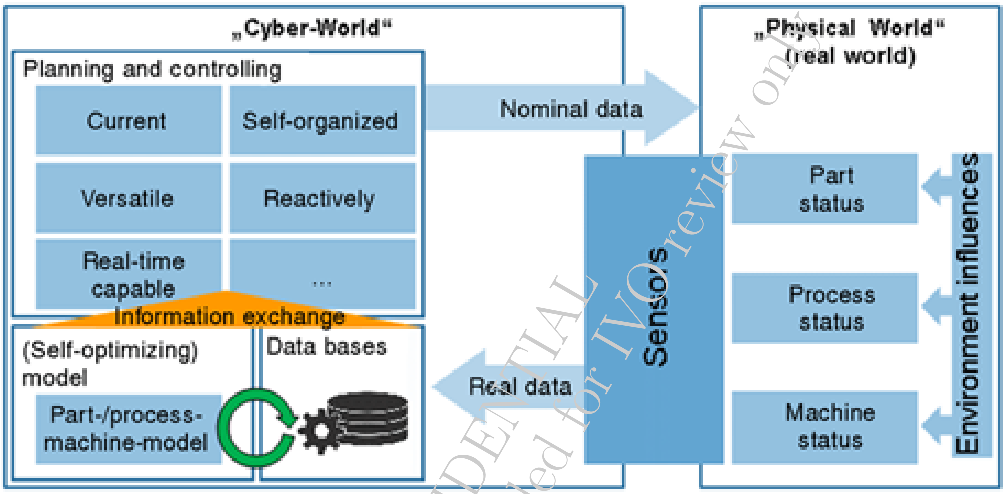


Fig. 2: Cyber–Physical Systems logical architecture.

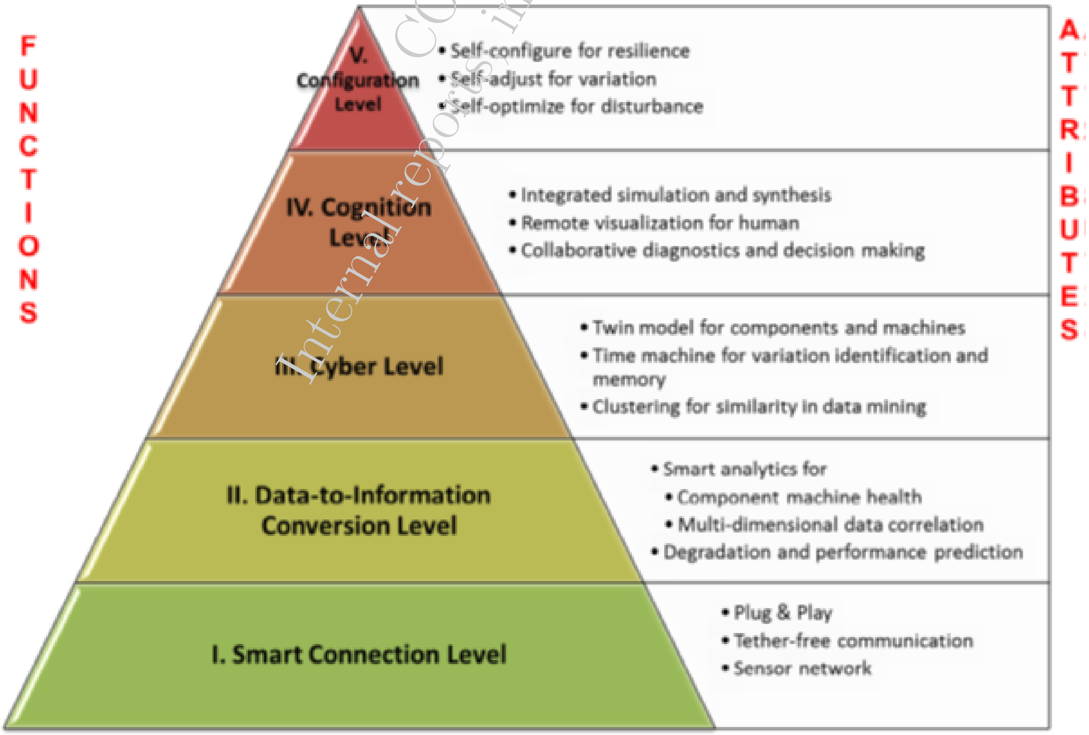

FUNCTIONS
ATTRIBUTES
V. Configuration Level
• Self-configure for resilience
• Self-adjust for variation
• Self-optimize for disturbance
IV. Cognition Level
• Integrated simulation and synthesis
• Remote visualization for human
• Collaborative diagnostics and decision making
III. Cyber Level
• Twin model for components and machines
• Time machine for variation identification and memory
• Clustering for similarity in data mining
II. Data-to-Information Conversion Level
• Smart analytics for
• Component machine health
• Multi-dimensional data correlation
• Degradation and performance prediction
I. Smart Connection Level
• Plug & Play
• Tether-free communication
• Sensor network

Fig. 3: Cyber – Physical Systems leveled architecture.

the latest advances in ICT technologies are the means for the interconnection of the different elements of a manufacturing system towards a digitalized ecosystem.

Integration standards with semantic representation of information, such as **OPC-UA**, along with **Web services** and **wireless sensor networks** enable the seamless communication among tangible resources and humans. Especially, with the use of mobile and wearable devices humans enter the cyber world and communicate with machines

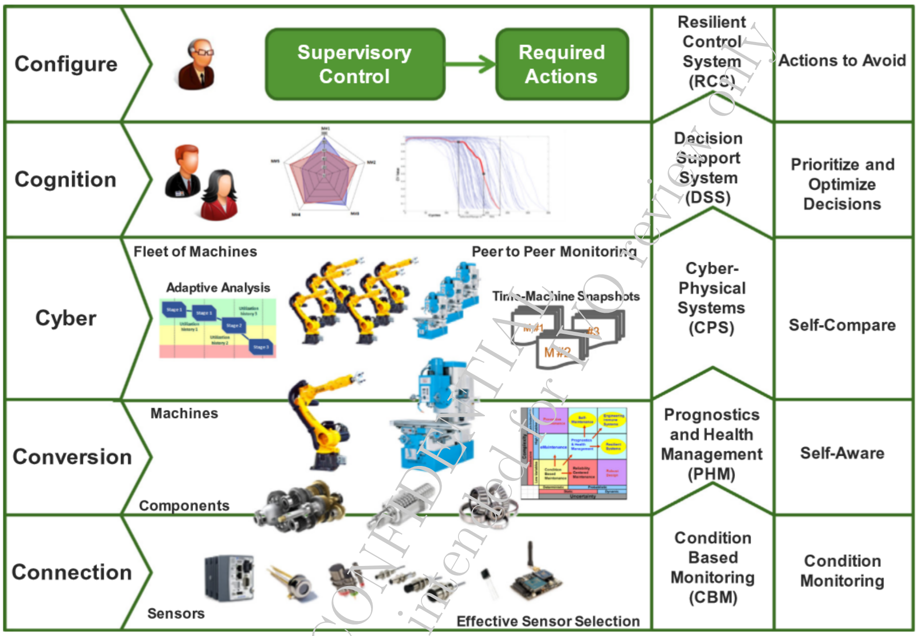
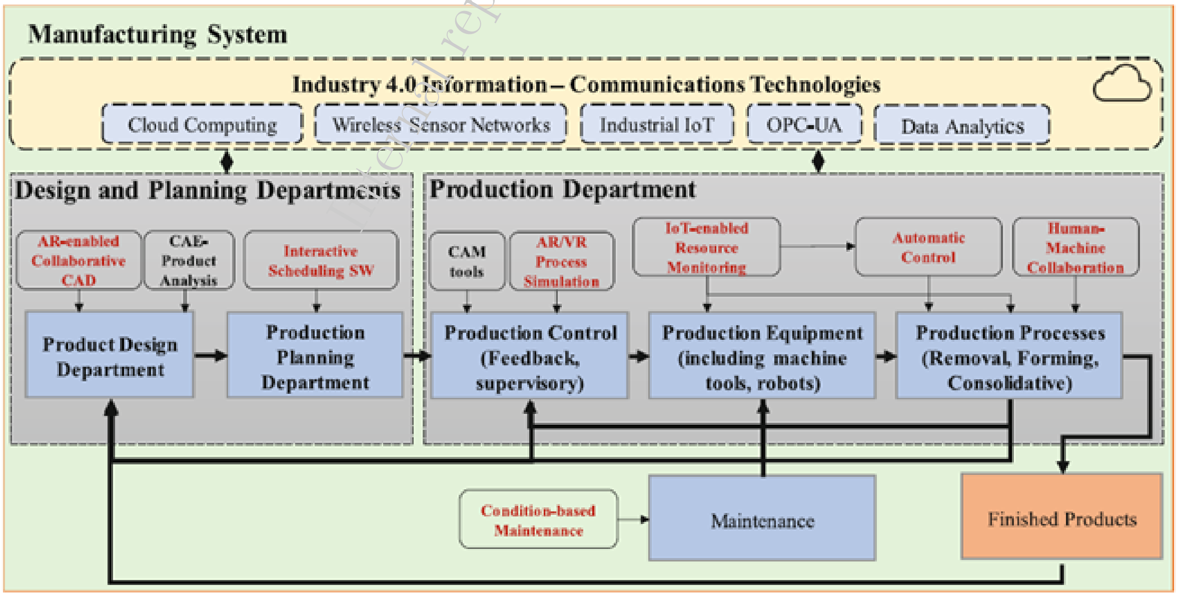


Fig. 4: Applications and techniques with each level of the CPS architecture.



Manufacturing System
Industry 4.0 Information – Communications Technologies
Cloud Computing
Wireless Sensor Networks
Industrial IoT
OPC-UA
Data Analytics
Design and Planning Departments
AR-enabled Collaborative CAD
CAE-Product Analysis
Interactive Scheduling SW
Product Design Department
Production Planning Department
Production Department
CAM tools
AR/VR Process Simulation
IoT-enabled Resource Monitoring
Automatic Control
Human-Machine Collaboration
Production Control (Feedback, supervisory)
Production Equipment (including machine tools, robots)
Production Processes (Removal, Forming, Consolidative)
Condition-based Maintenance
Maintenance
Finished Products

Fig. 5: Manufacturing work flow in Industry 4.0 [2]

II. Vulnerabilities And Potential Cyber-Security Risks of Manufacturing System In Industry 4.0

It is estimated that cyber risks costs the global economy up to $400 billion a year maybe even more [5]. For industrial control systems (ICSs) however, the risks are even more acute in the Industry 4.0. To understand the risk, we need a definition.

A. What is Vulnerability, Threats, and Risk?

Vulnerability - Weaknesses or gaps in a security program that can be exploited by threats to gain unauthorized access to an asset.

Threat - Anything that can exploit vulnerability, intentionally or accidentally, and obtain, damage, or destroy an asset.

Risk - The organizations such as the International Standards Organization (ISO) and National Institute of Standards and Technology (NIST) have developed definitions that are widely accepted and used. In both cases, risk is seen as a function of the vulnerability of an asset, the threat, which is the likelihood an attack will occur, and the consequence of such an attack being successful. So risks are defined by ISO and NIST as:

ISO: The potential that a given threat will exploit *vulnerabilities* of an asset or group of assets and thereby cause *harm* to the organization.

NIST: A function of the likelihood of a given threat-source's exercising a particular potential *vulnerability*, and the resulting impact of that adverse event on the organization.

To put those definitions in another way:

$$Risk = Vulnerability \times Threat \times Consequence.$$

B. Steps guide to Identify Cyber-security risks

Step 1 - Knowing Your Vulnerabilities: A vulnerability is any quality of an asset that could allow it to be exploited. All digital assets have them. Some are known; some aren't. Some are easier to exploit than others.

Step 2 - Identifying Threats: It is threats that turn vulnerability into an incident so we need to know about the importance of regular review and understanding the relationship between threats and vulnerabilities.

Step 3 - Measuring Consequences - The Final Piece: Consequences put these threats and vulnerabilities into perspective.

Step 4 - Bringing it together and Measuring Risk: Understanding and addressing the preceding elements gives a plant what it needs to begin to make a realistic assessment of its risks.

C. Security Threats and Vulnerabilities of IoT

Different architectures have been proposed by different researchers. In general, the IoT can be divided into four main levels. Figure 5 shows both the level architecture of the IoT and some basic components in each level.

- *Perception (Sensing) layer*: The perception layer is also called as 'Sensing Layer'. It composed of physical objects and the sensing devices such as various forms of sensory technologies, RFID sensors. These technologies allow devices to sense other objects.
- *Network layer*: Network layer is the infrastructure to support wireless or wired connections between sensor devices and the information processing system.
- *Service layer*: This layer is to ensure and manage services required by users or applications. It is responsible for the service management and has a link to the database.
- *Application (Interface) layer*: Application or interface layer composed of interaction methods with users or applications. It is responsible for delivering application services to the user.

1) Common security threats and vulnerabilities in the perception layer:

- *Unauthorized access*: At first node, unauthorized accesses are important threats due to physical capture or logic attack.
- *Confidentiality*: Attackers can place malicious sensors or devices in order to acquire information from the system.
- *Availability*: The system component stops working because it is physically captured or logically attacked.
- *Noisy data (transmission threats)*: The data may contain incomplete information or incorrect information due to transmission over networks covering large distances.
- *Malicious code attacks*: Attackers can cause software failure through malicious code such as virus, Trojan, and junk message.

2) Common security threats and vulnerabilities in the network layer:

- *Denial of Services (DoS) attack*: Attackers continually bombard a targeted network with failure messages, fake requests, and/or other commands. DoS attacks are the most common threat to the network.
- *Routing attack*: These are attacks on a routing path such as altering the routing information, creating routing loops or sending error messages.
- *Transmission threats*: These are threats in transmission such as blocking, data manipulation, interrupting.
- *Data breach*: A data breach is the intentional or unintentional release of secure or confidential information to an untrusted environment.
- *Network congestion*: A large number of sensor data along with a large number of device authentication can cause network congestion.

3) Common security threats and vulnerabilities in the services layer:

- *Manipulation*: The information in services is manipulated by the attacker.
- *Spoofing*: The information is returned by an attacker to spoof the receiver.
- *Unauthorized access*: Abuse of services accessed by unauthorized users.
- *Malicious information*: Privacy and data security are threatened with malicious tracking.

- *DoS attacks*: A useful service resource is made unavailable by being exposed to traffic above its capacity.

4) Common security threats and vulnerabilities in the application layer:

- *Configuration threats*: Failing configurations at interfaces and/or incorrect misconfiguration at remote nodes are the most important threats for this layer.
- *Malicious code (Malware) attacks*: These attacks are intentionally made directly to the software system in order to intentionally cause harm or subvert the intended function of the system.
- *Phishing Attacks*: In the interface layer, attackers may attempt to obtain sensitive information such as usernames, passwords, and credit card details.

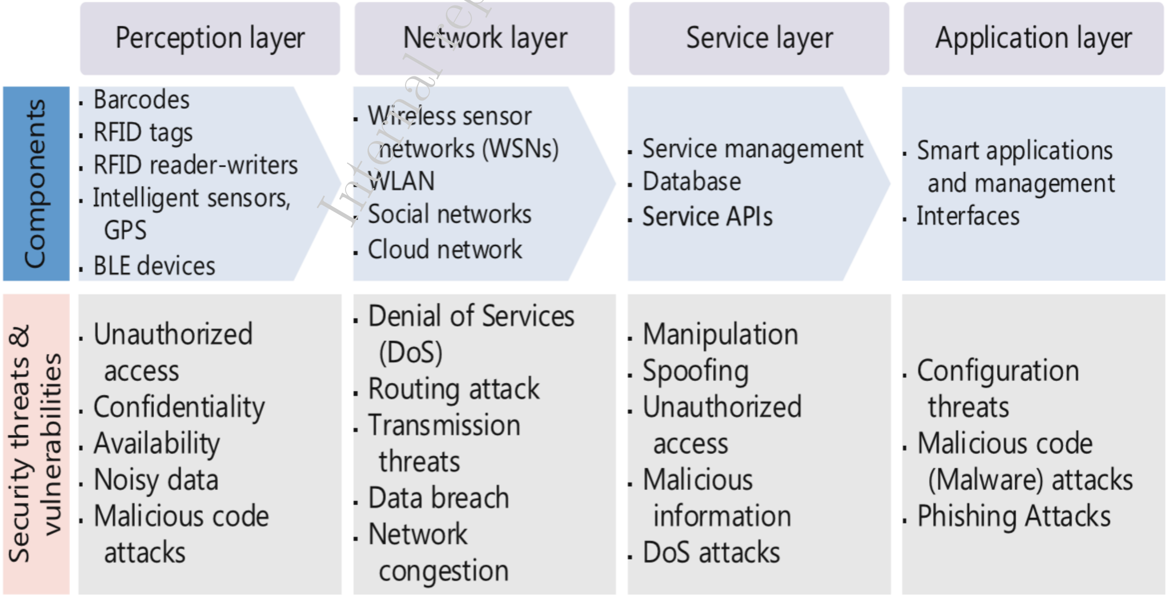
D. Security Threats and Vulnerabilities of CPS

E. Evolution of Cyber Attacks

Due to the widespread use of IoT based on computer networks, hackers have taken advantage of network-based services to gain personal benefit and reputation. Further, the cyber landscape is constantly altering and evolving due to the speed of technological change, the complexity of the attackers, the value of potential targets and the effects of attacks (Weber and Studer 2016). As a result, over time, the nature of cyber-attacks has been complicated and extremely sophisticated (Figure 6).

F. Security Challenges in Industries

The current application areas include smart manufacturing, smart homes, and smart cities, transportation and warehousing, healthcare, retail and logistics, environmental monitoring,



	Perception layer	Network layer	Service layer	Application layer
Components	· Barcodes · RFID tags · RFID reader-writers · Intelligent sensors, GPS · BLE devices	· Wireless sensor networks (WSNs) · WLAN · Social networks · Cloud network	· Service management · Database · Service APIs	· Smart applications and management · Interfaces
Security threats & vulnerabilities	· Unauthorized access · Confidentiality · Availability · Noisy data · Malicious code attacks	· Denial of Services (DoS) · Routing attack · Transmission threats · Data breach · Network congestion	· Manipulation · Spoofing · Unauthorized access · Malicious information · DoS attacks	· Configuration threats · Malicious code (Malware) attacks · Phishing Attacks

Fig. 6: Security threats and vulnerabilities by level [].

smart finance, and insurance. There are security challenges associated with all these application areas. Some of them are very obvious, for example, misuse of personal information, financial abuse. On the other hand, others are more specific depending on the structure of the industry. The industry challenges facing cyber security experts are outlined in Table I according to the industries

G. Most Frequently Targeted Industries

According to The IBM X-Force Threat Intelligence Index 2019 [6], 10 most frequently targeted industries in 2018 are ranked in percentages in the bar chart (Fig. 8).

Finance and Insurance: 19% According to X-Force data analysis, the finance and insurance sector has been the most-attacked industry for three years in a row, with 19 percent of total attacks and incidents in 2018. The allure financial services presents to a cybercriminal is clear: customer bank account information or payment card data can be monetized rapidly. Access to bank networks and switches for shifting large sums of money into criminal-controlled accounts, or robbing customer or employee Personally Identifiable Information (PII) can all lead to direct financial profit or be sold on the dark web.

Transportation Services: 13% The second most targeted sector, transportation services, includes airlines, bus, rail, and water transportation services, ranked second in 2018, and experienced 13 percent of total attacks and incidents. This sector, part of any country's critical infrastructure, is an attractive target for malicious threat actors. From financially motivated attackers seeking payment card information, PII, and loyalty-reward accounts to state-sponsored, advanced persistent threat (APT) groups aiming to disrupt the economy or

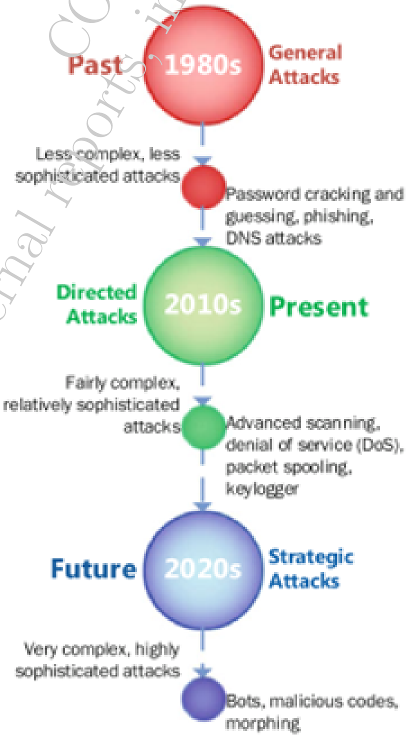

Past
1980s
General Attacks
Less complex, less sophisticated attacks
Password cracking and guessing, phishing, DNS attacks
Directed Attacks
2010s
Present
Fairly complex, relatively sophisticated attacks
Advanced scanning, denial of service (DoS), packet spooling, keylogger
Future
2020s
Strategic Attacks
Very complex, highly sophisticated attacks
Bots, malicious codes, morphing

Fig. 7: Evolution of cyber-attacks

TABLE I: Challenges according to the industry.

Finance	Protecting privacy and data security
	Managing third-party risk: Outsourcing contracts, such as cloud service agreements, impose complex data sharing regulations and generate a host of new cybersecurity challenges
	Emerging and advanced cyber threats
	Regulatory compliance
Energy	Protecting privacy and data security
	Lack of skills and awareness
	Information sharing: Many organizations do not share information about threats or cooperate externally
	Integrity of components used in energy systems
	Increased interdependence among market players
	Alignment of cyber security activities: All activities be aligned and fully integrated with national cybersecurity
Healthcare	Protecting privacy and data security: Healthcare organizations are required to comply with the Health Insurance Portability and Accountability Act (HIPAA), which requires healthcare vendors to ensure that the privacy of user data is not compromised in any case (Zhang and Liu 2010)
	Medical equipment issues: Healthcare organizations have specialized medical equipment that could pose particular security challenges (Korolov 2015)
	Managing third-party risk: Healthcare organizations are hesitant to move to cloud data protection to ensure that sensitive information is protected without leaving the company network (Zhang and Liu 2010)
Transportation	Protecting privacy and data security especially in the cargo industry (Xu et al. 2014)
	Emerging and advanced cyber threats (DoS attacks, Spoofing attacks) (Warren and Hutchinson 2000)
Manufaturing	The Industrial Control Systems (ICS) are isolated systems meaning that there is no interface with the company's network to protect it, and these systems are regulated against compliance standards. The result is a lack of security features, such as authentication and encryption. Passwords might be shared and can be easily stolen

target intellectual property data, attacks on the sector are on the rise. The transportation industry's extensive reliance on information technology to facilitate operations and its use of third-party vendors, presents an extended attack surface for various types of threat actors that either seek access to targeted data or aim to cause disruption.

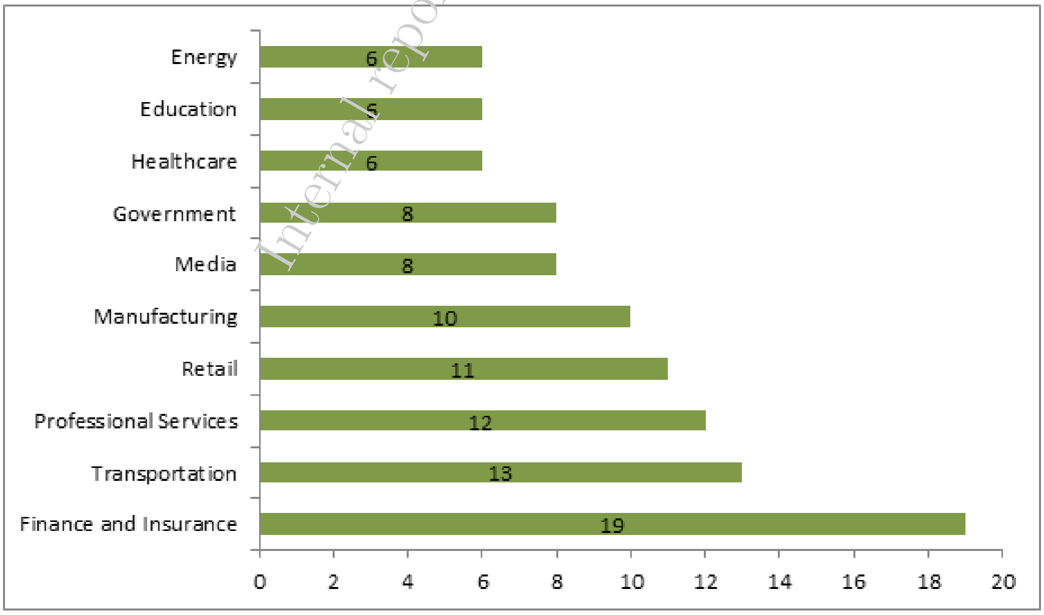


Fig. 8: Most frequently targeted industries in 2018 [6]

Professional Services: 12% The professional services sector made up of companies that provide specialized consulting services, such as legal, accounting, and architecture firms have come under increased risk for cyber-attack over the past several years. Malicious actors have discovered the value of the information these companies' process and house. Combined with their smaller security budgets, limited security staff, and a relatively immature security posture (in most cases), this sector is as vulnerable as it is lucrative.

Retail: 11% As the fourth-most targeted industry, retail experienced 11 percent of the total attacks and incidents in 2018. Retail companies sell products to consumers and businesses—from automobiles and apparel, to electronics, food, and furniture. More importantly, this sector works in hybrid mode: Services are extended to customers both onsite and over the internet, making for a decentralized and heterogenous operational environment. Retailers are attacked with Point-of-Sale (PoS) malware, skimming, and counterfeit card heists. They also experience sophisticated attacks on their web applications and service portals by fraudsters and organized cyber-criminals.

Manufacturing: 10% The fifth-most targeted industry is manufacturing, which includes companies that make a wide variety of goods, from chemicals and machinery to transportation equipment and electronics, and Internet-of-Things (IoT) devices. It experienced 10 percent of the total attacks and incidents. As the majority of cyber incidents in the manufacturing sector do not involve customer information that is subject to legal disclosure regulations, the percentage of publicly disclosed events in this industry is low when compared with other sectors. The numbers are therefore likely to be higher than those reported. Most attacks on manufacturing companies appear to target intellectual property (IP) and trade secrets. Confidential business communications, such as executives' email correspondence or company bank accounts are particularly lucrative targets for cybercriminals, nation-state groups, and even paid hackers hired by a competitor. This sector also absorbs many BEC attacks since manufacturers often wire substantial amounts of money to countries in Asia, Africa, and other developing regions.

Media: 8% The media sector, the sixth-most targeted industry, includes companies that produce, process, or distribute information and entertainment content. It also includes sub-industries, such as computer software and telecommunications, among others. This industry made up eight percent of the total attacks and incidents. The media sector also experienced the most publicly disclosed incidents, at 40 percent in 2018. Half of these publicly disclosed media incidents involved misconfiguration of systems or cloud servers, rather than premeditated attacks.

Government 8% The seventh-most targeted industry is government, and it experienced eight percent of the total attacks and incidents. X-Force researchers assess nation-state-backed groups are those most likely to target this sector. Depending on the type of objective the attack has, nation-state sponsored groups that breach government resources may use, sell, or deliver compromised information to their respective governments, typically for economic or political gain. Many times, these attacks are after top secret intellectual property. In other attacks, stolen data is used in espionage for the establishment of surveillance operations.

Healthcare: 6% Cyber security in the eighth-most targeted industry, healthcare, guards not only protected health information (PHI) and payment card data, but critical systems and devices that for some patients can mean the difference between life and death. The 2018

Ponemon Cost of a Data Breach study reveals the healthcare industry has the highest cost per record breached in a cyber-incident, at $408. This cost is nearly twice the amount of the next-highest industry financial services at $206 per record breached, and far above the grand average of $148.

Education: 6% The education industry, the ninth-most targeted industry, is attractive to attackers due to the sensitive—and lucrative—nature of some emerging research projects, as well as the wealth of PII on students, faculty staff, and organizations associated with universities and schools. Researchers assess nation-state sponsored threat actors are those most likely to breach university networks, based on their motivation for attacking this sector, and their capability for doing so. Moreover, educational institutions do not typically boast a large in-house security team and may not have many security controls in place. They also control a large network of users who can easily bring in malware from personal devices or email. Aside from nation-states, educational institutions may be targeted by financial criminals looking to take over bursary accounts and student identities. Another relevant threat are hacktivists looking to champion a cause by holding an institute for ransom or threatening to release stolen data.

Energy: 6% Organizations in the energy sector are a prime target for cyber-attacks. To begin, they are the backbone of every country's critical infrastructure. Energy is central to the economic, national security, and day-to-day function of cities and industries.

III. Vulnerabilities and Threats in Manufacturing and Case Studies

In previous sections we have introduced the architecture and the operating principle of manufacturing systems in Industry 4.0. In this section, a number of Industry 4.0 cyber vulnerabilities and threats in information technology (IT), operation technology (OT) networks will be examined.

A. IT Network Threats

1) Threat Exposure of Manufacturing Networks Longer Equipment Life Cycles: This situation is most likely caused by a combination of a *"Do not touch a working system"* mentality and the long replacement cycle in hardware and software equipment. The problem here is, the software used to operate the hardware may no longer be supported, maintained, and updated, thus forcing equipment operators to use old operating systems to be able to continue running the equipment. For example: the Window XP still used in manufacturing, which its support ended in 2014.

2) Pervasiveness of Network Worms: One of the side effects of having old and unsupported operating systems in the manufacturing industry is the presence of a large number of unpatched vulnerabilities that could be exploited by old variants of network malware. It is no surprise that the detection of such malware families as Downad (aka Conficker), WannaCry (WCry), and Gamarue (Andromeda) are relatively high on machines used in manufacturing environments. Detections of worms in general and Downad in particular are significantly prevalent in the manufacturing industry. The proportion of malware types and families are illustrated in Figure 10 and 11.

3) Autorun: One of the common propagation methods of Downad and other USB worms is autorun.inf abuse, by which they can automatically execute whenever an infected removable

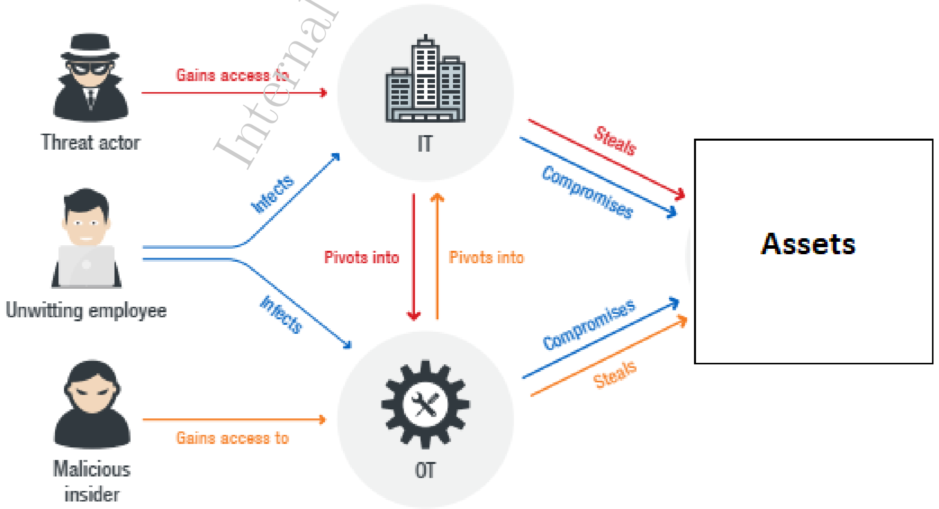


Fig. 9: How threats can figure into the IT, OT, and IP convergence [].

device is plugged in. Looking at the number of devices that detected autorun.inf, we observe that the manufacturing industry has significantly higher detections than other industries. This reflects the common practice in the industry of using USB drives to copy and transfer information between computers and networks (the IT and the OT) in a manufacturing environment. It is important to note that the famous Stuxnet malware [], which was designed to target a nuclear facility, was propagated using removable USB media, although the malware itself exploited a vulnerability in parsing shortcuts.

4) Targeted and Opportunistic Campaigns against the Manufacturing Industry: Manufacturing, like any other industry, suffers from both targeted campaigns and opportunistic hacking incidents. One of the recent incidents involving the PlugX malware attracted our attention.

Case study: PlugX is an advanced remote access tool (RAT) that is commonly used in targeted attacks for espionage or information exfiltrationSo when we identified a breach of a Chinese manufacturing company with the PlugX malware family, it appeared to us as unusual. There were several surprising facts about this series of incidents. For one thing, we rarely observe PlugX campaigns inside China. The hacker groups that use PlugX commonly focus on other countries and often infiltrate government and technology sectors.

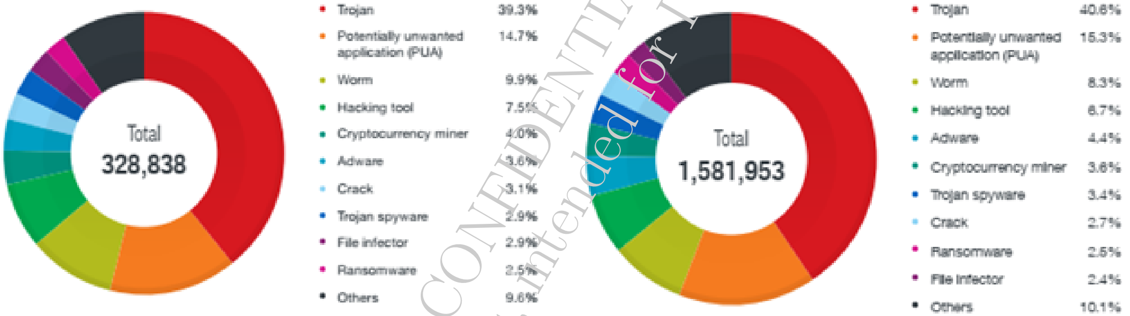


Fig. 10: Top malware types in the manufacturing (left) and in other industries (right) [].

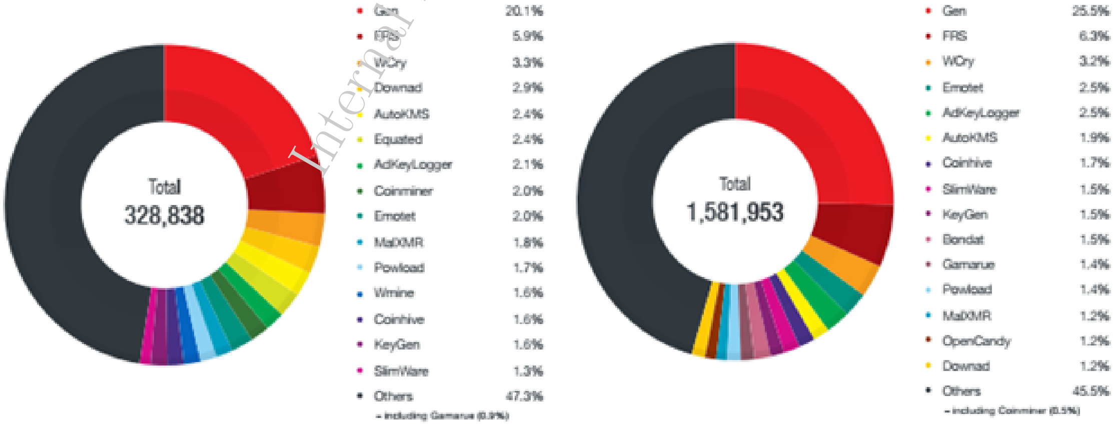


Fig. 11: Top malware families in the manufacturing industry (left) and in other industries (right) [].

B. *OT Network Threats*

1) ICS Vulnerabilities: The vulnerabilities come from modern manufacturing equipment, which has human-machine interfaces (HMIs) that allow operators and engineers to monitor and control the equipment, and programmable logic controllers (PLCs) are used to program logic into several pieces of equipment. Further, there are some industrial grade routers, hubs and gateways in the manufacturing networks.

According to the Industrial Control Systems Computer Emergency Response Team (ICS-CERT) as of September 2018, the number of vulnerabilities affecting manufacturing-related equipment jumped significantly in 2014 — a trend that continues to this day. Figure 10 and 11. show that the ICS vendors Siemens, Rockwell Automation, and Schneider Electric top the list, this is because these vendors have a wide range of products and the highest market shares in this industry.

2) Publicly Exposed ICSs: In some cases, attackers do not need to exploit any vulnerability in order to control or sabotage a critical manufacturing machine or production line. We have seen several cases where an HMI is directly exposed to the internet, without authentication. This basically allows anyone to tamper with values and issue commands on manufacturing machinery if the HMI is not read-only. Some of these interfaces provide read-only access and are used for monitoring purposes only, while others are not. Any unauthorized tampering of such systems can result in production delays, product contamination, physical hazards, or destruction of equipment.

Case: The risks to exposed critical infrastructures, specifically those in the energy and water industries, were identified in a report published by Trend Micro in 2018. Cyber - attacks against these industries could lead to supply disruption. For instance, operational disruption in a water facility could mean manipulated temperatures and supply of drinking water in an area. And power services to homes and businesses could be cut off by malicious actors.

3) Malware Targeting ICSs: The main goal of adversaries targeting the manufacturing industry is to gain access to ICSs for sabotage, control, or extortion. The increasing connectivity between OT and IT networks enables adversaries to pivot from a breach originating in the IT network to ICS devices in the OT network. Although uncommon, malware specifically designed to target ICSs has been seen before.

Case: Maroochy Shire sewage spill in Australia (March 2000, Australia) - The attacker

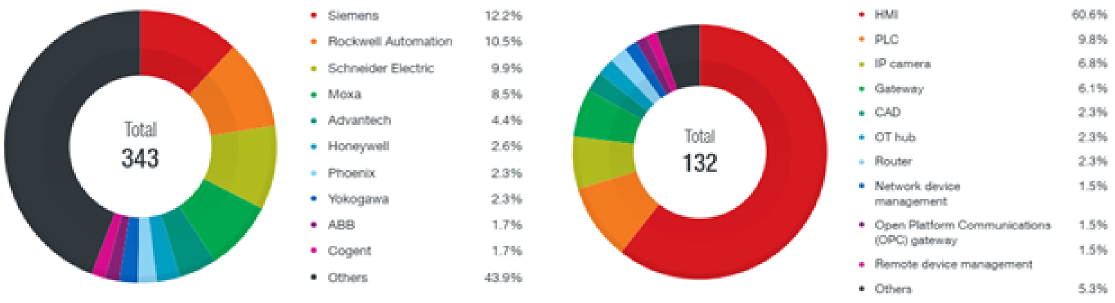


Fig. 12: Vendor distribution of vulnerabilities reported to the ICS-CERT (left) and equipment type distribution of 132 ICS/SCADA-related exploits on ExploitDB (right) [].

changed the electronic data using the stolen wireless radio, the SCADA controller, and the control software, and all operations failed. It leaded to release up to one million litres of sewage into the river and coastal waters of Maroochydore in Queensland, Australia (RISI 2015).

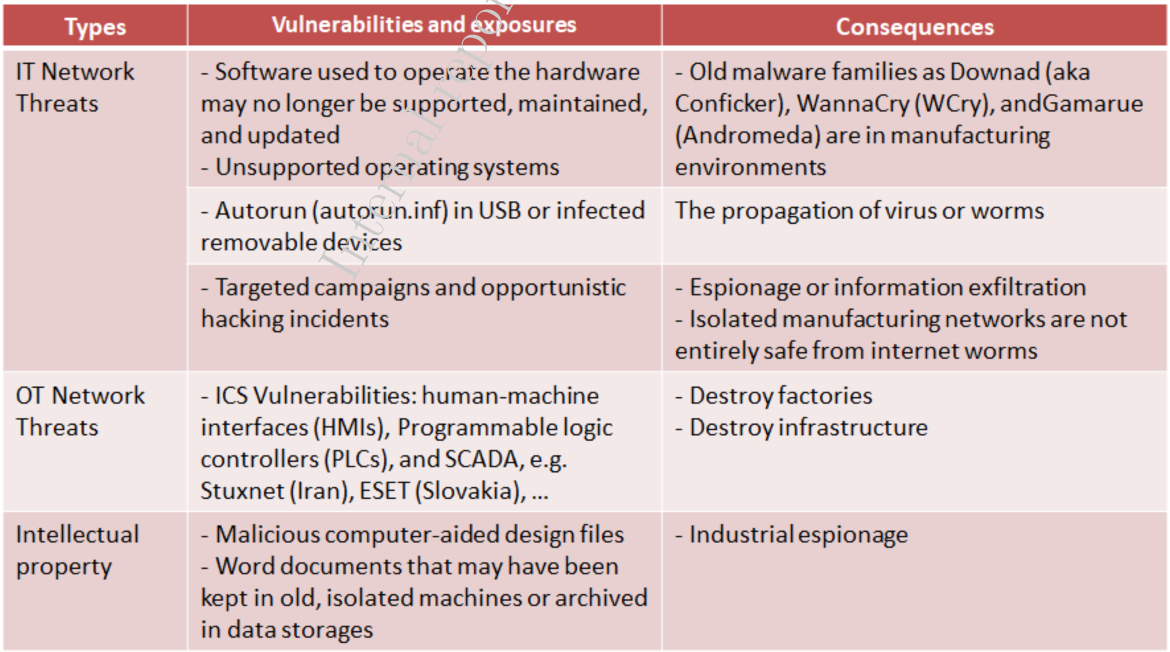
C. Intellectual Property (IP) Threats

One of the unique characteristic of the manufacturing industry is the presence of IP in digital form. Digital IP content is extremely important in Industry 4.0. The content can be a product design, a manufacturing process, or system information. There are some threats would lead to the leakage of proprietary information.

1) Malicious computer-aided design (CAD) Files: In the manufacturing industry, IP can take the form of a computer-aided design (CAD) file or a document file. CAD files serve as the digital blueprint for physical products, while document files often contain technical specifications, manufacturing processes, recipes, or inspection and quality assurance records. Both of these file types can be infected by viruses or trojanized to aid attackers in gaining access to critical machines.

Case: the CAD malware family called ACM_SHENZ.A, which was discovered in 2013, is known to weaken the security of infected computers for further attacks. The malware creates a user with admin privileges (which can be used later by an attacker to issue commands), creates writable network shares, and opens ports and services that have vulnerabilities.

2) Microsoft Word Macros: Unlike CAD files that contain product designs and information, Word documents in a manufacturing industry setting typically contain manuals and technical information such as parts lists, design specifications, or business-related matters such as order details, terms of service, and warranty information.



Types	Vulnerabilities and exposures	Consequences
IT Network Threats	- Software used to operate the hardware may no longer be supported, maintained, and updated - Unsupported operating systems	- Old malware families as Downad (aka Conficker), WannaCry (WCry), andGamarue (Andromeda) are in manufacturing environments
	- Autorun (autorun.inf) in USB or infected removable devices	The propagation of virus or worms
	- Targeted campaigns and opportunistic hacking incidents	- Espionage or information exfiltration - Isolated manufacturing networks are not entirely safe from internet worms
OT Network Threats	- ICS Vulnerabilities: human-machine interfaces (HMIs), Programmable logic controllers (PLCs), and SCADA, e.g. Stuxnet (Iran), ESET (Slovakia), ...	- Destroy factories - Destroy infrastructure
Intellectual property	- Malicious computer-aided design files - Word documents that may have been kept in old, isolated machines or archived in data storages	- Industrial espionage

Fig. 13: Main potential threats of manufacturing system

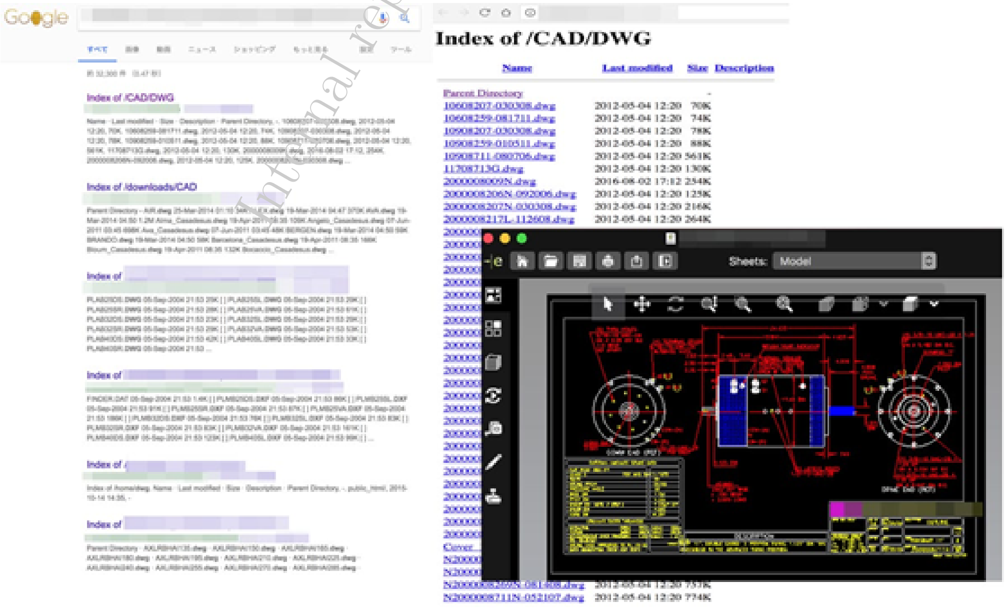
Document sharing between departments, vendors, and third parties also poses security risks in the form of information leakage and infected documents. Unauthorized sharing of documents can lead to information leakage. Document sharing is properly done only on corporate-approved channels for auditing and paper trail. This requirement is aimed at maintaining the ability to track with which a particular IP is shared and whether the sharing is authorized. If documents containing IP are shared through nonstandard channels, then the audit trail is lost and access policies cannot be enforced.

3) Unintentional Leaks Due to Poor Configuration: Poor security configuration can expose these proprietary design documents to the internet and lead to data leakage. Using simple open-source intelligence (OSINT) techniques, we were able to find CAD files that we believe were not supposed to be exposed to the public (Figure 14).

D. Black Market Related to the Industrial and Manufacturing Sectors

Although industrial cyberespionage has normally been the realm of advanced and persistent attackers, we have seen increasing cybercriminal interest in targeting the industrial and manufacturing sectors. Figure 15 shows an example of someone posting in a popular underground forum to solicit confidential information, including CAD and computer-aided manufacturing (CAM) files, source code, and confidential documents, for industrial espionage. In addition, ICS/SCADA-specific hacking tools being sold and advertised online. Figure 16 shows two examples of tools that can be used to crack or brute-force the passwords of PLCs. Although there are OT administrators or engineers who use these tools legitimately, the gray nature of these crackers makes it possible for attackers to read and modify the logic of hacked PLCs. Case: the National Police Agency of Japan issued in 2015 a notification regarding an observed increase in scanning activity related to PLCs.

It can be predicted that in a few years, industrial espionage will not only be performed by persistent attackers, but will also become commoditized with tools and services created and



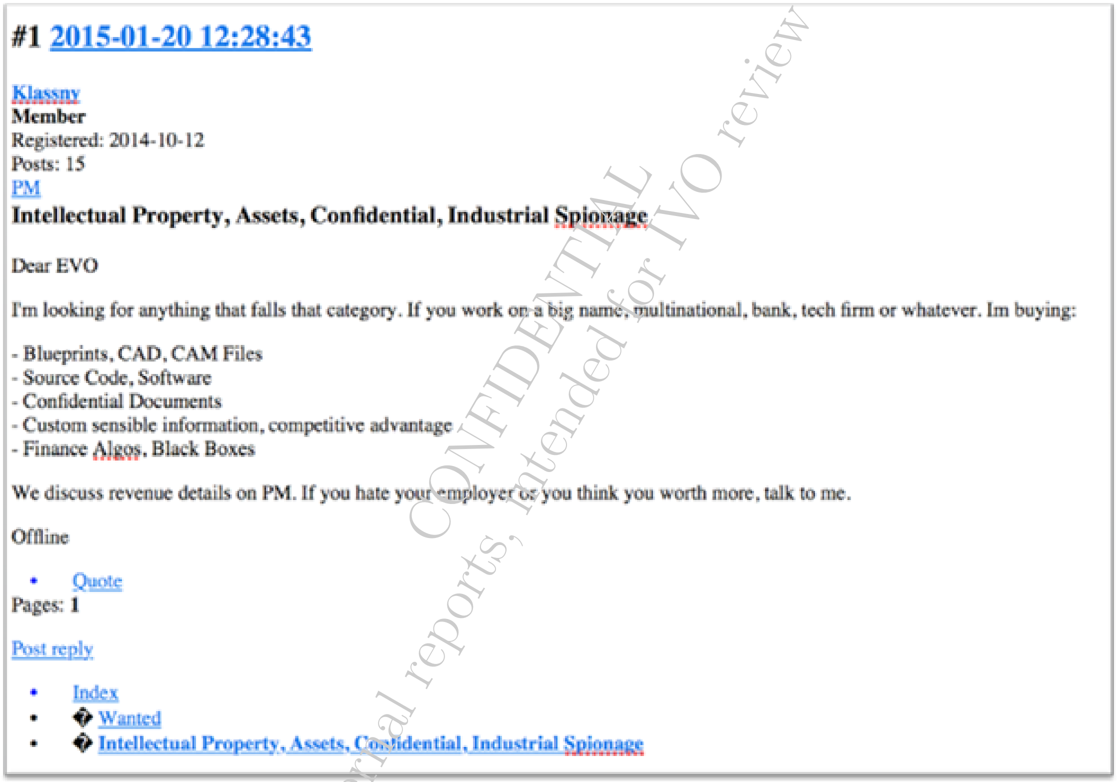
```
Index of /CAD/DWG

      Name                          Last modified      Size  Description

Parent Directory                                         -
10608207-030308.dwg             2012-05-04 12:20   70K
10608259-081711.dwg             2012-05-04 12:20   74K
10908207-030308.dwg             2012-05-04 12:20   78K
10908259-010511.dwg             2012-05-04 12:20   88K
10908711-080706.dwg             2012-05-04 12:20  561K
11708713G.dwg                   2012-05-04 12:20  130K
2000008009N.dwg                 2016-08-02 17:12  254K
2000008206N-092006.dwg          2012-05-04 12:20  125K
2000008207N-030308.dwg          2012-05-04 12:20  216K
2000008217L-112608.dwg          2012-05-04 12:20  264K
N2000008269N-081408.dwg         2012-05-04 12:20  757K
N2000008711N-052107.dwg         2012-05-04 12:20  774K
```

Fig. 14: Using open-source intelligence to open CAD files [].

offered by common cybercriminals.



#1 2015-01-20 12:28:43

Klassny
Member
Registered: 2014-10-12
Posts: 15
PM

Intellectual Property, Assets, Confidential, Industrial Spionage

Dear EVO

I'm looking for anything that falls that category. If you work on a big name, multinational, bank, tech firm or whatever. Im buying:

- Blueprints, CAD, CAM Files
- Source Code, Software
- Confidential Documents
- Custom sensible information, competitive advantage
- Finance Algos, Black Boxes

We discuss revenue details on PM. If you hate your employer or you think you worth more, talk to me.

Offline

- Quote

Pages: **1**

Post reply

- Index
- � Wanted
- � Intellectual Property, Assets, Confidential, Industrial Spionage

Fig. 15: Posting in an underground forum by a user looking for IP assets and other confidential information [].

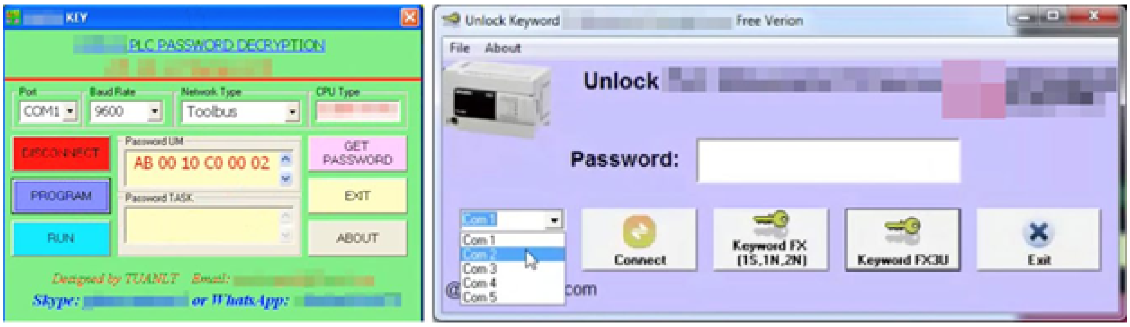


Fig. 16: Two examples of PLC password crackers sold online [].

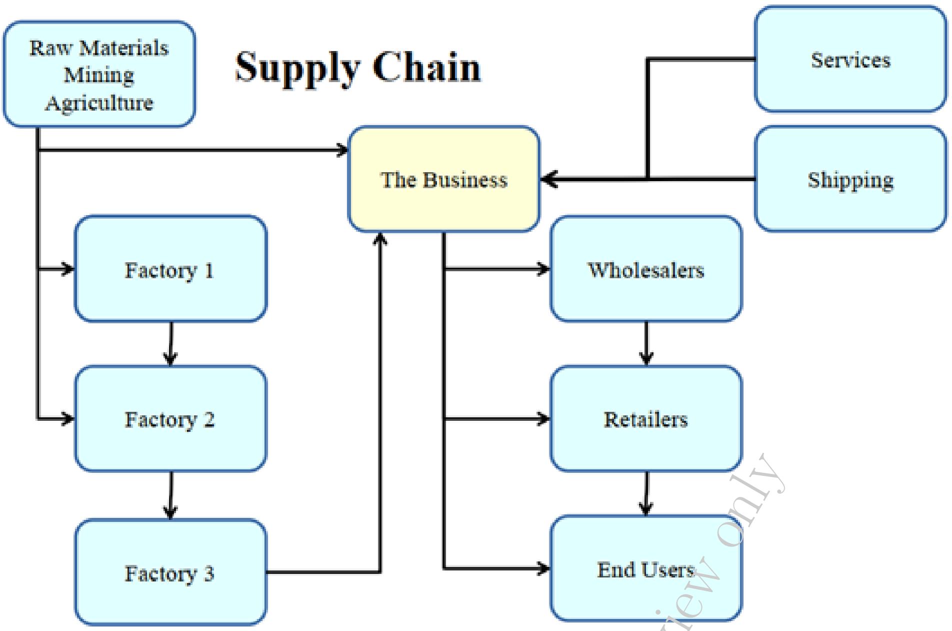


Fig. 17: Supply chain and cloud service [].

IV. Vulnerabilities and Threats from Third Parties

A. *Who are Third Parties?*

In general, third parties could be: Government (taxes, etc), Deliveries and Pick up services, Stationary suppliers, Internet provider, Web site developer, Cloud services, Recruitment agency, Marketing Agency, Customers, Accountant, Lawyer, Raw material, suppliers, Parts suppliers, Parts designers, Insurance Agent, Consultants, Maintenance, Cleaning services. In this report we will focus on supply chain and cloud service as described in Fig. 17 .

B. *Vulnerabilities and Threats from Third Parties - Cases in Industry 4.0*

When working in supply chain with third parties, we could face some threats

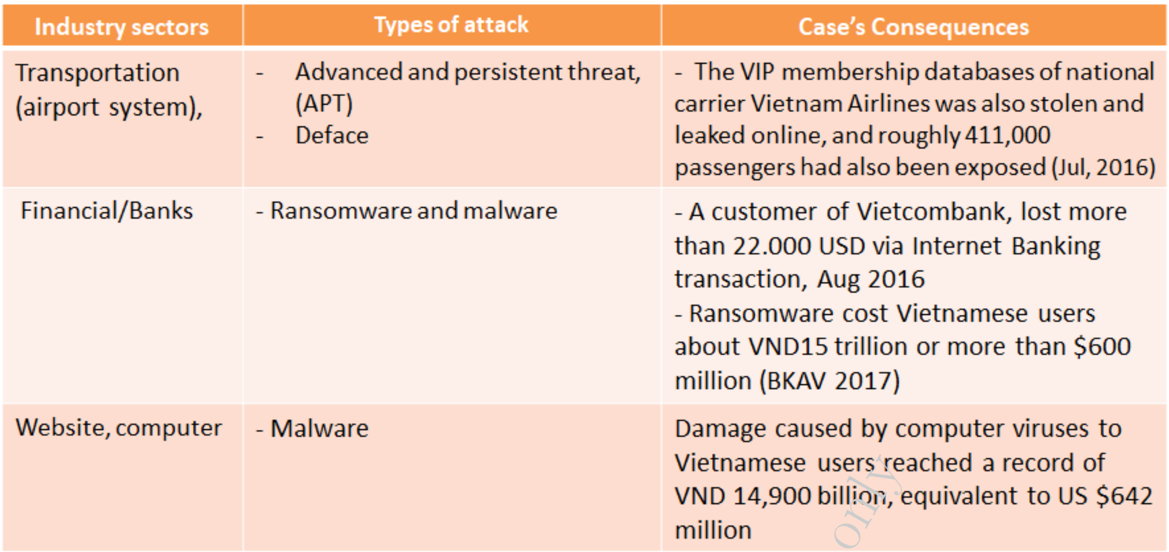
Threat 1- Shared Credentials. This is one of the most dangerous authentication practices we encounter in large organizations.

Threat 2- Irregular Access. Companies granting insider credentials to partnering companies must understand they are committing to a long and serious relationship. Managing and monitoring trusted outsiders could result in ongoing difficulties when trying to resolve whether an account has been compromised.

Threat 3-The Joint Cloud. Many companies are taking their first steps in deploying cloud-driven security solutions. While cloud-app usage regulation has received most of the attention, we are seeing more complex relations forming between our traditional environments and newly erected clouds, forming another under-addressed space. Looking forward, we suggest adopting cross-environment authentication protocols and measures that will enable more fine-grained monitoring over these evolving attack surfaces.

Threat 4- Public Internet Exposure. A device that is both connected to the Internet and enables third party remote access is an external attacker's prized desire

Threat 5- Proximity to Privileges. Privileged accounts provide both rogue insiders and malicious outsiders the access-level they need to approach sensitive resources securely and/or modify their own access-level.



Industry sectors	Types of attack	Case's Consequences
Transportation (airport system),	- Advanced and persistent threat, (APT) - Deface	- The VIP membership databases of national carrier Vietnam Airlines was also stolen and leaked online, and roughly 411,000 passengers had also been exposed (Jul, 2016)
Financial/Banks	- Ransomware and malware	- A customer of Vietcombank, lost more than 22.000 USD via Internet Banking transaction, Aug 2016 - Ransomware cost Vietnamese users about VND15 trillion or more than $600 million (BKAV 2017)
Website, computer	- Malware	Damage caused by computer viruses to Vietnamese users reached a record of VND 14,900 billion, equivalent to US $642 million

Fig. 18: Case studies in Vietnam

V. The situation in Asean and Vietnam

ASEAN is becoming one of the fastest growing region in the world with the population of about 634 million (over 100 million more than the European Union). It is the third most populous market in the world and with the combined GDP of more than 2.55 US dollars trillion making ASEAN the world's seventh largest economy (ASEANstats, 2018). A study by ATKearney indicating that digital economy could add 1 trillion USD to the GDP of the region which could boost the GDP of the region by 35. However, the "digital economy" which heavily relies on technology for business transactions opens an avenue for new threats such as online fraud, hacking and distribution of inappropriate materials. Deterring cybercrime is therefore, necessary for national security and protection of the information infrastructure. It is therefore a priority for legislators to adopt proper legislation to prevent the use of information and communication technologies for criminal activities. The challenge is for governments to make the use of the technology safer without minimizing the developmental opportunities. The governments establish cybersecurity scheme based on three specific goals: Ensuring open internet to promote Innovation; Combating Cybercrime; and Ensuring Privacy of their citizen.

In Vietnam, cybersecurity issues are in an alarming state. A series of targeted attacks on the airport system, banks, websites are typical evidence. Cybersecurity threatsin Vietnam are currently focusing on 4 types, including denial of service, phishing (information theft fraud), deface and malware.

Recently, these threats target on organizations, individuals, banks to steal sensitive information and also to extort. Besides, with the evolution of technology, information systems are faced with new threat stemming from artificial intelligence platforms. Particularly in 2016, 7.000 websites/web portals were attacked in Vietnam. A lot of devices connected with the Internet are exposed to security vulnerabilities that lead to the risk, allowing hackers to exploit and escalate privilege. On 29 July 2016, a hacker group launched an attack on the website of Vietnam Airlines with client information leaked and on-flight information screens at Vietnam's 2 biggest airports [4], Tan Son Nhat International Airport and Noi Bai International Airport.

Independent security expert Nguyen Hong Phuc said the hackers had shared three links leading to files that contain personal data of over 400,000 members of Vietnam Airlines' frequent passengers club, Golden Lotus. According to Mr. Phuc, this information may have fallen into the hands of the hackers four days before the attack. The hackers also targeted at the financial sector. Typically, in August 2016, a customer of Vietcombank, one of the biggest banks in Vietnam, lost more than 22.000 USD via Internet Banking transaction. On the next day, Viecombank's shares fell by VND 150,000 (6.7 USD) per share to VND 54,500 (2.45 USD) per share at the end of the session. The bank's market capitalisation therefore fell by VND 4 trillion (180 million USD). After that incident, the bank has made significant changes to its online banking policies in order to prevent similar incidents. According to the top online security firm BKAV, cyber-attacks including the rise of ransomware cost Vietnamese users VND12.3 trillion or more than 542.8 million USD in 2017. This year saw strong attacks from ransomware and malware containing cryptocurrency mining tools, causing losses that were more than 18 percent up from 2016. More than 1,900 computers in Vietnam were infected by the global WannaCry attack in May. WannaCry is a ransomware, which targeted computers running the Microsoft Windows operating system by encrypting data and demanding ransom payments in the Bitcoin cryptocurrency. Meanwhile, a cryptocurrency mining malware which appeared on Facebook has infected more than 23,000 computers in Vietnam. As cryptocurrencies became popular worldwide, hackers were prompted to launch attacks on computers to turn them into mining tools. In 2018, the damage caused by computer viruses to Vietnamese users reached a record of VND 14,900 billion, equivalent to US 642 US million dollar, 18 percent more than the damage of 2017. According to Bkav's research, more than 60 percent of agencies and enterprises in Vietnam are infected with malicious code. The main reason is that agencies and enterprises have not yet equipped with comprehensive antivirus solutions for all computers in the intranet. Therefore, as long as a computer on the network is infected with malicious code, all the other computers on the same network will be attacked and infected. In addition to slowing down the machine, the Cryptocurrency-Mining Malware also has the ability to update and download other malicious codes to erase data, steal personal information or even perform APT attacks.

The IoT devices deliver substantial benefits to end users, but also bring unprecedented security challenges. IoT devices typically possess low processing capabilities, limited memory and storage and minimal network protocol support. It is a significant challenge to design complex and comprehensive security measures. Using these weaknesses, in 2016, the first wave of IoT device attacks brought down the Internet. The Mirai Botnet hacked into some Internet of Things devices - in this case mainly routers and Internet Protocol (IP) cameras - and transformed the devices into botnets. The centrally-controlled IoT botnets flooded Dyn's, a Domain Name Services (DNS) provider [6], traffic causing a disruptive bottleneck that blocked Internet access for millions of users worldwide. Overall, IP addresses of Mirai-infected devices were spotted in 164 countries, such as Brazil, Vietnam, China.

References

[1] H. Well, “The 4th industrial revolution,” *Report*, 2018. [Online]. Available: https://kemptechnologies.com/

[2] J. Lee, B. Bagheri, and H.-A. Kao, “A cyber-physical systems architecture for industry 4.0-based manufacturing systems,” *Manufacturing letters*, vol. 3, pp. 18–23, 2015.

[3] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[4] A. Ustundag and E. Cevikcan, *Industry 4.0: managing the digital transformation*. Springer, 2017.

[5] Gartner, “Conn. stamford. gartner says by 2020, more than half of major new business processesand systems will incorporate some element of the internet of things, 2016.” [Online]. Available: http://www.gartner.com/newsroom/id/3185623

[6] IBM, “2019 ibm x-force threat intelligence index report.” [Online]. Available: https://www.ibm.com/security/

Vietnam National University, Hanoi
University of Engineering and Technology

Advanced Institute of Engineering and Technology

Technical Report

UET-AVITECH-2020002 November, 2020

Risk models for the security of Industry 4.0 systems

Tran Viet Khoa, Bui Minh Tuan, Dinh Thai Hoang,
Nguyen Linh Trung, Diep N. Nguyen, Nguyen Viet Ha,
Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Risk models for the security of Industry 4.0 systems

Tran Viet Khoa[1], Bui Minh Tuan[1], Dinh Thai Hoang[2], Nguyen Linh Trung[1], Diep N. Nguyen[2], Nguyen Viet Ha[1], and Eryk Dutkiewicz[2]

[1] AVITECH, VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[2] School of Electrical and Data Engineering, University of Technology Sydney, Australia

November, 2020

Abstract

The Industry 4.0 witnessed the era of technology developments with a sharply increase of Internet of Things (IoT) devices. But their evolution is challenged by cybersecurity problems which is one of the biggest obstacle of the technology developments. A number of different security systems have been developed to deal with cybersecurity problems, motivated by which we study various kinds of risk models to rank security systems. By reviewing two main risk analysis models – quantitative and qualitative–, we analyze each model and propose their applicability to ranking security systems in Industry 4.0.

Index Terms

Cyberattack detection, Industry 4.0, IoT, federated learning, deep learning, and cybersecurity.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. Introduction

Industry 4.0 creates a huge transformation in the world's manufacture, from manual labor to automation. In Industry 4.0, smart factories, houses, offices are connected to each others through the Internet to promote automation, improve productivity, reduce labor working time and improve the quality of human life. The Internet of Things (IoT) devices (e.g., sensors, cameras) play an important role for the automation of smart systems in Industry 4.0. According to [1], [2], the number of smart devices increases dramatically and can reach up to 50 billion devices in 2020, nearly 6 times as much as the number of people in the Earth. IoT devices have limited computing and energy resources, and thus they are vulnerable by cyberattacks. In Industry 4.0, with a large number of devices connected to the Internet, if they were attacked and under-controlled by a botnet, the botnet would have more power than in conventional systems and could launch large-scale attacks to any victims in the cyberworld.

Along with the tremendous growth of cyberattacks, on the other side of the front line, various kinds of security systems have been developed to prevent attackers. Although there are a number of modern security systems that can strongly protect themselves from cyberattacks, many private databases and core servers were still shut down by normal attacks. This is because the design and deployment of security systems is not uniform, with the lack of security knowledge of designers or operators.

Motivated by this problem, we would like to analyze the risk models to find the best approach, and then to create a novel model that can analyze and hence support the security systems by ranking the threats from low to high. In this report, three kinds of analyses were reviewed namely quantitative, qualitative and mixed quantitative-qualitative risks analysis. Also, the applicability of each method to ranking security systems is discussed. This work aims to create a risk model that leverages Industry 4.0 cyber-security threats with the input coming from the cybersecurity threats of manufacturing in Industry 4.0.

II. Risk Analysis

The National Research Council [3] defines risk analysis with three core elements, namely: risk assessment, risk management, and risk communication. The interaction and overlap of three elements are described in Fig. 1 [4].

Risk assessment is the process that measures the frequency of a loss of a system and the magnitude of the loss (consequence). Risk estimates, evaluates, controls and minimizes the potential (likelihood or frequency) of magnitude risk. Risk communication provides the information about the nature of risk (expected loss) and consequences. Risk communication supports the exchange and discussion of risk assessment approach and risk management options between the decision makers and other stakeholders.

Risk analysis is the way to estimate the potential and magnitude of loss to control and minimize it. There are two approaches for risk analysis depending on data. If data (such as losses ...) are sufficient, the risk can be directly estimated from the actual loss. In this case the data could be the frequency and loss of DoS (Denial-of-Services), botnet attacks. If data are insufficient, the loss is “modeled” to estimate the potential loss and the analysts have

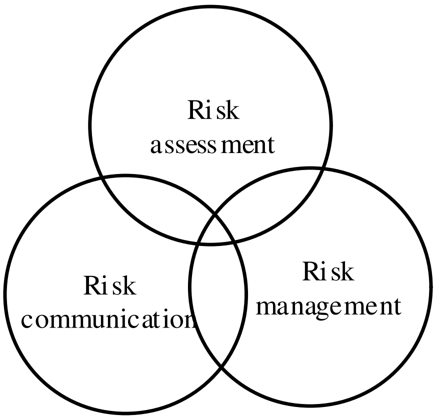


Fig. 1: Elements of risk analysis.

to model and predict the risks. Risk analysis attempts to measure the magnitude of a loss (consequence) by complex systems, including evaluation, risk reduction, and control policies. Generally, there are three types of risk analysis namely quantitative, qualitative, and a mix quantitative-qualitative. All these methods are widely used with different purposes, strengths, and weaknesses.

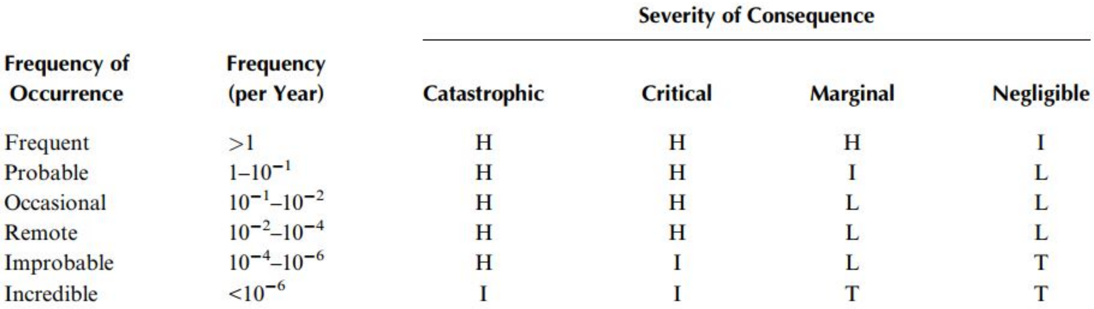
A. Quantitative Risk Analysis

Quantitative risk analysis contains the methods to estimate the risk through the probability of a loss and make decision through the probability. When data are insufficient, the uncertainties associated with the quantitative results play a decisive role in the use of the results. Quantitative analysis is the better choice then others when the data or evidences are insufficient to estimate the probability (or frequency) and magnitude of the losses. This method is widely used in recent years, because availability of quantitative techniques and tools, and our ability to make quantitative estimation from limited data. Nevertheless, this method is restricted to use in a large scale because of the complexity, expensive and time-consuming properties.

B. Qualitative Risk Analysis

This type of risk analysis may be most widely in use because of the simple and quick performance. In qualitative analysis, the potential loss is measured and estimated by a scale such as high, low and medium. In this type, the result is a qualitative matrix which is formed by the frequency, the magnitude of loss. Then, this matrix is used to make policy and risk management decisions. This analysis is much easier and simpler to use and understand because it does not require gathering precise data. The rank-ordered approximations in this approach are sufficient and quickly estimated. Rank-ordered approximations of probability and consequence can have useful approximations of risk. For example, one approximation scale defines probability from high to low. Qualitative probability categories and their accompanying definitions are frequent, probable, occasional, remote, improbable and incredible.

Similarly, consequence can be defined in descending order of magnitude. Table 2 describes an example of the risk values associated with each frequency–consequence category. Quali-



Frequency of Occurrence	Frequency (per Year)	Severity of Consequence			
		Catastrophic	Critical	Marginal	Negligible
Frequent	>1	H	H	H	I
Probable	$1–10^{-1}$	H	H	I	L
Occasional	$10^{-1}–10^{-2}$	H	H	L	L
Remote	$10^{-2}–10^{-4}$	H	H	L	L
Improbable	$10^{-4}–10^{-6}$	H	I	L	T
Incredible	$<10^{-6}$	I	I	T	T

Fig. 2: Qualitative risk assessment matrix.

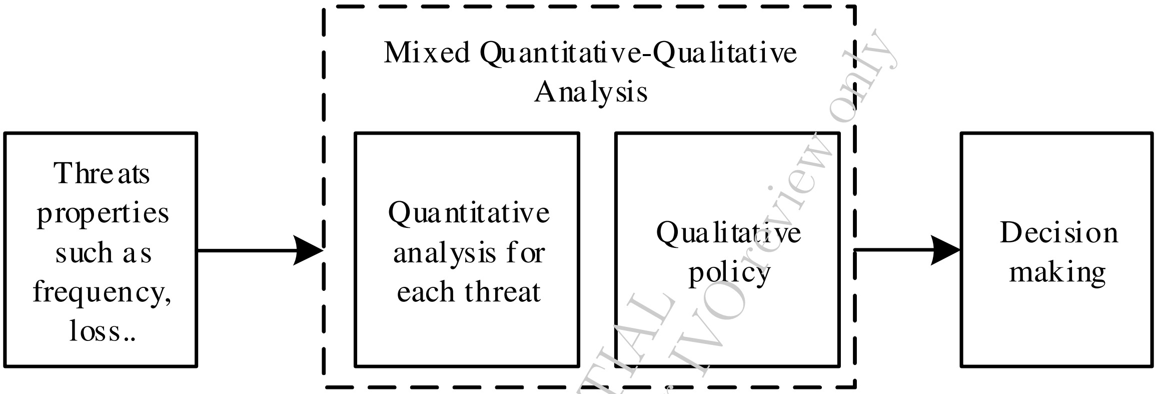

Mixed Quantitative-Qualitative Analysis
Threats properties such as frequency, loss..
Quantitative analysis for each threat
Qualitative policy
Decision making

Fig. 3: Risk model for cyber attack detection.

tative risk analysis is the method for simple systems such as a single cyberattack detection, simple physical security.

C. Mixed Quantitative-Qualitative Risk Analysis

Risk analysis can use a mix of quantitative and qualitative analyses. The mix can be performed in two ways: measure the frequency or potential loss quantitatively and measure the consequences quantitatively, or vice versa. Furthermore, it is possible to measure the frequency and magnitude of the loss quantitatively, but using qualitative methods for decision making. Also, the quantitative risk values may be increased by other quantitative or qualitative risk information to make a decision. This is the method for the U.S. Nuclear Regulatory Commission's new regulatory paradigm called "risk-informed" regulation. In this case the risk information from probabilistic risk assessment (PRA) are cooperated with other quantitative and qualitative results obtained from deterministic analyses and engineering judgments to set regulatory decisions and policies.

III. Risk model for the cyber-security of industry 4.0

The industry 4.0, which is much more different than other sectors, is distributed with a hug number of Internet of Thing (IoT) devices directly connect to the Internet. So the security system to protect them have to ensure to face with the highest threat. But how can we know about the highest or lowest threat? Our applied model would analyze the threats and answer this question.

After carefully analyzing the advantage and disadvantage of different risk analysis in precious section, in this section we would like to discuss about the risk assessment model which is the main element of risk assessment model could be applied to classify the threats in cybersecurity of industry 4.0.

Although quantitative analysis can provide the accurate results after calculation, they need a complex system with a lot of time-consuming to analyze the risks. This scenario seems inappropriate for IoT systems which contain a large number of light-weight devices and require real-time processing. Qualitative analysis seems to overcome the disadvantages of quantitative model with fast and simple properties but they still need to improve substantially in the accuracy of evaluation. In Industry 4.0, if the threats are misplaced in classification, it would cause many serious problems such as increasing the risk for systems, wasting time and money for system design and implementation. In this scheme, we choose the mixed quantitative and qualitative analysis for the risk model of cybersecurity of Industry 4.0. The mixed quantitative-qualitative analysis is described in Fig. 3.

Fig. 3 described our research model. The threats with frequency and loss properties of manufacturing in Industry 4.0 are treated as inputs of the model. The quantitative analysis is the first component in the model receiving the input information. This quantitative analysis will calculate and predict the line of different threats as described in Fig. 4. After that, the qualitative policy will mark and classify the area between two lines corresponding to its properties as in Fig. 4 such as low, medium, high, ... Finally, the decision masking will encourage appropriate actions for each policy to reduce the risk of systems.

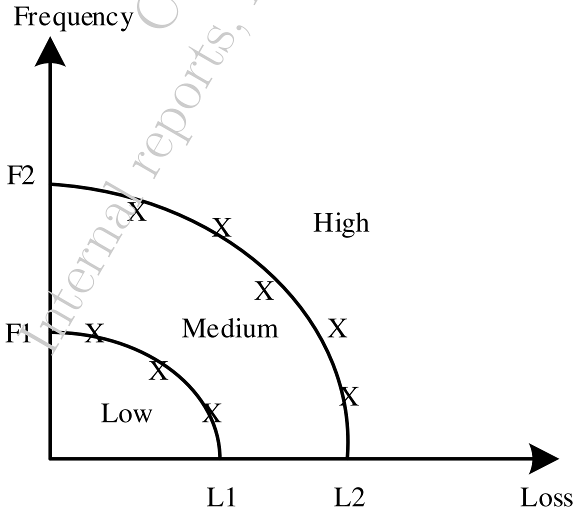


Fig. 4: Risk model for cyber attack detection.

IV. Conclusion and Future Plan

In this report, we have briefly looked the advantages and disadvantages of the different kinds of risk analyses. We choose the mixed quantitative-qualitative risk analysis model as the most appropriate model for cybersecurity of Industry 4.0. In future work, we intend to implement the real model by analyzing the input threats of manufacturing of Industry 4.0 and compare the evaluation results of the real model with theory to adjust them so as to have the most applicable model for cybersecurity in Industry 4.0.

References

[1] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with internet of things: A tutorial introduction," *IEEE Design & Test*, vol. 33, no. 2, pp. 76–96, 2016.

[2] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *CISCO white paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[3] N. R. Council *et al.*, *Risk assessment in the federal government: managing the process*. National Academies Press, 1983.

[4] M. Modarres, *Risk analysis in engineering: techniques, tools, and trends*. CRC press, 2006.

Vietnam National University, Hanoi
University of Engineering and Technology

Advanced Institute of Engineering and Technology

Technical Report

UET-AVITECH-2020004 November, 2020

Transfer learning model for cyberattack detection systems

Tran Viet Khoa, Dinh Thai Hoang, Nguyen Linh Trung,
Diep N. Nguyen, Nguyen Viet Ha, Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Transfer learning model for cyberattack detection systems

Tran Viet Khoa[1], Dinh Thai Hoang[2], Nguyen Linh Trung[1], Diep N. Nguyen[2], Nguyen Viet Ha[1], and Eryk Dutkiewicz[2]

[1] AVITECH, VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[2] Electrical and Data Engineering, University of Technology Sydney, Australia

November, 2020

Abstract

In Industry 4.0, the number of Internet-of-Thing (IoT) devices connecting to internet have been increasing dramatically. However, their growth has to face security problems which directly challenge their development. While various solutions were introduced to deal with cyberattack detection, their performance and complexity still need to enhance. We are motivated by the challenges of enhancing the accuracy while detecting cyberattacks in distributed environments such as IoT networks. In this report, we first review the challenges of cyberattack detection and the different methods to solve them. Then, our approach to deal with these challenges was proposed via the use of a transfer learning model. Through experimental results, we demonstrate that this model could enhance the system performance and complexity by increasing the accuracy in comparison with non-distributed deep learning method.

Index Terms

Cyberattack detection, Industry 4.0, IoT, federated learning, deep learning, and cybersecurity.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. Introduction

In Industry 4.0, an enormous number of industrial devices connect to each others through the internet to create smart factories, systems, as well as intelligent manufacturing and management ecosystems [1]. The uncountable number of industrial internet of things (IoT) devices (e.g., sensors, cameras, smart devices) play an important role for the automation of smart systems in these ecosystems. However, with a numerous devices, having light-weight properties, connected to the Internet by open IP addresses, these smart systems create more avenues for attackers to perform cyberattacks. Not only manufacturing, according to [2], government and financial services sectors are alse threatened by cyberattacks. In these sensitive factors, the effect of cybercrimes does not stop at the border of the cyberworld, it could widely affects to our daily life.

To prevent the damage of cyberattacks in the cyberworld, it is essential to develop methods for analyzing data, in order to figure out as early as possible the attacks before they seriously damage systems. Various solutions were proposed to deal with cyberattacks. The authors of [3] proposed methods to identify and prevent DoS attacks. Furthermore, alternative techniques based on game theory [4] and supervised learning [5] methods were proposed. Nevertheless, these methods are not ready to apply in real-time mobile cloud environments and their performance in terms of accuracy still needs to improve. To deal with this problem, the authors of [6] introduced a framework based on the Deep Belief Network to enhance relatively the accuracy in detecting various kinds of cyberattacks in mobile cloud environments. Deep learning is a very promising solution to detect cyberattacks with high accuracy.

However, in Industry 4.0, the data are usually separated in various local places so each place does not have enough data to train a deep learning model. This problem affects dramatically the accuracy of cyberattack detection. In this case, we have to send all data to a central server to be aggregated before training. This scheme has to face various issues such as network congestion and privacy when the local data are sent throughout the network and the complexity when a deep learning mechanism spends a great amount of time to train the aggregated data. In [7], the authors introduced federated learning (FL) for distributed data environments. FL is an algorithm that permits the models in use at different local places could exchange their learning knowledge while it does not have to expose their data. In a recently survey [8], the authors analyzed the techniques that apply FL to mobile edge networks. In the recent paper about cybersecurity, the authors of [9] proposed to apply FL to detect cyberattacks in fox-to-thing computing.

In [10], we proposed a collaborative learning model to detect cyberattacks in Industry 4.0 for classification and anomaly detection. This model could detect various kinds of cyberattacks for Industry 4.0 systems in different environments such as mobile cloud computing and IoT devices with more effective performance and low complexity than conventional machine learning techniques. In this environment, each local system analyzed its data by its local deep learning model and updated gradients to other local systems in order to create an aggregated model. Through experimental results, we demonstrated that this model could enhance accuracy, reduce complexity in comparison with conventional machine learning techniques while avoiding network congestion and preserving data privacy for the whole system.

Although this collaborative learning model in [10] can improve the accuracy of cyberattacks detecting by exchange their learning knowledge, it can work well only when their datasets have the same properties. Can the local systems exchange their knowledge when their datasets have different properties? In [11] the author proposed Federated Transfer Learning (FTL) which combined FL with techniques of transfer learning to permit the models exchange their learning knowledge while learning from datasets with different properties. Although FTL permits the model to exchange the knowledge to improve the accuracy, while not having to expose the local data across the network, the authors of [12], [13] demonstrated that the information from local datasets could be leaked from exposing gradients. In [11] the authors proposed an improvement of FTL which enhances the security while transferring parameters across the network. Inspired by the secure FTL in [14], we proposed to apply this model to cyberattack detection in Industry 4.0 in order to improve the accuracy through exchanging knowledge while learning from various sources. Through simulation results, we show that our proposed approach can improve the accuracy compared with those of other conventional machine learning techniques.

II. System Model

A. Network Architecture

We introduce the network architecture in Fig. 1. Each Local Model (LM) has its own neural network and runs the same deep learning algorithm. We denote $\mathcal{L} = \{1, \ldots, l, \ldots, L\}$ to be the number of LMs that cooperate in this model and $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_l, \ldots, \mathcal{T}_L\}$ as the set of training datasets of the LMs. Each LM is a deep learning machine which is trained and tested with its local dataset.

The raw data containing both attack and normal behaviors is trained by the autoencoder deep learning model. After completing the training process, each LM exports their update parameters and sends them to the Center Point (CP). The CP plays an aggregation role, accumulating all update gradients and update to other LMs. Therefore this model creates a cooperative environment for all LMs to train and exchange their learning machines to increase the accuracy of the whole system.

B. Cyberattack Detection System With Secure Federated Transfer Learning Model.

To improve the efficiency of cyberattack detection in IoT Industry 4.0, the secure federated transfer learning model should be applied in this situation is described in Fig. 2. The raw data from datasets A and B are first used to train the autoencoder learning models. After training, the parameters such as weight, bias, loss, gradient are encrypted by the asynchronous encryption key which was generated from third-party. Then, the encrypted parameters is sent to the center point for aggregation and creating a global model. Finally, based on the trained global model, each local model will be updated from the global model. In this way, a LM can “learn the knowledge” from other LMs without the need of sharing raw datasets.

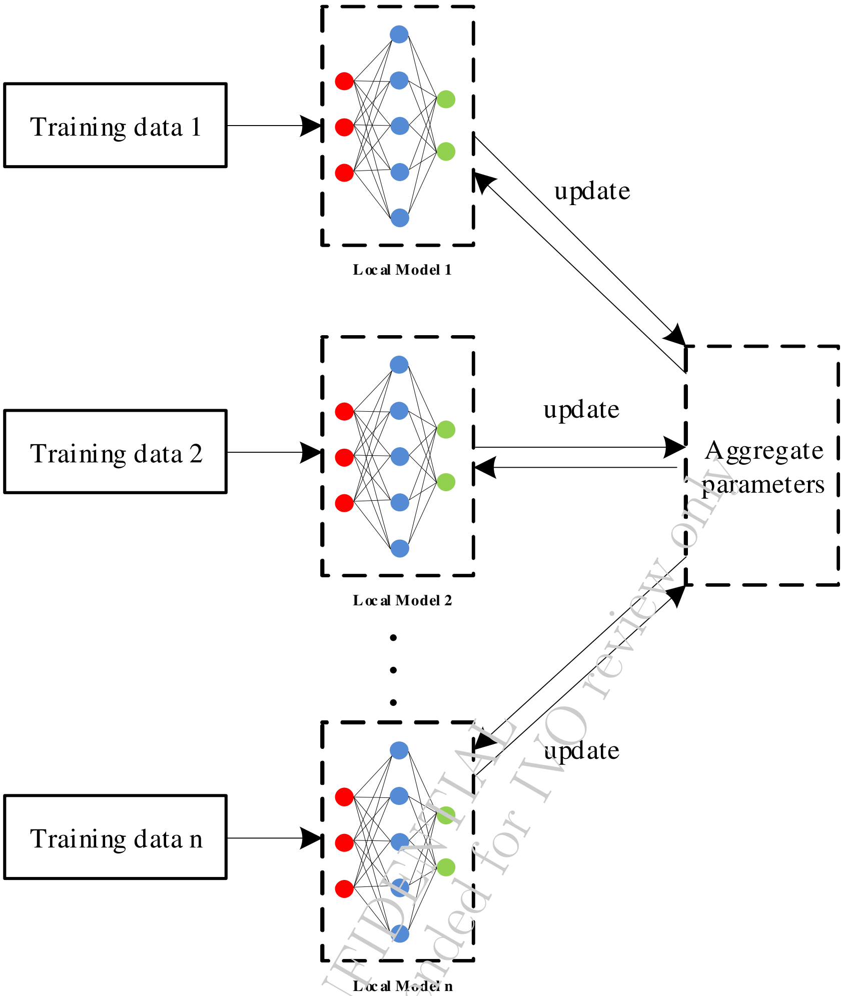


Fig. 1: Cooperative model of cyberattack detection.

III. Federated Transfer Learning-based Cyberattack Detection Model

In this section, we discuss the algorithm that we apply for cybersecurity in IoT Industry 4.0. This method is applicable to predict and identify the behavior of incoming packets for the cyberattack detection system. We assume that we have two parties A and B with datasets $D_A = x_i^A, y_i^A$ and $D_B = x_i^B, y_i^B$. Each party has it own neural network called Net^A and Net^B. The outputs of two neural networks are $u_i^A = Net^A(x_i^A)$ and $u_i^A = Net^A(x_i^A)$ and the prediction functions $\varphi(u_j^B) = \varphi(u_1^A, y_1^A, \ldots, u_{N_A}^A, y_{N_A}^A, u_j^B)$. The purpose of training is to minimize the loss function using the labeled dataset:

$$\arg \min_{\Theta^A, \Theta^B} L_1 = \sum_{i}^{N_c} l_1(y_i^A, \varphi(u_i^B)), \tag{1}$$

where Θ^A, Θ^B are the training parameters of Net^A and Net^B, N_c is the overlapping dataset between A and B, and l_1 represents the loss function. Besides, we also want to minimize the

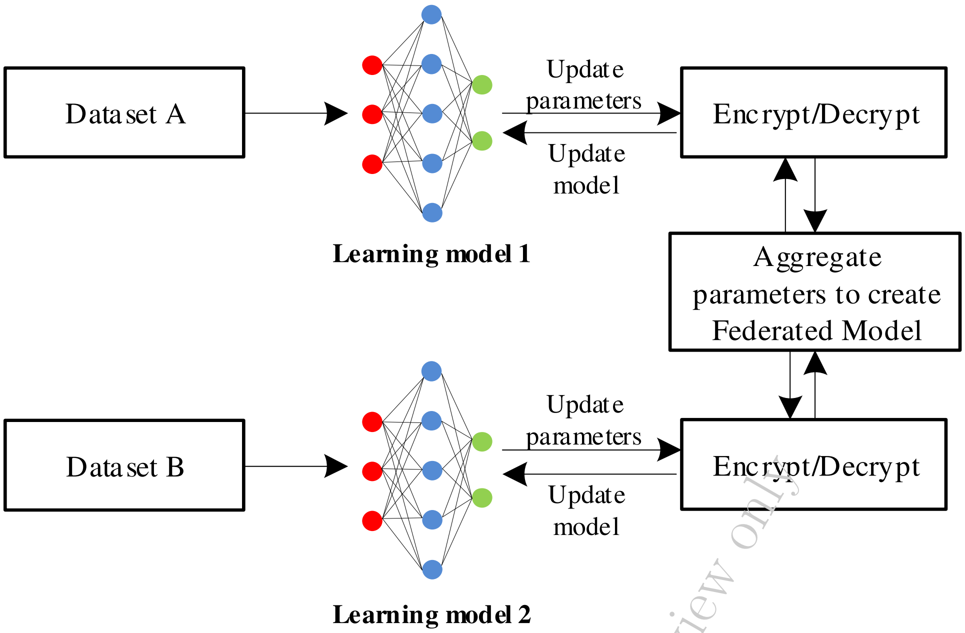


Fig. 2: Cyberattack detection system with secure federated transfer learning model.

alignment loss function between A and B:

$$\arg\min_{\Theta^A,\Theta^B} L_2 = -\sum_{i}^{N_{AB}} l_2(u_i^A, u_i^B), \tag{2}$$

where l_2 represents the alignment loss function. The last function that needs to be minimized is:

$$\arg\min_{\Theta^A,\Theta^B} L = L_1 + \gamma L_2 + \frac{\lambda}{2}(L_3^A + L_3^B), \tag{3}$$

where γ and λ are the weight parameters, L_3^A and L_3^B are the regularization terms. The gradients for updating Θ^A, Θ^B are calculated by the following formula:

$$\frac{\partial L}{\partial \theta_l^i} = \frac{\partial L_1}{\partial \theta_l^i} + \gamma \frac{\partial L_2}{\partial \theta_l^i} + \lambda \theta_l^i. \tag{4}$$

The dataset of each party is equally divided into three parts for training, prediction and testing. The algorithm in Algorithm 1 describes the training process of the algorithm to deal with the previous objectives. Let us denote $[[.]]_A$ and $[[.]]_B$ the homomorphic encryption with public keys A and B, respectively. After initializing Net^A and Net^B to get u_i^A and u_i^B, party A computes and encrypts $\{h_k^A(u_i^A, y_i^A)\}_{k=1,\dots,K_A}$ and sends them to B to support the calculation of the gradients of Net^B. Similarly, party B computes and encrypts $\{h_k^B(u_i^B, y_i^B)\}_{k=1,\dots,K_B}$ and sends them to A to support the calculation of the gradients of Net^A and the loss L. To protect the gradients from exposing across the network, A and B add a mask to each gradient with an encrypted random value. After that, A and B send the encrypted masked gradients and losses to each other and received the encrypted values. A and B then unmasks the gradients and losses and update their parameters on each iteration. The prediction process accompanies with the training process on every iteration to optimize the parameters and minimize the loss. Finally, testing is taken to check the accuracy and the performance of the whole algorithm.

Algorithm 1 Secure Federated Transfer Learning

```
Input: learning rate $\eta$, weight parameter $\gamma\lambda$, max iteration $m$, tolerance $t$;
Output: model parameter $\Theta^A, \Theta^B$
A, B initializes $\Theta^A, \Theta^B$ and creates an encryption key pair, respectively and sends public key to each other;
$iter = 0$
while $iter \leqslant m$ do
    A do:
    $u_i^A \leftarrow Net^A\{\Theta^A, x_i^A\}$ for $i \in D_A$;
    computes and encrypts $\{h_k^A(u_i^A, y_i^A)\}_{k=1,2...K_A}$ and sends to B;
    B do:
    $u_i^A \leftarrow Net^B\{\Theta^B, x_i^B\}$ for $i \in D_B$;
    computes and encrypts $\{h_k^B(u_i^B, y_i^B)\}_{k=1,2...K_B}$ and sends to A;
    A do:
    creates random mask $m^A$;
    computes $[[\frac{\partial L}{\partial \theta_i^A} + m^A]]_B$ and $[[L]]_B$ and sends to B;
    B do:
    creates random mask $m^B$;
    computes $[[\frac{\partial L}{\partial \theta_i^B} + m^B]]_A$ and $[[L]]_A$ and sends to A;
    B do:
    decrypts $\frac{\partial L}{\partial \theta_i^A} + m^A, L$ and sends to A;
    A do:
    decrypts $\frac{\partial L}{\partial \theta_i^B} + m^B, L$ and sends to B;
    B do:
    update $\theta_l^B = \theta_l^B - \eta\frac{\partial L}{\partial \theta_i^B}$;
    A do:
    update $\theta_l^A = \theta_l^A - \eta\frac{\partial L}{\partial \theta_i^A}$;
    if $L_{prev} - L \leqslant t$ then
        Send stop signal to B;
        Break;
    else
        $L_{prev} = L$;
        $inter = iter + 1$;
        continue;
```

IV. Simulation results

A. Dataset

In this simulation, we use the NSL-KDD dataset [15] to evaluate the performance of the proposed method and compare with another baseline method, i.e., the centralized learning model for classification [6]. For the baseline method, the Central Server first needs to collect datasets from all the parties and then performs the machine learning algorithms to detect the normal and malicious packets. For the proposed method, we distribute the dataset into different parties for the local training process.

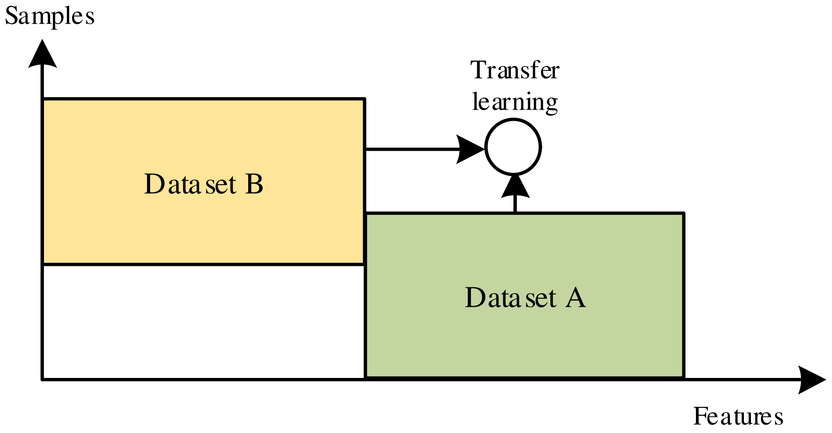


Fig. 3: Federated Transfer learning.

The NSL-KDD dataset [15] was built by the cybersecurity group at the University of New Brunswick, Canada. This dataset, which collected from the network, includes 41 features such as service, protocol types, duration, flag, source bytes, destination bytes, ... 24 types of attacks in the training dataset and 38 types of attacks in the testing dataset. The types of attacks are categorized into 4 groups including denial-of-service (DoS), attack from remote to local machine (R2L), unauthorized access to local administrator user (U2R), and probing attack. Although this dataset contains the same properties of the KDD dataset, it eliminates many drawbacks of the KDD dataset including removing any duplicate samples in the dataset such that all records in both training and testing datasets are unique and providing better proportion of training and testing datasets. In this simulation, we aim to identify the attack in the network, so we set the label for attack and normal behavior are 1, -1 respectively.

To ensure randomization of data, two FTL datasets A and B are extracted from random samples from the NSL-KDD dataset. As can be seen from Fig. 3, two selected datasets have the same number of samples but are different in features. As described in the secure federated transfer learning algorithm, the datasets would be trained by their own neural network while transferring and updating their parameters to the other under secure condition.

B. Evaluation Methods

As mentioned in [16], [17], the confusion matrix is typically used to evaluate the performance of system, especially machine learning system. We denote TP, TN, FP, and FN to be “True Positive”, “True Negative”, “False Positive”, and “False Negative”, respectively. Then, to evaluate the performance of the algorithm by Area under the curve (AUC) of the Receiver Operator Characteristic (ROC) curve is described in Fig. 4. The ROC curve is the graph that is created with FP and TP as x and y axes while the AUC is the area under the curve that demonstrates the probability of the positive is higher than the negative:

$$A = \int_{x=0}^{1} TP(FP^{-1}(x))\, dx. \tag{5}$$

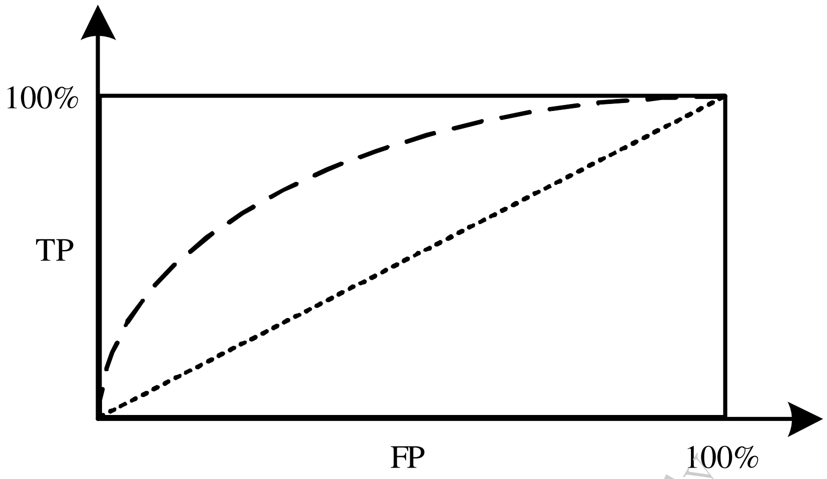


Fig. 4: ROC curve and AUC.

TABLE I: The performance comparison of FTL and Centralized deep learning method.

Samples	1500		3000		15000	
Times	Central	FTL	Central	FTL	Central	FTL
1	91.92	**98.67**	93.44	**96.25**	95.48	**98.34**
2	94.73	**98.58**	95.14	**97.08**	95.57	**97.95**
3	94.55	**98.8**	95.19	**96.85**	95.44	**98.29**
4	92.89	**98.84**	93.64	**97.03**	95.83	**98.23**
5	91.48	**99.16**	93.66	**96.71**	94.93	**98.49**
6	93.27	**98.8**	93.72	**96.54**	95.27	**98.54**
7	93.07	**98.84**	93.41	**96.8**	95.27	**97.56**
8	91.31	**99.07**	92.76	**96.87**	95.33	**97.99**
9	92.11	**99.2**	92.54	**96.81**	95.22	**98.31**
10	93.33	**99.2**	93.72	**96.86**	95.41	**98.28**

C. Performance Evaluation

In this section, we compare the performance of FTL and the centralized deep learning method in term of AUC score, privacy and communication overhead. To ensure the nature of data, the dataset of the centralized deep learning method is randomly selected from nearly 150 thousand samples of the NSL-KDD dataset. This dataset is then divided into three parts for training, optimization and testing. The sample procedure is taken to the dataset of FTL except we need to selected two datasets for parties A and B. Table I and Fig. 5 show the detailed calculation results of the two algorithms. As can be seen in Fig. 5, although the AUC of the centralized deep learning algorithm slowly grows up when the number of samples increases, the lines of FTL show the stable state and outperform than centralized deep learning in any situation. Especially with 1500 samples, FTL can improve the accuracy up to 7%.

In addition to improving the accuracy of cyberattack detection, FTL can significantly reduce the network traffic in the whole system. Instead of sending all the local data to a center point then aggregating to a dataset as in the centralized deep learning method, FTL only transmits network parameters such as gradients and loss throughout the network. In this situation, FTL not only mitigates the network traffic but also preserves the privacy of the local data. Furthermore, to prevent the leaked data from exposing gradients, the secure FTL method with homomorphic encryption with public and private keys is implemented. This can be considered as the second guard to ensure the security of the system.

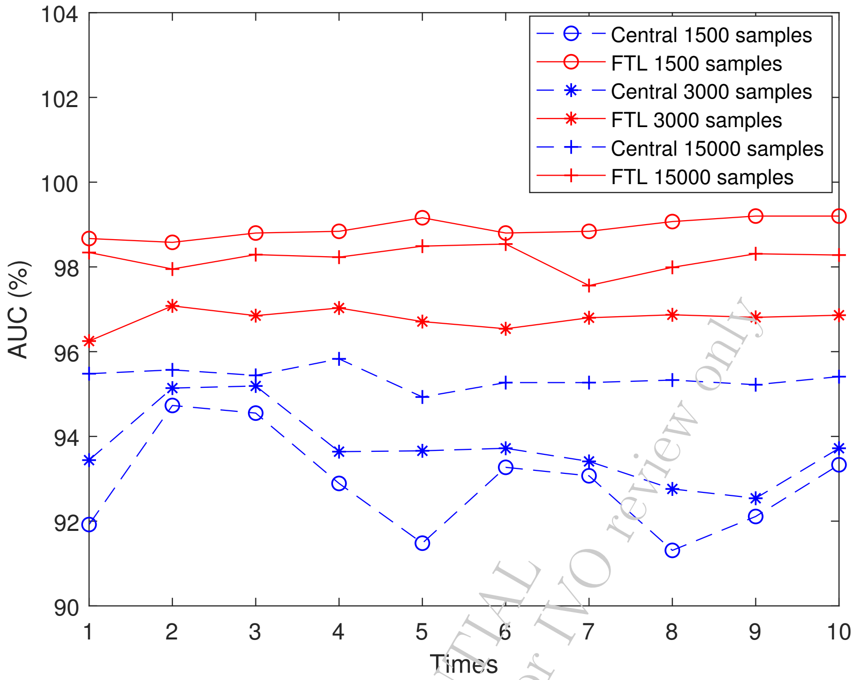


Fig. 5: Comparison of AUC between FTL and the centralized method.

V. Conclusion

In this report, we have summarized our latest results in applying federated transfer learning to cyberattack detection. FTL not only enhances the AUC score, in comparison with centralized deep learning, but also can perform the cooperative learning process with datasets that have different features. Through simulation results, we have demonstrated that FTL outperforms the baseline method in various situations. These results open the room for an extensive study of the application of FTL in different scenarios in cybersecurity to improve the performance of detection.

References

[1] L. Thames and D. Schaefer, *Cybersecurity for industry 4.0*. Springer, 2017.

[2] S. Dua and X. Du, *Data mining and machine learning in cybersecurity*. Auerbach Publications, 2016.

[3] A. Sahi, D. Lai, Y. Li, and M. Diykh, "An efficient DDoS TCP flood attack detection and prevention system in a cloud environment," *IEEE Access*, vol. 5, pp. 6036–6048, April 2017.

[4] A. Nezarat, "A game theoretic method for VM-to-hypervisor attacks detection in cloud environment," in *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. IEEE, 2017, pp. 1127–1132.

[5] G. Nenvani and H. Gupta, "A survey on attack detection on cloud using supervised learning techniques," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*. IEEE, 2016, pp. 1–5.

[6] K. K. Nguyen, D. T. Hoang, D. Niyato, P. Wang, D. Nguyen, and E. Dutkiewicz, "Cyberattack detection in mobile cloud computing: A deep learning approach," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2018, pp. 1–6.

[7] J. Konečnỳ, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," *arXiv preprint arXiv:1610.02527*, 2016.

[8] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *arXiv preprint arXiv:1909.11875*, 2019.

[9] D. Andročec and N. Vrček, "Machine Learning for the Internet of Things Security: A Systematic Review," in *The 13th International Conference on Software Technologies*, 2018.

[10] T. V. Khoa, Y. M. Saputra, D. T. Hoang, N. L. Trung, D. Nguyen, N. V. Ha, and E. Dutkiewicz, "Collaborative learning model for cyberattack detection systems in iot industry 4.0," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, 2020, pp. 1–6.

[11] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[12] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[13] Y. Aono, T. Hayashi, L. Wang, S. Moriai *et al.*, "Privacy-preserving deep learning via additively homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1333–1345, 2017.

[14] Y. Liu, T. Chen, and Q. Yang, "Secure federated transfer learning," *arXiv preprint arXiv:1812.03337*, 2018.

[15] "University of New Brunswick," https://www.unb.ca/cic/datasets/nsl.html.

[16] T. Fawcett, "An introduction to ROC analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[17] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, pp. 37–63, 2011.

Vietnam National University, Hanoi
University of Engineering and Technology

Advanced Institute of Engineering and Technology

Technical Report

UET-AVITECH-2020005 November, 2020

Data security using blockchain technology: *from Ethereum 1.0 to Ethereum 2.0*

Do Hai Son, Tran Thi Thuy Quynh, Dinh Thai Hoang,
Nguyen Linh Trung, Dusit Niyato, Diep N. Nguyen,
Nguyen Viet Ha, Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Data security using blockchain technology: from Ethereum 1.0 to Ethereum 2.0

Do Hai Son[1], Tran Thi Thuy Quynh[1], Dinh Thai Hoang[2],
Nguyen Linh Trung[1], Dusit Niyato[3], Diep N. Nguyen[2],
Nguyen Viet Ha[1], and Eryk Dutkiewicz[2]
[1] AVITECH, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
[2] Electrical and Data Engineering, University of Technology Sydney, Australia
[3] Computer Science and Engineering, Nanyang Technological University, Singapore

November, 2020

Abstract

Ethereum is the second most valuable cryptocurrency in the world. Although Ethereum has many benefits like global money and Distributed Application (DAPP), its consensus mechanism makes difficult and expensive for the scaling of this network. To deal with this problem, the next generation of Ethereum is developed with the huge change of mechanism from Proof-of-Work (PoW) to Proof-of-Stake (PoS), and it is named Ethereum 2.0 or “Serenity”. In this report, we provide an overview of Ethereum 1.0 and the migration to Ethereum 2.0. Then, we implement a private Ethereum 2.0 network and compare its power consumption versus the previous generation. Next, we review of the Random Decentralised Autonomous Organisation Algorithm (RANDAO) and its weakness which can be used for “last-revealer” attacks. Finally, a potential research direction about changing RANDAO is considered and discussed.

Index Terms

Ethereum 2.0, Proof-of-Stake, beacon-chain, validator.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. Introduction

Blockchain technology has a tremendous development in recent years with a famous project named “Bitcoin” [1]. Through Bitcoin, blockchain technology demonstrates its valuable characteristics such as decentralization, transparency, immutability, and security-and-privacy. Given the aforementioned outstanding benefits, blockchain technology has many applications in a number of areas. Some major applications of blockchain technology are as follows:

- *Cryptocurrencies*: Cryptocurrencies, e.g., Bitcoin, Ethereum [2], are the most famous applications of blockchain technology. With high value and high daily trade volume, cryptocurrencies can be utilized for various financial applications, such as digital assets and online retail.
- *Internet-of-Things (IoT) network*: Its security-and-privacy and anonymity make blockchains applicable to many IoT networks, e.g., smart home [3], energy trading [4]–[6], Internet-of-vehicles [7], [8].
- *Service*: Blockchain networks have been employed by many service providers. Blockchain technology can support automatic payments, contents distribution, and services delivery [9].
- *Military*: Blockchains have the potential to be applied in various military operations, such as enhancing data integrity in supply chain management, ensuring transparency in equipment management, and providing a distributed and decentralized database for military intelligence [10].

With high applicability, some cryptocurrencies are extremely valuable and widely used. As of October 21 2020, the top three largest market capitalization cryptocurrencies [11] are Bitcoin (BTC): $12,217.04 per coin, Ethereum (ETH): $379.76 per coin and Tether (USDT): $1 per coin. These cryptocurrencies all inherited the consensus mechanism of the blockchain technology’s first version which is Proof-of-Work (PoW). This mechanism likes the backbone of e.g., Bitcoin, Ethereum. But over time, PoW has not been suitable for scaling blockchain networks. The reason is the PoW’s mechanism of choosing the leader, who received a reward for firstly confirms a new block. The probability of a miner becoming the block winner (leader) is illustrated by the hash rate (computational power) of itself. This has caused a rat race to increase more efficient mining hardware when everyone wants to be the leader. Thus, the energy for maintaining the blockchain network becomes huge. According to [12], Bitcoin miners spend about 74 terawatt-hours power consumption per year - the same energy-consumption as Austria. Fig. 1 shows the percentage of Bitcoin’s energy consumption in several countries.

Various solutions have been proposed to solve PoW’s drawback, for instance: Proof-of-Stake (PoS), Proof-of-Concepts (PoC), Proof-of-Achievement (PoAch) [13]. One of the most famous algorithms is Proof-of-Stake (PoS), which was first proposed by Peercoin [14] in 2012. Nowadays, many coins have developed their PoS-based protocol, they are separated into two groups: firstly, the new coins use PoS, for example, Dashcoin [15], Peercoin [14]. Secondly, the old coins migrate from PoW come to PoS, for example Ethereum (Eth).

Eth is the most popular coin which has a plan to migrate from PoW to PoS. The most complex in this migration is that it has to keep all data from Eth1.0 (PoW) to Eth2.0 (PoS). Vitalik Buterin – Founder of Ethereum – have researched and planned about the upgrade

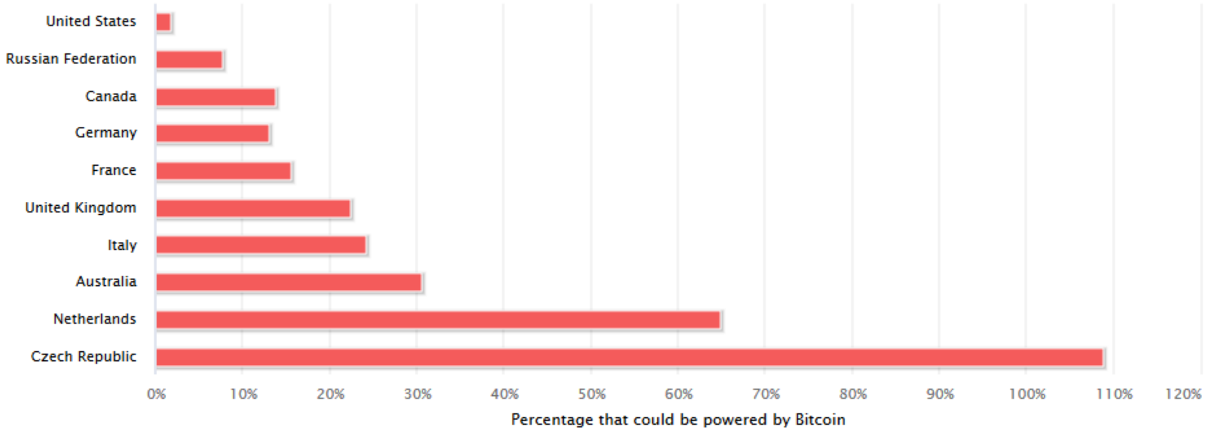


Fig. 1: Energy Consumption of Bitcoin Network in Several Countries [12].

to Eth2.0 since 2015 [16]. In 2017, the work in [17] introduced Casper, which is a PoS based finality system that overlays an existing proof of work blockchain. Recently, in [18], the consensus mechanism of Eth2.0 was almost completed. Besides these works, another document is published on the Internet, which is named eth2.0-specs [19]. From there, the migration has been segregated into three phases toward the final Eth2.0 network, as illustrated in Fig. 2:

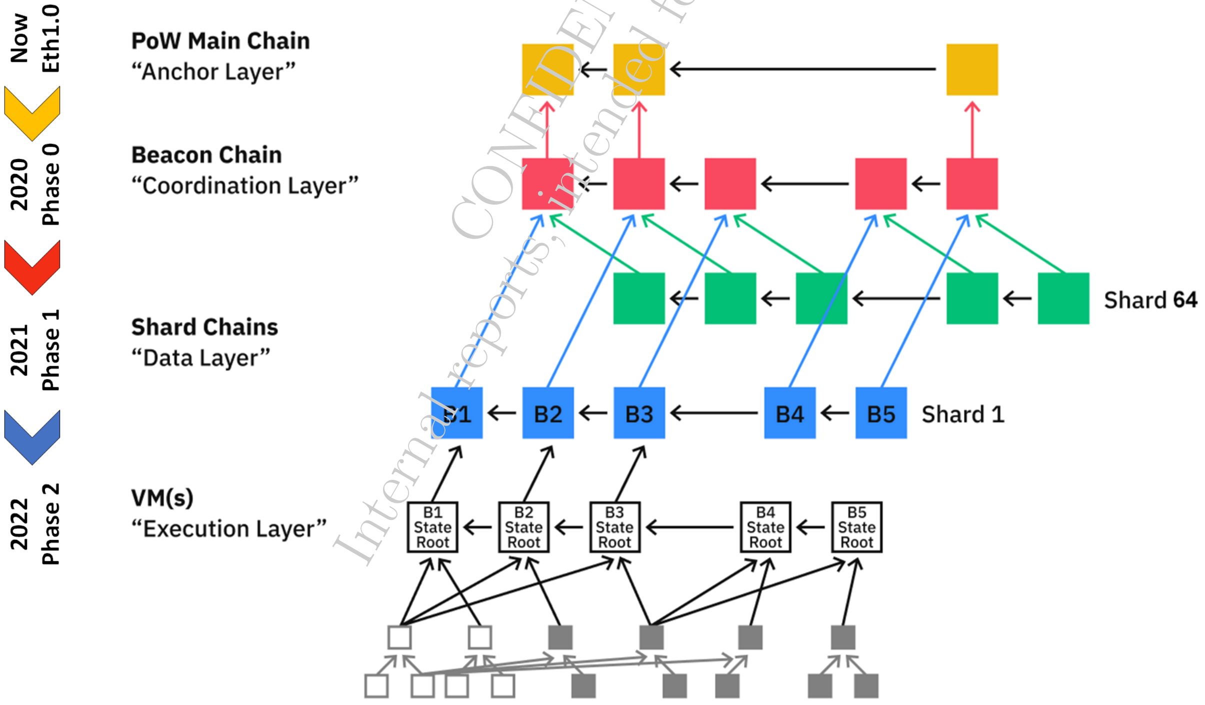


Fig. 2: Upgrade Ethereum 2.0 Timeline.

In Phase 0, Beacon Chain (heart) has been deployed since 2020. At this stage, the beacon chain network exists independently and Eth2.0 has not have any transactions yet. Firstly, the beacon chain will be responsible for building the function, which allows users at Eth1.0 stake their ETHs to regist "validators" (virtual miners) at Eth2.0. And the core of PoS-based consensus mechanism will be built in this phase.

In Phase 1, Shards (limbs) are expected from 2021. This phase is splited into 2 sub-phases. The main chain will be fragmented into 64 sub-chains, each of them is referred by a “shard”. These shards are referenced with the beacon chain by “crosslinks”. The shards are delegated a portion of Ethereum’s transactions and account data. In phase 1.5, Eth1.0 mainnet will officially become a shard and transit to PoS.

In Phase 2, Execution (brain) is planned to launch in 2022. This is the last phase, in there, shards should be fully functional chains. Shards will now be compatible with smart contracts and they will be able to communicate with each other more freely. Phase 2 is still a research phase.

From eth2.0-specs, many teams are building their own Ethereum 2.0’s implementation, such as Prysm (Prysmatic Labs) [20], Lighthouse [21]. Both have the same purpose, but Prysm is written in Golang and Lighthouse uses Rust language. Because Prysm’s team was donated by Vitalik Buterin and Golang is used in mainnet Eth1.0 [22], so we joined their community to do research together. This is a very new and large project, and thus requires many participants from the community for testing, reporting, and improvement.

Currently, Prysm has been in beta 1 and almost successfully performed Phase 0 of Eth2.0. From energy-consumption perspective, in this report, we set up a private Eth2.0 network and checked the power consumption of this new generation. This is one of the first reports to test the actual performance of the Eth2.0 network. From our experiment, we found out weaknesses in the Gasper mechanism and proposed potential solutions.

In the rest of the report, Section II reviews Eth1.0 and Eth2.0 technologies and the improvement of Eth2.0. Then, Section III describes our implementation of Eth2.0’s private testnet. Section IV shows the future works of this report. Finally, Section V summarizes the report.

II. Ethereum blockchain technology

In the summer of 2015, Vitalik Buterin launched a new blockchain network called Ethereum. Retaining core benefits from Bitcoin such as PoW, Peer-to-Peer (P2P), he added a key feature called “smart contract”. This makes Ethereum applicable for many purposes depending on the user’s programming. Typically, the Ethereum network is divided into seven protocol layers [23]: Storage, Data, Network, Protocol, Consensus, Contract, Application. These protocol layers are described by the structure given in Fig. 3.

The term “blockchain” is defined as a chain of blocks which are organized in chronological order. In the lowest level named Storage, all of this chain includes the history of blocks, transactions, hash, logs, these all stored in a database file (.db). This database is synchronized in any node of the Ethereum network using P2P connection.

The Data layer is the data structure of the blockchain network, which includes numerous cryptographic techniques, such as hashing functions and asymmetric encryption techniques. This data structure includes two primary components [24]: pointers and a linked list. The pointers are the variables, which refer to the location of another variable, and the linked list is a list of chained blocks, where each block has data and pointers to the previous block. A Merkle tree is a binary tree of hashes. Each block contains a hash of the Merkle root

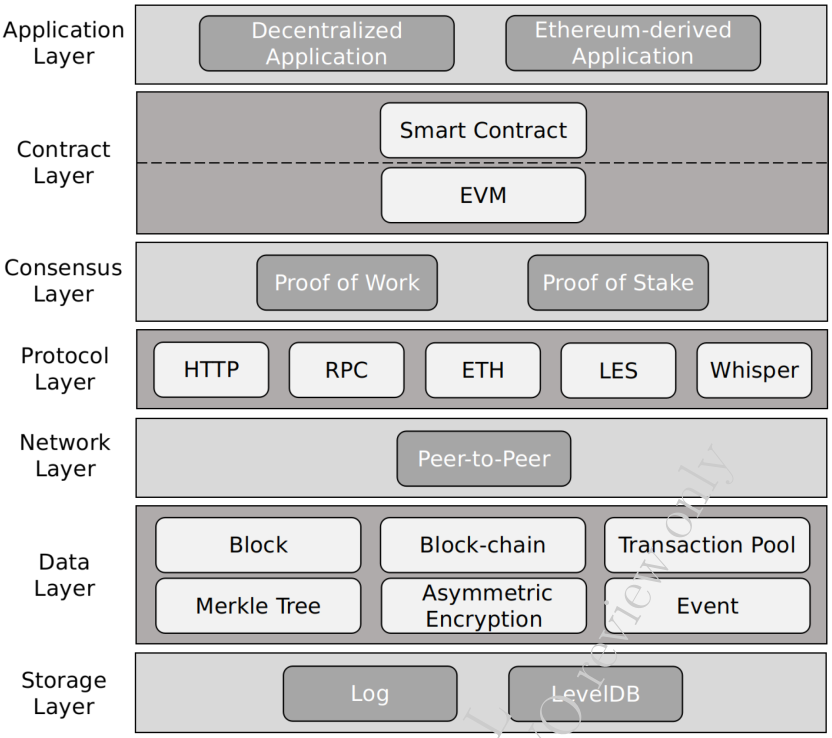


Fig. 3: Ethereum network layer classification [23].

with information such as the hash of the previous block, timestamp, nonce, and the current difficulty target as shown Fig. 4. A Merkle tree offers security, integrity, and irrefutability for the blockchain technology. Merkle trees, along with cryptography and consensus algorithms, are the basis of the blockchain technology. For example, an Ethereum blockchain uses a Alice tree database to store information. Alice tree is a Merkle tree, which is like a key-value store. Just like Merkle tree, a Alice tree has a root hash. This root hash can be used to refer to the entire tree. Hence, you can not modify the content of the tree without modifying the root hash. Each block contains a list of transactions that happened since the last block, and after applying those transactions, the root hash of the Alice tree represents the new state (state tree).

At the Network layer, the protocol in P2P network is used so all the data is encrypted and transmitted over P2P links. The chain of blocks includes all the history of transactions which is broadcast and stored in all nodes of the network. By this way, blockchain technologies can protect the privacy and data integrity of the network. Accounts in the same node interact with each other by Remote Procedure Calls (RPC) in the Protocol layer. The two main protocols used in Eth1.0 are Http and Websocket [26].

On top of Network and Protocol layer is the Consensus layer, which maintains the consistency, originality of data across blockchain network. In Eth, two main mechanisms are used, that is PoW in Eth1.0 and PoS in Eth2.0. Both of them will be clearly presented in the next sub-section.

Smart contract in the Contract layer is the highlight of Ethereum that is simply a piece of code, which is running on Ethereum. It is called a "contract" because code that runs on Ethereum can control valuable things like ETH or other digital assets. A smart contract can be built with Solidity language as shown in Fig. 5. The Solidity Compiler will compile the smart

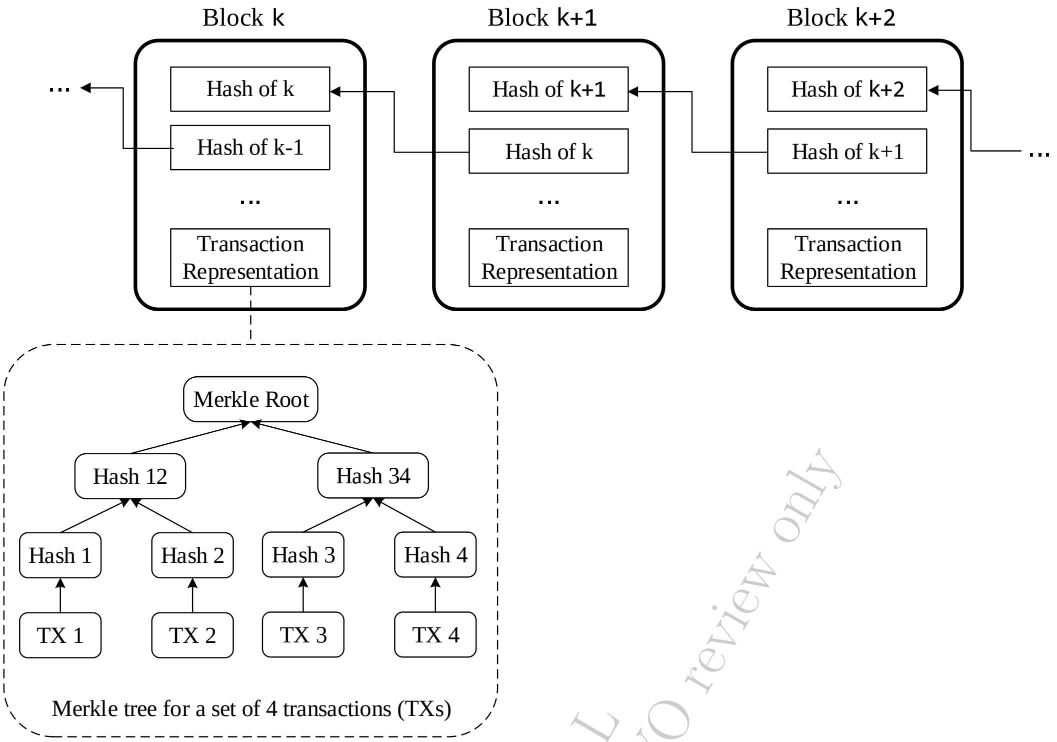


Fig. 4: A chain of blocks and the presentation of transaction in Merkle tree [25].

contract into Bytecode and Application Binary Interface (ABI). Both of them are packaged into a transaction and deployed into the Ethereum network. Where Bytecode is an executable code on Ethereum Virtual Machine (EVM) and Contract ABI is an interface to interact with EVM Bytecode. Web3 [27] is a tool provided for users to interact with smart contracts. With the address of smart contract and its ABI, Web3 allows the user to call functions and extract data from the smart contract for their intentions.

At the top of OSI scheme, the Application layer provides the user interface for each application to the system. People can work with blockchain technologies by their visible distributed applications such as web browsers, cryptocurrencies, Distributed Application (DAPP).

A. Nakamoto consensus mechanism

In [1], Nakamoto proposed a consensus protocol called "Proof-of-Work" to solve the problem of pseudonymity, synchronization, scalability, and security of blockchain. In a single node, the Nakamoto consensus protocol defines three procedures [28]: check each block in the chain and ensure that there is no conflict between any existing transaction; sync with the longest chain in the network; the last one is PoW solution and searching for propose new block, this procedure is the main point of the protocol. Typically, the nodes in a PoW-based blockchain network reach consensus by participating in a solution searching process, where each node must find a nonce for its proposed new block. To find the nonce, the previous blocks hash, and the transactions in the new block are used as the input of the hash function, e.g., SHA-256, the hash function output must be in a target range so that the block can be accepted. Due to the property of the hash function, the nonce can only be found by repeatedly trying different nonce values until the output is within the target range. When a participant

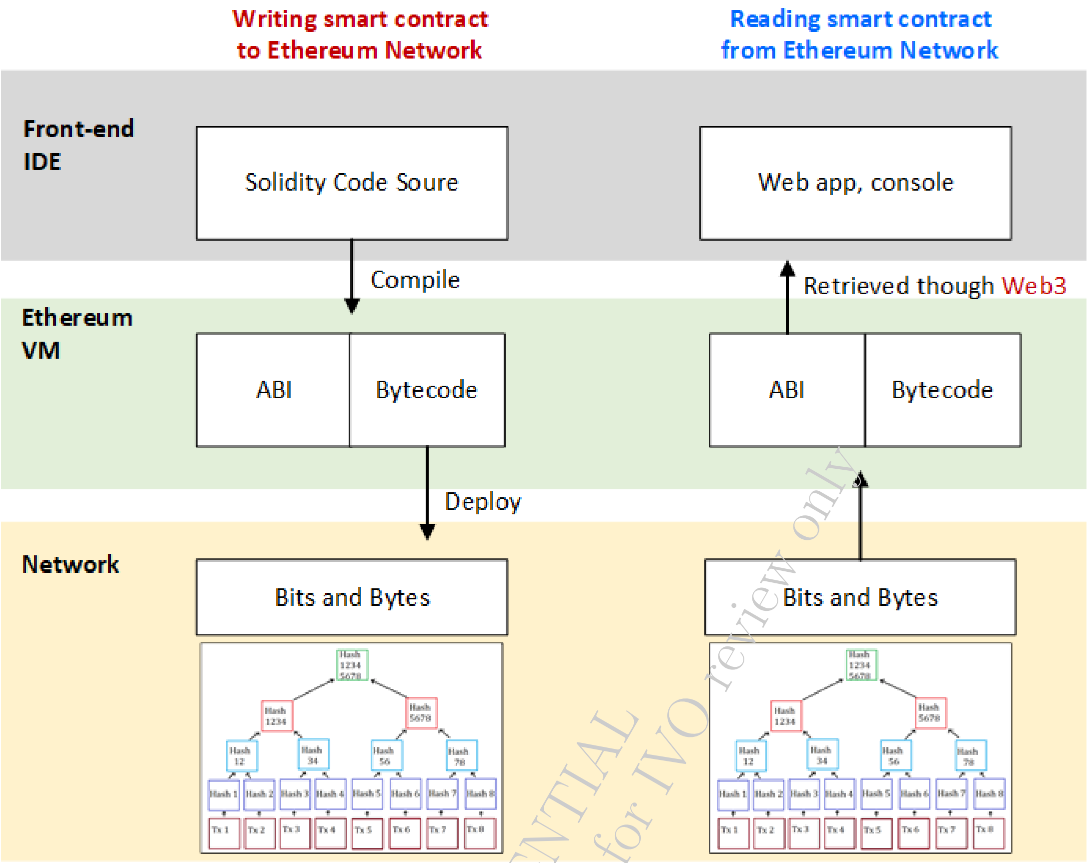


Fig. 5: Smart contract of Ethereum.

finds the nonce, it will broadcast the block along with the transactions to other nodes. Then, if the new block is verified and determined to be the first block mined after the last block in the chain, it will be integrated into the current chain and become the latest block in the chain. All this work is employed by "miner" which is the term for a computer in the network.

This solution searching procedure can be considered to be a weighted random coin-tossing process where a participant with a higher hash rate (computational power) might have higher chances to be the block winner (leader) who can receive the reward. The probability p_i that participant i is selected to be the leader in a network of N participants is:

$$p_i = \frac{c_i}{\sum_{j=1}^{N} c_j}, \tag{1}$$

where c_i is hash rate of participant i. Thus, if anyone wants to increase their chance to be the leader and receive rewards, they have to upgrade their computational power – Graphics Processing Unit (GPU) hardware. This results in the growth of power consumption by miners. Moreover, miners with low hash rates have very low chance to win this game and become leaders. Hence, they usually join mining pool to have more opportunities to get revenue. A mining pool consists of participants who want to collaborate by contributing their computing resources to the pool. In a pool, mining tasks will be distributed to the miners, due to the large number of miners, the computational power of the pools is superior to that of single miners. Rewards earned are divided among miners according to their hash rate contribution.

In fact, mining pools have been dominating processes making new blocks in most of current blockchain networks. To illustrate, the top five mining pools control up to 55.9% total hash rate of the Bitcoin network [29]. This is a serious problem of PoW mechanism, because it is against the decentralized spirit of blockchain technology. Another issue of PoW is delay, when a new block is added to chain, there is still a possibility that this block will not be included in the main chain for several reasons, e.g., network delay causing several versions of the chain or two participants finding two blocks simultaneously. This possibility decreases exponentially as the block is deeper in the chain. Therefore, a block is considered to be "finalized" only when it is a certain k blocks deep in the chain, such as $k = 6$ in Bitcoin.

B. *Ethereum 1.0: GHOST protocol*

Ethereum has many similarities with the aforementioned Nakamoto consensus mechanism. However, in Bitcoin, confirming time of a new block is about 98 minutes [30], and as mentioned above it will take $98 \times 6 = 588$ minutes to finalized a block. That was too long, so Ethereum has solved this by the Greedy Heaviest Observed Subtree (GHOST) protocol. The motivation behind GHOST is that blockchains with fast confirmation times currently suffer from reduced security due to a high stale rate - because blocks take a certain time to propagate through the network, if miner A mines a block and then miner B happens to mine another block before miner A's block propagates to B, miner B's block will end up wasted and will not contribute to network security. Furthermore, there is a centralization issue: if miner A is a mining pool with 30% hashpower and B has 10% hashpower, A will have a risk of producing a stale block 70% of the time (since the other 30% of the time A produced the last block and so will get mining data immediately) whereas B will have a risk of producing a stale block 90% of the time. Thus, if the block interval is short enough for the stale rate to be high, A will be substantially more efficient simply by virtue of its size. With these two effects combined, blockchains which produce blocks quickly are very likely to lead to one mining pool having a large enough percentage of the network hashpower to have the fact control over the mining process.

As described by Sompolinsky and Zohar [31], GHOST solves the first issue of network security loss by including stale blocks in the calculation of which chain is the "longest"; that is to say, not just the parent and further ancestors of a block, but also the stale descendants of the block's ancestor (in Ethereum, it is called "uncles") are added to the calculation of which block has the largest total proof of work backing it. Figure 6 illustrates a scenario in which a highly forked block tree was created by the honest network. The attacker secretly creates a chain of 6 blocks (denoted 1A, 2A,. . . , 6A) which is clearly longer than the network's longest chain (ending in block 5B). If block propagation was faster (in relation to the creation rate), all blocks in the honest network's tree would form a single long chain and would not be overtaken by the attacker.

To solve the second issue of centralization bias, Ethereum go beyond the protocol described by Sompolinsky and Zohar, and also provide block rewards to stales: a stale block receives 87.5% of its base reward, and the nephew that includes the stale block receives the remaining 12.5%. Transaction fees, however, are not awarded to uncles. Ethereum implements a simplified version of GHOST which only goes down seven levels. Specifically, it is defined as follows [2]:

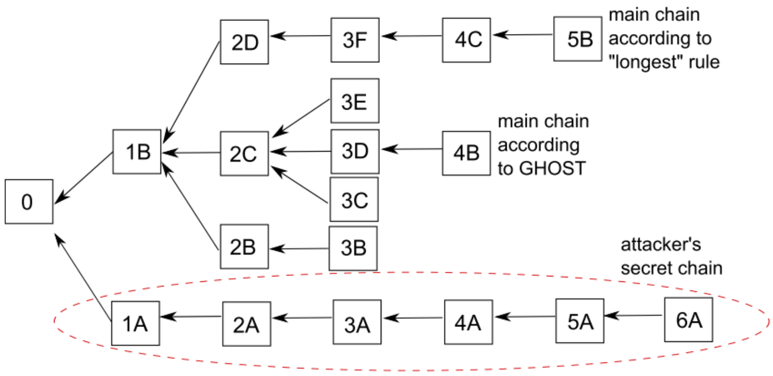

0
1B
1A
2D
2C
2B
2A
3F
3E
3D
3C
3B
3A
4C
4B
4A
5B
5A
6A
main chain according to "longest" rule
main chain according to GHOST
attacker's secret chain

Fig. 6: A block tree in which the longest chain and the chain selected by GHOST differ. An attacker's chain is able to switch the longest chain, but not the one selected by GHOST [31].

- A block must specify a parent, and it must specify 0 or more uncles.
- An uncle included in block B must have the following properties:
  - It must be a direct child of the k-th generation ancestor of B, where 2 <= k <= 7.
  - It can not be an ancestor of B.
  - An uncle must be a valid block header, but does not need to be a previously verified or even valid block.
  - An uncle must be different from all uncles included in previous blocks and all other uncles included in the same block (non-double-inclusion).
- For every uncle U in block B, the miner of B gets an additional 3.125% added to its coinbase reward and the miner of U gets 93.75% of a standard coinbase reward.

Generally, GHOST protocol has reduced resource waste, time to confirm block, increased security and profitability for miners. In fact, confirming time of a new block is about 15 seconds [32], and as mentioned above it will take $15 \times 7 = 105$ seconds to finalized a block. This is much faster than Bitcoin, however GHOST protocol is still based on aggregate computational power, so the high energy consumption and the risk of a 51% attack [33] are always there. In order to solve these two issues as well as enhance the network scalability, two main groups of methods have been employed: firstly, improvement from core of Eth1.0, that is Eth2.0 project; secondly, technologies are built based on Eth1.0, they only change the protocol at the Contract and Application layer to reduce the number of transactions required. The first solution is the long-term development direction, it is still being researched and deployed, but promised to be the core technology of blockchain in the future. Therefore, in the next sub-section, we will review about Eth2.0 and their key concepts, for example: shards, staking validators, attestations, committees, checkpoints, and finality.

C. Ethereum 2.0 (Serenity): Gasper protocol

1) Scalability: As mentioned above, the main problem in scalability that blockchains, including Ethereum, currently face is that every node has to verify and execute every transaction. In computer science, there are two main approaches to scaling:

1) Scaling vertically: basically, make nodes more and more powerful.
2) Scaling horizontally: basically, add more nodes.

For decentralization, blockchains need to scale horizontally as shown in Fig.2. A goal of Ethereum 2.0 is for nodes to run on consumer hardware. "Sharding" is the term for horizontally partitioning a database. Generally, a shard chain has a subset of nodes processing it. Virtual miners - "validators" are assigned to shards, and only process and validate transactions in that shard (chain). Ethereum's shards have a dynamic subset of nodes processing it block-by-block.

The main challenge with sharding a blockchain is the security of shards. Since validators are spread out across shards, malicious validators could takeover a single shard. This issue is solved by random shuffling of validators. Where every shard block has a (pseudo) randomly chosen "committee" of validators, ensures that it is mathematically improbable that an attacker controlling less than 1/3 of all validators can attack a single shard. That is a big picture of Ethereum 2.0's sharding, which has three phases to be completed as mentioned above.

2) Gasper: Proof of Stake mechanism: In 2019, Vitalik Buterin et al. proposed the Gasper mechanism which combined the GHOST and Casper protocol together. This is the new consensus mechanism in Eth2.0, and because it is a big upgrade, so many terms have been defined, as follows:

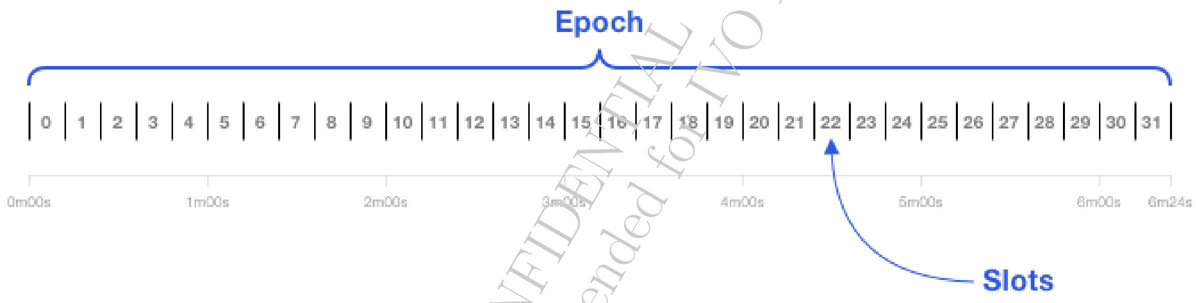


Fig. 7: An Epoch in Eth2.0 [34].

- Validator: Validators are actively participating in the consensus of the Eth2.0 protocol, they are virtual and are activated by stakers. In PoW, users employ powerful computers to become miners. In Eth2.0, users stake 32 ETH to activate and control validators.
- Slot: Slot is a chance for a block to be added to the Beacon Chain and shards. Every 12 seconds, one beacon (chain) block and 64 shard blocks are added when the system is running optimally. Validators need to be roughly synchronized with time. A slot is like the block time, but slots can be empty. Genesis blocks for the Beacon Chain and shards are at Slot 0. Shards will start at a future epoch than the Beacon Chain's Epoch 0, but will have their own Epoch 0 that includes their genesis blocks.
- Epoch: An Epoch is 32 slots, this takes place 6m24s, as shown in Fig 7.
- Proposer: A validator is chosen to propose a new block.
- Committee: A group of validators. A committee is responsible for its block's validation.
- Attestation: Validators in a committee will vote for that chain's head. A vote weighted by the validator's balance, that voting is called by an attestation.

Fig 8 shows the mechanism of choosing proposers and committees for each slot in Eth2.0. When new transactions are added, they have chance to take a slot for validation. Then, a proposer and a committee that have been pseudorandomly selected to propose and attest this block. The Beacon Chain enforces consensus on a pseudorandom process called RAN-

DAO [35] (Section IV). At every epoch, a pseudorandom process RANDAO selects proposers for each slot and shuffles validators to committees, respectively. For security, each slot (in the Beacon Chain and each shard) has committees of at least 128 validators. An attacker has less than a one in a trillion probability of controlling 2/3 of a committee [36]. All of the validators from that slot attest to the Beacon Chain head. A shuffling algorithm scales up or down the number of committees per slot to get at least 128 validators per committee.

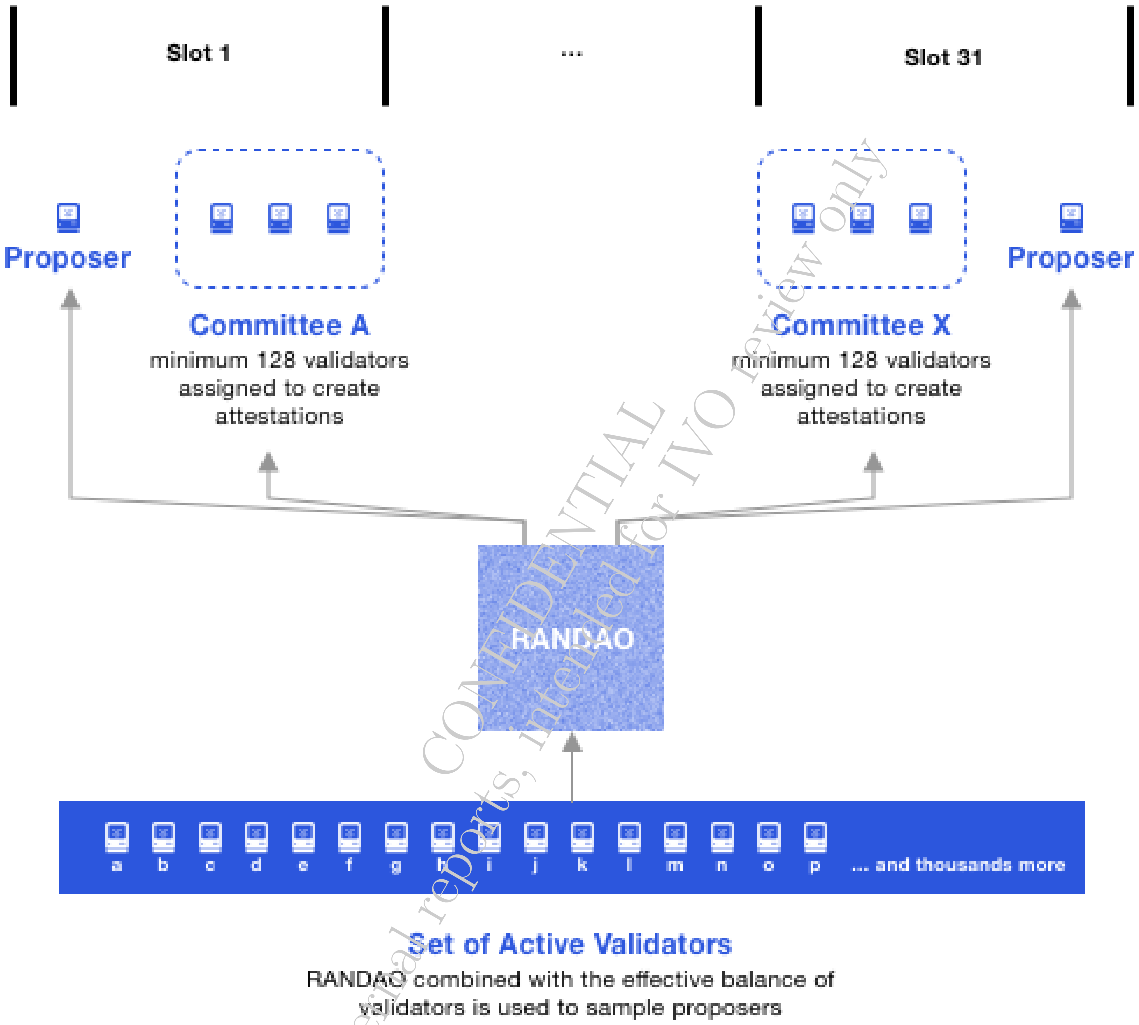

Slot 1
...
Slot 31
Proposer
Committee A
minimum 128 validators assigned to create attestations
Committee X
minimum 128 validators assigned to create attestations
Proposer
RANDAO
a
b
c
d
e
f
g
h
i
j
k
l
m
n
o
p
... and thousands more
Set of Active Validators
RANDAO combined with the effective balance of validators is used to sample proposers

Fig. 8: RANDAO: choice proposers and committee [34].

Penalties and rewards for validators are designed to “slashing” stalker and incentive honest users. First, validators get rewards for making attestations that the majority of other validators agree with. On the flip side, validators get penalties for not attesting or if they attest to blocks that are not finalized. Firstly, proposers of blocks that get finalized, also obtain a sizable reward. Validators that are consistently online doing a good job accrue ∼1/8 boost to their total rewards for proposing blocks with new attestations. Slashings are penalties ranging from over 0.5 ETH up to a validator’s entire stake. For committing a slashable offense a validator loses at least 1/32 of their balance and is deactivated (“forced exit”). The validator is penalized as if it was offline for 8192 epochs. The protocol also imposes an additional penalty based on how many others have been slashed near the same time. The basic equation for the additional penalty is:

$$\text{additional\_penalty} = \text{validator\_balance} \times 3 \times \text{fraction\_of\_validators\_slashed}. \qquad (2)$$

An effect is that if 1/3 of all validators commit a slashable offense in a similar period of time, they lose their entire balance. When a slashing happens, proposers also get a small reward for including the slashing evidence in a block. In Eth2 Phase 0, all of the whistleblower's reward actually goes to the proposer.

A checkpoint is a block in the first slot of an epoch. If there is no such block, then the checkpoint is the preceding most recent block. There is always one checkpoint block per epoch. A block can be the checkpoint for multiple epochs. For example, in Fig 9, Slot 65 to Slot 128 are empty. The Epoch 2 checkpoint would have been the block at Slot 128. Since the slot is missing, the Epoch 2 checkpoint is the previous block at Slot 64. Epoch 3 is similar: Slot 192 is empty, thus the previous block at Slot 180 is the Epoch 3 checkpoint. These checkpoints are the reference points for finalized blocks.

To finalize a slot, this slot will take two states. That is justified and finality [18] as named Gasper protocol. For the sake of simplicity, it requires at least two valid epochs to finality validate a slot. Gasper is proven to be secure as long as 2/3 of voting power is controlled by honest validators. In addition, withdrawing also takes time to execute (at least 27 hours [19]), which reduces the chance of the network being attacked.

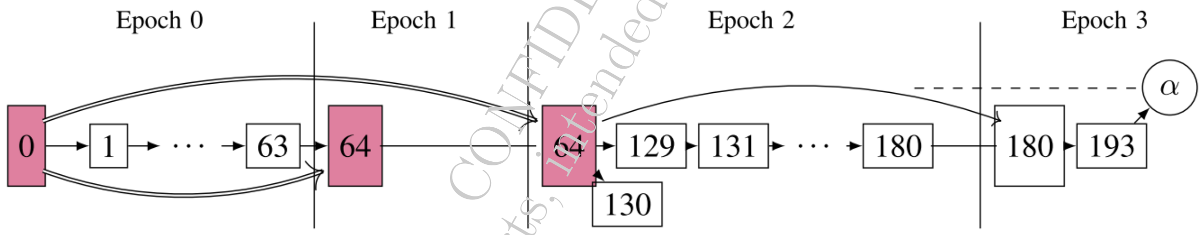


Fig. 9: Checkpoint in Eth2.0 [18].

Phases 1 and 2 of Ethereum 2.0 have been developing, and details of them are expected from 2021.

In summary, through the new Gasper mechanism, Eth2.0 will solve the energy problem because it does not require complicated calculations. All based on a pseudorandom with a weighting on the validator's balance. Table I shows the great specification of Eth2.0.

Table I: Consensus mechanisms comparisons.

	Bitcoin	**Ethereum 1.0**	**Ethereum 2.0**
Type	PoW	PoW	PoS
Proposer selection	Base on hash rate	Base on hash rate	Base on stake
Hardware requirement	High	High	Medium to none
Average transaction mining time	Avg around: 98 minutes [30]	Avg around: 15 seconds [32]	Fixed 12 seconds
Finality time	After 6 new blocks are added Avg around: 10 hours	After 7 new blocks are added Avg around: 2 minutes	After at least 2 Epochs are added 12.8 minutes

III. Testing Ethereum 2.0 at Phase 0

In this section a private Ethereum 2.0 testnet is built and the network is tested under the power-consumption context.

A. *Set up an Ethereum 2.0 network at Phase 0*

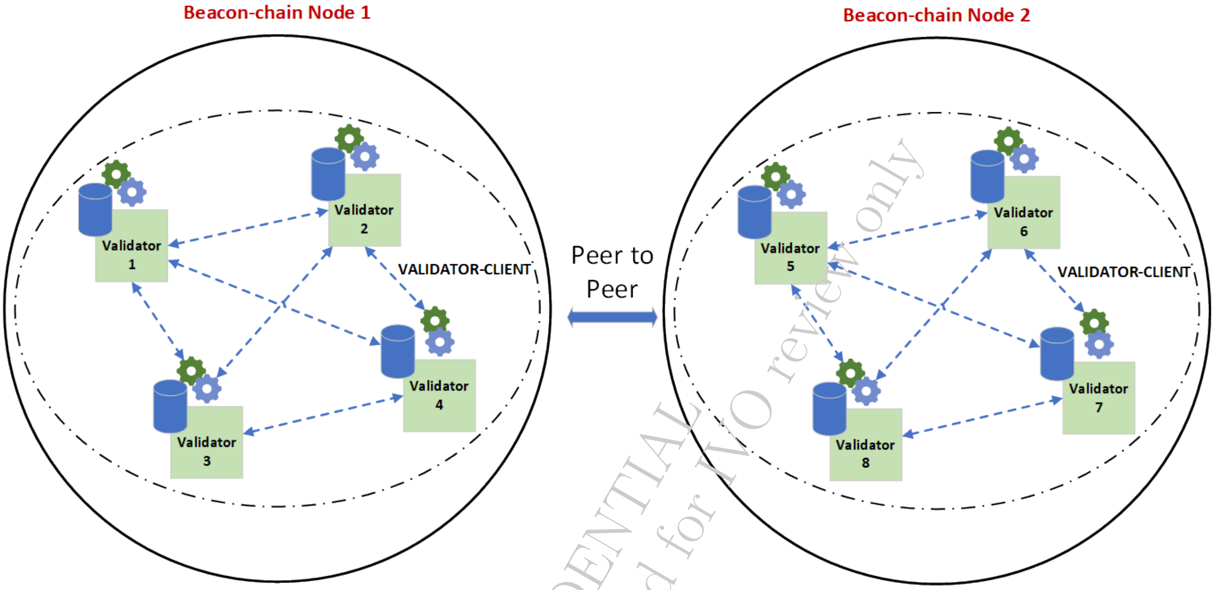


Fig. 10: Private Ethereum 2.0 network.

Prysm Labs [20] is a team of individuals with a deep understanding of blockchain technology. From Eth2.0 specs, they build a project named Prysm on Github. It is one of the first Ethereum 2.0 projects, that was deployed on the Internet for everyone testing. At the time of writing this report, Prysm has gradually completed Eth2.0 Phase 0 and provided two different testnets: Medalla [37] and Zinken [38]. Prysm is written by GoLang and built by Bazel for fast coding and fast compiling.

In order to create a private Ethereum 2.0 testnet, Prysm is used. Firstly, initialization parameters for a node must be defined. The work is completed by creating a “genesis.ssz” file. This file includes several parameters such as the number of validators, network-config. For example, the network, shown in Fig. 10, will be initialized by this command below:

```
$ bazel run //tools/genesis-state-gen –num-validators=4 –output-ssz=/tmp/genesis.ssz –mainnet-config
```

when completed, a ”genesis.ssz” file has been saved on output-ssz location. While running beacon chain node by ssz file, on another termial window, run the following command:

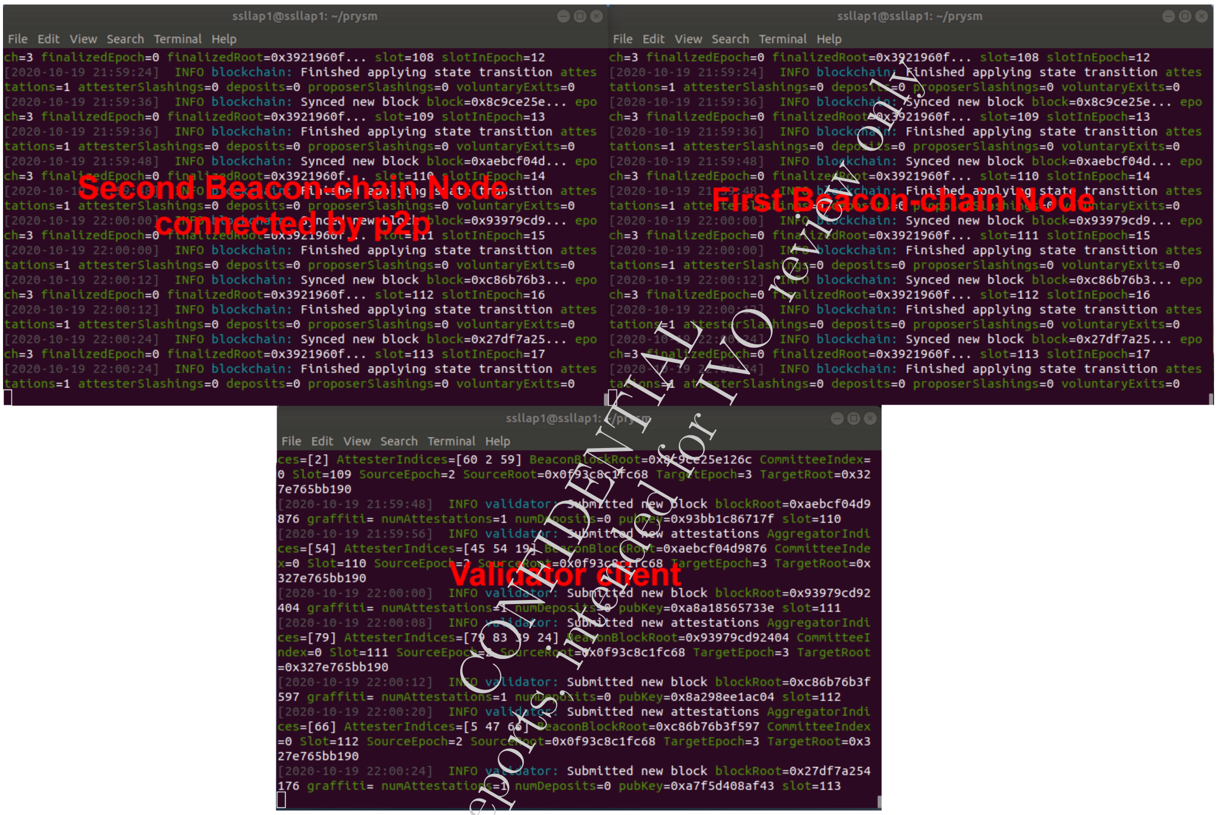
```
$ bazel run //beacon-chain –define ssz=mainnet – –bootstrap-node= –datadir /tmp/chaindata –force-clear-db –interop-genesis-state /tmp/genesis.ssz –interop-eth1data-votes –min-sync-peers=0
```

where –datadir is location of Beacon chain node database, –interop is a option for running private node. The last, a validator client will be run:

$ bazel run //validator – –beacon-rpc-provider localhost:4000 –interop-num-validators=4 –interop-start-index=0 –clear-db

where –beacon-rpc-provider is rpc address of beacon chain node. In addition, to add a new beacon chain node with the created network, simply add this flag: "–peer=enr" when run second node. where enode is a parameter that defines a specific node and can be got by "http://localhost:8080/p2p".

In Fig. 11, the power consumption of the Ethereum 2.0 and Ethereum 1.0 testnet are shown.



```
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=108 slotInEpoch=12
[2020-10-19 21:59:24]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 21:59:36]  INFO blockchain: Synced new block block=0x8c9ce25e... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=109 slotInEpoch=13
[2020-10-19 21:59:36]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 21:59:48]  INFO blockchain: Synced new block block=0xaebcf04d... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=110 slotInEpoch=14
[2020-10-19 21:59:48]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:00]  INFO blockchain: Synced new block block=0x93979cd9... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=111 slotInEpoch=15
[2020-10-19 22:00:00]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:12]  INFO blockchain: Synced new block block=0xc86b76b3... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=112 slotInEpoch=16
[2020-10-19 22:00:12]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:24]  INFO blockchain: Synced new block block=0x27df7a25... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=113 slotInEpoch=17
[2020-10-19 22:00:24]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
```

```
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=108 slotInEpoch=12
[2020-10-19 21:59:24]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 21:59:36]  INFO blockchain: Synced new block block=0x8c9ce25e... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=109 slotInEpoch=13
[2020-10-19 21:59:36]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 21:59:48]  INFO blockchain: Synced new block block=0xaebcf04d... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=110 slotInEpoch=14
[2020-10-19 21:59:48]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:00]  INFO blockchain: Synced new block block=0x93979cd9... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=111 slotInEpoch=15
[2020-10-19 22:00:00]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:12]  INFO blockchain: Synced new block block=0xc86b76b3... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=112 slotInEpoch=16
[2020-10-19 22:00:12]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
[2020-10-19 22:00:24]  INFO blockchain: Synced new block block=0x27df7a25... epo
ch=3 finalizedEpoch=0 finalizedRoot=0x3921960f... slot=113 slotInEpoch=17
[2020-10-19 22:00:24]  INFO blockchain: Finished applying state transition attes
tations=1 attesterSlashings=0 deposits=0 proposerSlashings=0 voluntaryExits=0
```

```
ces=[2] AttesterIndices=[60 2 59] BeaconBlockRoot=0x8c9ce25e126c CommitteeIndex=
0 Slot=109 SourceEpoch=2 SourceRoot=0x0f93c8c1fc68 TargetEpoch=3 TargetRoot=0x32
7e765bb190
[2020-10-19 21:59:48]  INFO validator: Submitted new block blockRoot=0xaebcf04d9
876 graffiti= numAttestations=1 numDeposits=0 pubKey=0x93bb1c86717f slot=110
[2020-10-19 21:59:56]  INFO validator: Submitted new attestations AggregatorIndi
ces=[54] AttesterIndices=[45 54 19] BeaconBlockRoot=0xaebcf04d9876 CommitteeInde
x=0 Slot=110 SourceEpoch=2 SourceRoot=0x0f93c8c1fc68 TargetEpoch=3 TargetRoot=0x
327e765bb190
[2020-10-19 22:00:00]  INFO validator: Submitted new block blockRoot=0x93979cd92
404 graffiti= numAttestations=1 numDeposits=0 pubKey=0xa8a18565733e slot=111
[2020-10-19 22:00:08]  INFO validator: Submitted new attestations AggregatorIndi
ces=[79] AttesterIndices=[79 83 39 24] BeaconBlockRoot=0x93979cd92404 CommitteeI
ndex=0 Slot=111 SourceEpoch=2 SourceRoot=0x0f93c8c1fc68 TargetEpoch=3 TargetRoot
=0x327e765bb190
[2020-10-19 22:00:12]  INFO validator: Submitted new block blockRoot=0xc86b76b3f
597 graffiti= numAttestations=1 numDeposits=0 pubKey=0x8a298ee1ac04 slot=112
[2020-10-19 22:00:20]  INFO validator: Submitted new attestations AggregatorIndi
ces=[66] AttesterIndices=[5 47 66] BeaconBlockRoot=0xc86b76b3f597 CommitteeIndex
=0 Slot=112 SourceEpoch=2 SourceRoot=0x0f93c8c1fc68 TargetEpoch=3 TargetRoot=0x3
27e765bb190
[2020-10-19 22:00:24]  INFO validator: Submitted new block blockRoot=0x27df7a254
176 graffiti= numAttestations=1 numDeposits=0 pubKey=0xa7f5d408af43 slot=113
```

Fig. 11: Ethereum 2.0 network: Beacon chain nodes, validators.

B. Power consumption

Difficulty and power consumption are mainly limitations of Ethereum 1.0. In order to analyze them, firstly, a private network as mentioned above has been deployed on a laptop that uses CPU core i7-4810MQ, RAM 16GB. Blocks are sequentially added and validated, while the information about CPU consumed (difficulty) and active power of CPU (power consumption) are collected in a sample per second.

Similarly, Go Ethereum (Geth) [22] has been created the same network with Ethereum 2.0. The network includes two nodes and four miners in each node. The initialization difficulty of this network is 1/10 of mainnet Ethereum 1.0 [39], about 4.5 Tera Hash per second. Blocks are also added and mined.

The obtained results show that a new block of Ethereum 2.0 takes place **12s** to confirm and Ethereum 1.0 needs **5m49s** to be mined a block, this time will be increased whenever a new block is added. CPU consuming in process of mining and validating is shown Fig. 12. The PoS mechanism of Ethereum 2.0 solved the difficulty of Ethereum 1.0.

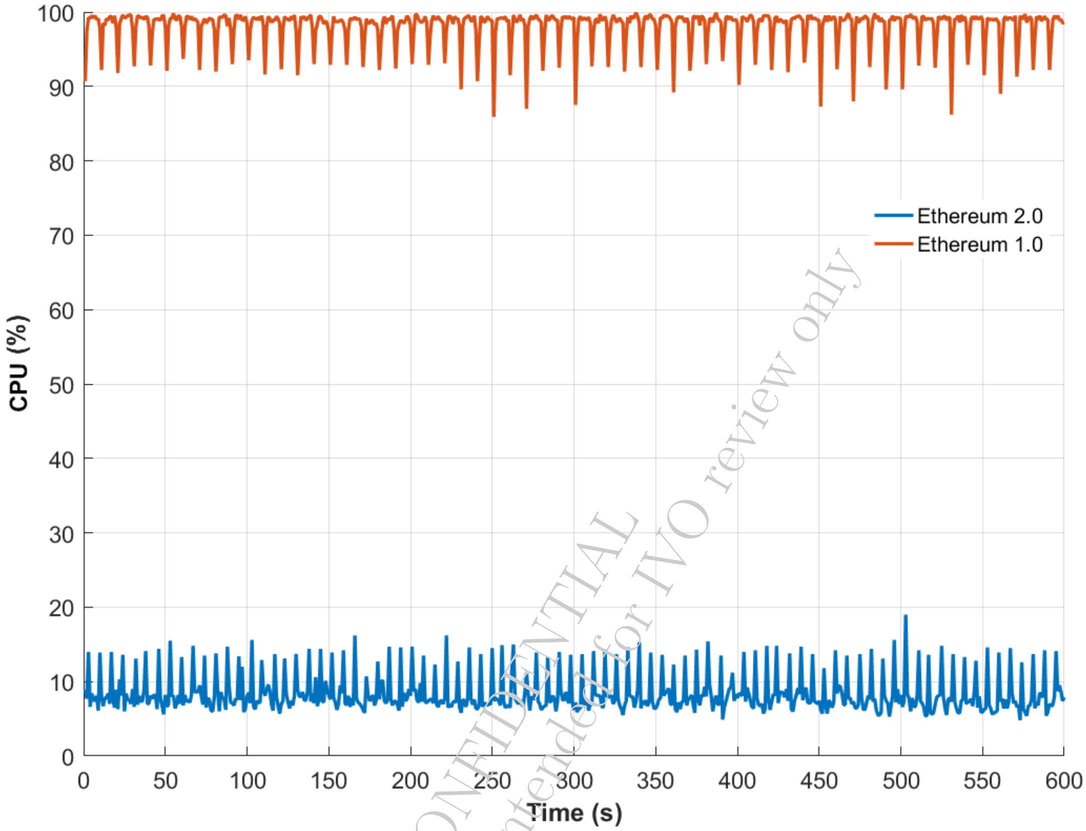


Fig. 12: CPU consumption comparisons.

Fig. 13 presents power consumption by CPU. The above results show only very little CPU is used, so that power consumption is also minimal.

IV. Future work: RANDAO

RANDAO is one of the most essential mechanisms in Eth2.0. It is in charge of creating a new "seed" to generate random proposers and committees. Thus, this mechanism directly affects to Stake of validators. In this section, we present the RANDAO mechanism on Eth2.0 and the last-revealer attack to RANDAO. In the end, Shamir's Secret Sharing algorithm is considered to deal with this problem.

Firstly, RANDAO [35] is a commitment scheme [40], this is a well-known example presenting the fundamental ideas of this scheme [41], suppose Alice and Bob want to resolve some dispute via coin-flipping. If they are physically in the same place, a typical procedure might be:

1) Alice "calls" the coin flip.
2) Bob flips the coin.

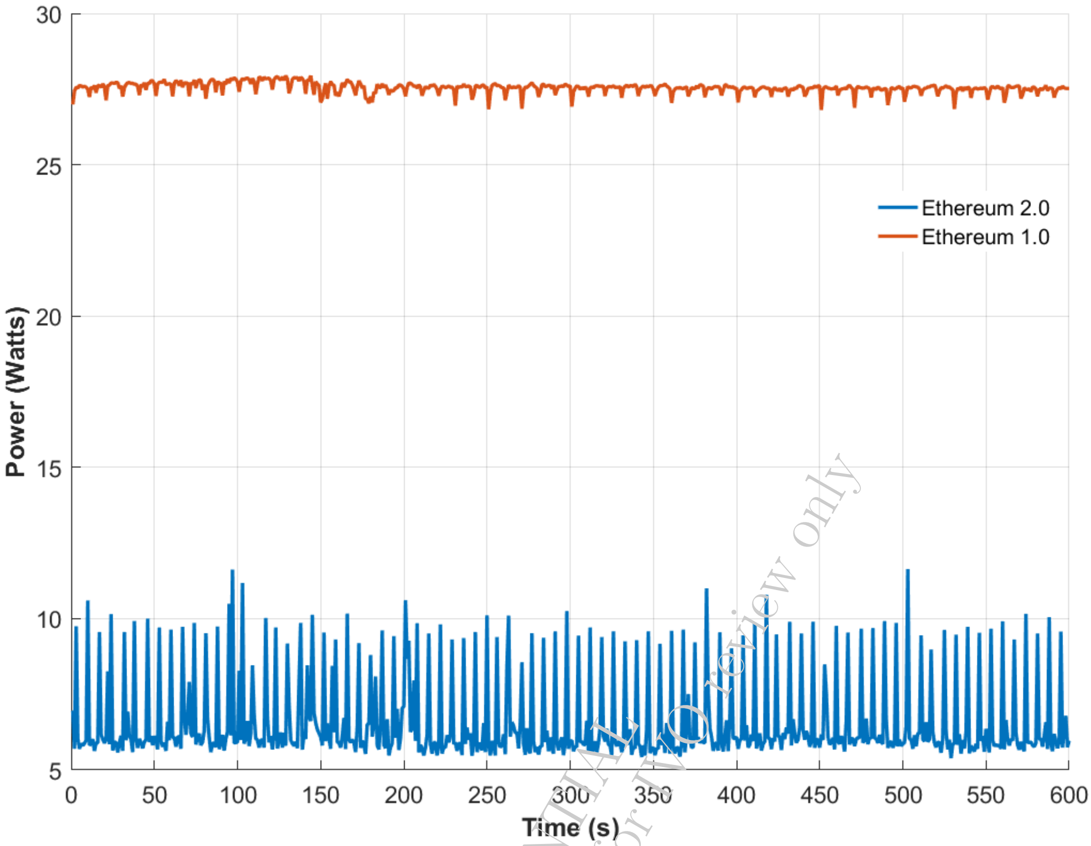


Fig. 13: Power consumption comparisons.

3) If Alice's call is correct, she wins, otherwise Bob wins.

If Alice and Bob are not in the same place a problem arises. Once Alice has "called" the coin flip, Bob can stipulate the flip "results" to be whatever is most desirable for him. Similarly, if Alice does not announce her "call" to Bob, after Bob flips the coin and announces the result, Alice can report that she called whatever result is most desirable for her. Alice and Bob can use commitments in a procedure that will allow both to trust the outcome:

1) Alice "calls" the coin flip but only tells Bob a commitment to her call.
2) Bob flips the coin and reports the result.
3) Alice reveals what she committed to.
4) Bob verifies that Alice's call matches her commitment.
5) If Alice's revelation matches the coin result Bob reported, Alice wins.

For Bob to be able to skew the results to his favor, he must be able to understand the call hidden in Alice's commitment. If the commitment scheme is a good one, Bob can not skew the results. Similarly, Alice can not affect the result if she can not change the value she commits to.

The commitment scheme is applied to RANDAO where Alice and Bob are replaced by validators. Typically in an Epoch, we have 32 proposers. These proposers are now responsible for choosing "seed" which is used to randomly select proposers and committees of the next Epoch. The "seed" is a 32 bytes number and it is mixed by secret numbers of proposers. Each proposer includes the commit by a hash in their block. After all 32 slots, the reveal-period is started, proposers subsequently reveal their 32 bytes secret number (r_1, ..., r_n) which can be verified by their committed hash. Secrets which were not revealed or skipped are considered 0x00000000... And the output of RANDAO is "XOR" ($r = \oplus_i r_i$) of all proposers' secret numbers.

RANDAO is great for pseudorandomness but it suffers from last-revealer attacks in cryptography. Because during the reveal-period, each member publicly commits to a secret contribution to the final output as Fig. 14, specifically, a malicious actor can observe the network once others start to reveal their numbers and choose to reveal or not to reveal their number based on XOR of the numbers observed so far. This allows a single malicious actor to have one bit of influence on the output, and a malicious actor controlling multiple participants have as many bits of influence as the number of participants they are controlling [42].

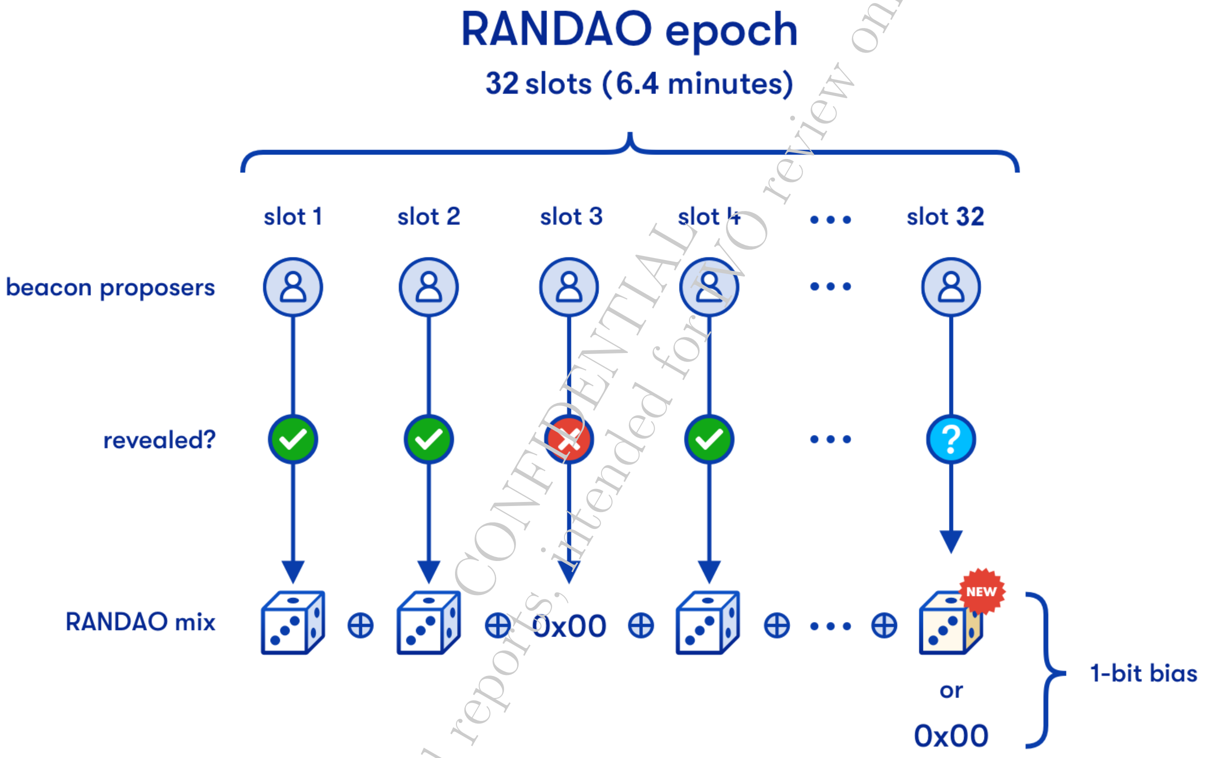


Fig. 14: An illustrate: Last-reveal attack on RANDAO.

A solution was proposed to replace RANDAO on Eth2.0, that is Verifiable Delay Function (VDF) [43]. This method is to add a delay function after mixed, making it slow to compute the beacon outcome from an input of RANDAO mix-period. VDF reduces the probability of a last-revealer attack to at least 1 honest validator in an Epoch. However, we need a new period that named "Eval", and this is also has to "Verify" by ASIC hardware [44]. To keep the time per epoch, VDF is employed with the pipeline technique. The benefits of VDF is not only Eth2.0 but also Proof of replication, Resource-efficient blockchain, and computational timestamping [43]. But, ASIC hardware research is still very much in the research phase.

In this report, we considered using Shamir's secret sharing (SSS) [45] to deal with last-revealer attacks. SSS is an algorithm that divides a secret into "shares". The secret can be recovered by combining certain numbers of shares. We defined some terms used in SSS, as follows:

Definition 1. Secret (S) is a secret message that you want to share with others securely.

Definition 2. Share is a piece of secret. Secret is divided into pieces $(\mathbf{S1}, \mathbf{S2}, ..., \mathbf{SN})$ and each piece is called share. It is computed from given secret.

Definition 3. Threshold (N) is the minimum number of shares we need in order to recover your secret.

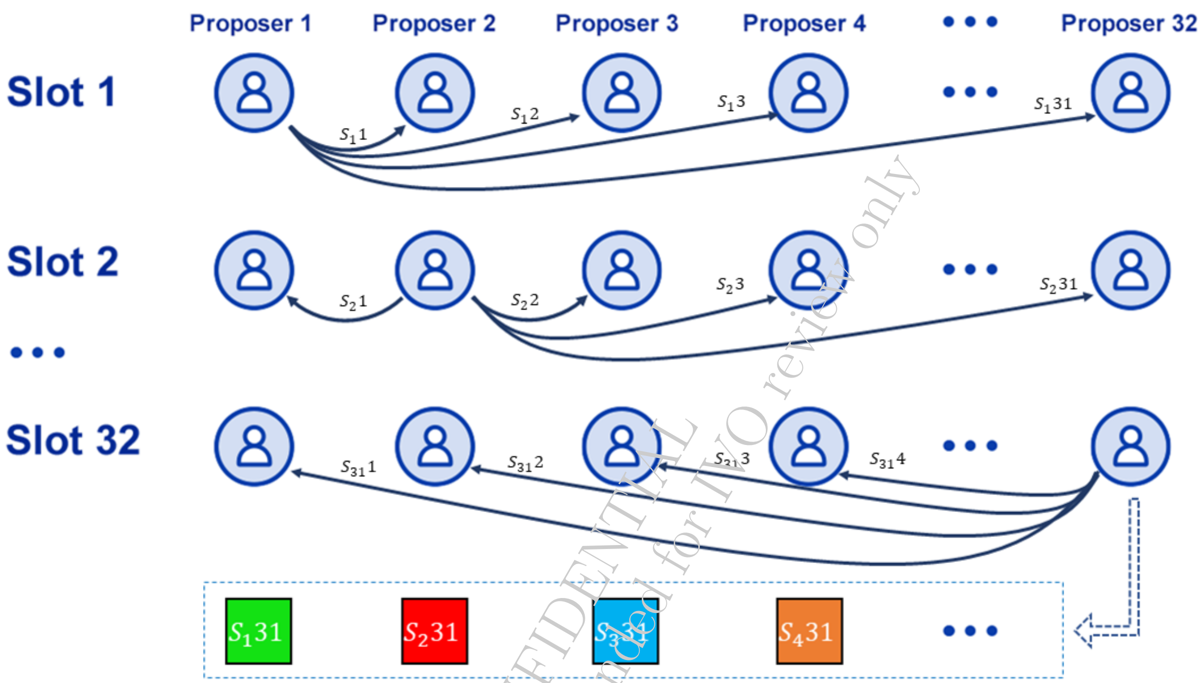


Fig. 15: An illustrate: Shamir's Secret Sharing algorithm in an Epoch.

In Fig. 15, SSS algorithm is applied to Eth2.0. The secret message (S) is 32 bytes secret number. Assuming that a secret message is divided into 31 Shares. In any slot, the proposer will send one Share (S_i) to each other proposer. Similarly until the last slot, at the moment, the last proposer stored 31 Shares of 31 proposers ($S_1 31, ..., S_{31} 31$). And with a single Share of a secret, this one can not recover any secret number. So the last proposer can not predict output of the mix-period, so it can not bias the result of random "seed". At the end of Epoch, all proposers public their Shares and combine them to secret numbers then XOR them to calculate the output.

But we have some conditions to make SSS work, first, $k >= N$ where k is the number of proposers which propose their block. If k is less than N, we can not recover the secret key so the mixed output is not controlled. Then, $k1 > 32 - N$ where $k1$ is the number of the honest validators in an Epoch, if a malicious actor controls more than N proposers in an Epoch, they have many bits of influence. In the future, we will analyze how many stakes a malicious actor needs to control the output "seed". And a complex algorithm is not mentioned above, that is how to send Shares to each other proposers but no one could decrypt Shares which send to other proposers.

V. Conclusion

In this report, we provides an overview of Ethereum networks: from 1.0 to 2.0.

A private Ethereum 2.0 testnet was implemented to show performance's superiority of Ethereum 2.0 versus Ethereum 1.0.

RANDAO mechanism using Shamir's Secret Sharing algorithm to deal with last-revealer attacks was analyzed. For future work, we will further calculate, analyze the results of the SSS algorithm in Eth2.0, and hope that a better algorithm will be proposed.

References

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System.
URL bitcoin.org/bitcoin.pdf

[2] Vitalik Buterin, Ethereum Whitepaper.
URL https://ethereum.org/en/whitepaper/

[3] A. Dorri, S. S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, 2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017 (January) (2017) 618–623. doi:10.1109/PERCOMW.2017.7917634.

[4] M. E. El-Hawary, The smart grid - State-of-the-art and future trends, Electric Power Components and Systems 42 (3-4) (2014) 239–250. doi:10.1080/15325008.2013.868558.

[5] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, G. Dong, GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid, IEEE Access 6 (c) (2018) 9917–9925. doi:10.1109/ACCESS.2018.2806303.

[6] Z. Huang, K. Suankaewmanee, J. Kang, D. Niyato, N. P. Sin, Development of reliable wireless communication system for secure blockchain-based energy trading, JCSSE 2019 - 16th International Joint Conference on Computer Science and Software Engineering: Knowledge Evolution Towards Singularity of Man-Machine Intelligence (2019) 126–130doi:10.1109/JCSSE.2019.8864210.

[7] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, J. Zhao, Toward Secure Blockchain-Enabled Internet of Vehicles: Optimizing Consensus Management Using Reputation and Contract Theory, IEEE Transactions on Vehicular Technology 68 (3) (2019) 2906–2920. arXiv:1809.08387, doi:10.1109/TVT.2019.2894944.

[8] Z. Yang, K. Yang, L. Lei, K. Zheng, V. C. Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE Internet of Things Journal 6 (2) (2019) 1495–1505. doi:10.1109/JIOT.2018.2836144.

[9] W. Wang, D. Niyato, P. Wang, A. Leshem, Decentralized Caching for Content Delivery Based on Blockchain: A Game Theoretic Perspective, IEEE International Conference on Communications 2018-May (2018) 1–6. arXiv:1801.07604, doi:10.1109/ICC.2018.8422547.

[10] A. McAbee, M. Tummala, J. McEachen, Military Intelligence Applications for Blockchain Technology, Proceedings of the 52nd Hawaii International Conference on System Sciencesdoi:10.24251/hicss.2019.726.

[11] Cryptocurrency Prices, Charts And Market Capitalization.
URL https://coinmarketcap.com/

[12] Digiconomist, Ethereum Energy Consumption.
URL https://digiconomist.net/ethereum-energy-consumption

[13] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, E. Dutkiewicz, Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities, IEEE Access 7 (2019) 85727–85745. doi:10.1109/ACCESS.2019.2925010.

[14] Sunny King; Scott Nadal, PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security - CCS'16 1919 (January) (2017) 1–27. arXiv:1703.04057.

[15] Dashcoin Whitepaper.
URL https://github.com/dashpay/dash/wiki/Whitepaper

[16] IEEE Spectrum, Ethereum Plans to Cut Its Absurd Energy Consumption by 99 Percent (2019).
URL https://spectrum.ieee.org/computing/networks/ethereum-plans-to-cut-its-absurd-energy-consumption-by-99-percent

[17] V. Buterin, V. Griffith, Casper the Friendly Finality Gadget, Computing Research Repository (CoRR) (2017) 1–10arXiv:1710.09437.
URL http://arxiv.org/abs/1710.09437

[18] V. Buterin, D. Hernandez, T. Kamphefner, K. Pham, Z. Qiao, D. Ryan, J. Sin, Y. Wang, Y. X. Zhang, Combining GHOST and Casper (2020) 1–38arXiv:2003.03052.
URL http://arxiv.org/abs/2003.03052

[19] Ethereum, Ethereum 2.0 Specifications.
URL https://github.com/ethereum/eth2.0-specs

[20] Prysmatic Labs, Go implementation of the Ethereum 2.0 blockchain.
URL https://github.com/prysmaticlabs/prysm

[21] sigp/lighthouse: Rust Ethereum 2.0 Client.
URL https://github.com/sigp/lighthouse/

[22] Ethereum, Official Go implementation of the Ethereum protocol.
URL https://github.com/ethereum/go-ethereum

[23] H. Zhao, X. Bai, S. Zheng, L. Wang, RZcoin: Ethereum-based decentralized payment with optional privacy service, Entropy 22 (7) (2020) 1–28. doi:10.3390/E22070712.

[24] V. Acharya, A. Eswararao Yerrapati, N. Prakash, Oracle blockchain quick start guide : a practical approach to implementing blockchain in your enterprise, Packt Publishing, 2019.
URL https://cds.cern.ch/record/2699330

[25] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, D. I. Kim, A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks, IEEE Access 7 (c) (2019) 22328–22370. arXiv:1805.02707, doi:10.1109/ACCESS.2019.2896108.

[26] JSON-Remote Procedure Call (RPC).
URL https://eth.wiki/json-rpc/API

[27] Web3.js - Ethereum JavaScript API (2018).
URL https://web3js.readthedocs.io

[28] A. Y. B, Advances in Cryptology - EUROCRYPT 2015 9057 (2015) 817–836. doi:10.1007/978-3-662-46803-6.
URL http://link.springer.com/10.1007/978-3-662-46803-6

[29] Blockchain, Hashrate Distribution and Estimation of Hashrate Distribution Amongst the Largest Mining Pools.
URL https://www.blockchain.com/pools

[30] Blockchain Charts: Average Confirmation Time.
URL https://www.blockchain.com/charts/avg-confirmation-time

[31] Y. Sompolinsky, A. Zohar, Secure High-Rate Transaction Processing in.

[32] Ethereum Average Block Time Chart.
URL https://etherscan.io/chart/blocktime

[33] Deep Chain Reorganization Detected on Ethereum Classic (ETC).
URL https://blog.coinbase.com/

[34] Ethos.dev, The Beacon Chain Ethereum 2.0 explainer you need to read first.
URL https://ethos.dev/beacon-chain/

[35] Randao: Verifiable Random Number Generation.
URL http://random.org/

[36] Chih-Cheng Liang, Minimum Committee Size Explained.
URL https://medium.com/@chihchengliang/minimum-committee-size-explained-67047111fa20

[37] Prysmactic Labs, Medalla Ethereum 2.0 testnet.
URL https://medalla.launchpad.ethereum.org/

[38] Prysmactic Labs, Zinken Ethereum 2.0 testnet.
URL https://zinken.launchpad.ethereum.org/

[39] Ethereum Scan, Ethereum Network Difficulty Chart .
URL https://etherscan.io/chart/difficulty

[40] C. Cr, B. Commitment, Commitment (1980) 1–5.

[41] Manuel Blum, Coin Flipping by Telephone, Proceedings of CRYPTO (1981) 11–15.
URL https://www.cs.cmu.edu/{~}mblum/research/pdf/coin/

[42] P. Dworzanski, A note on committee random number generation, commit-reveal, and last-revealer attacks 1–4.

[43] D. Boneh, J. Bonneau, B. Bünz, B. Fisch, Verifiable delay functions, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 10991 LNCS (2018) 757–788. doi:10.1007/978-3-319-96884-1_25.

[44] Collaboration with the Ethereum Foundation on VDFs.
URL https://filecoin.io/blog/collaboration-on-vdfs/

[45] J. Blömer, How to share a secret, Algorithms Unplugged (2011) 159–168doi:10.1007/978-3-642-15328-0_17.

ASEAN IVO Forum

Advanced Institute of Engineering and Technology

Technical Report

IVO-AVITECH-2020006 November, 2020

Learning based friendly jamming with imperfect CSI for security in MIMO wiretap channel

Bui Minh Tuan, Diep N. Nguyen, Nguyen Linh Trung,
Marwan Krunz, Ta Duc Tuyen, Nguyen Viet Ha,
Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Learning based friendly jamming with imperfect CSI for security in MIMO wiretap channel

Bui Minh Tuan[1], Diep N. Nguyen[2], Nguyen Linh Trung[1], Marwan Krunz[3], Ta Duc Tuyen[1], Nguyen Viet Ha[1], and Eryk Dutkiewicz[2]

[1] AVITECH, VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[2] School of Electrical and Data Engineering, University of Technology Sydney, Australia

[3] Department of Electrical and Computer Engineering, University of Arizona

November, 2020

Abstract

Using deep learning in communication security has been a topic of interest recently. In this report, we proposed a method called learning based friendly jamming (FJ) to guarantee secrecy in MIMO wiretap channels, which is applicable for IoT security due to its low computational complexity at the receivers. Unlike the previous works that require full channel state information (CSI) of legitimate channel at the transmitter, we show that it is possible to rely on this characteristic to construct a robust FJ method with imperfect CSI. We leverage MINE based FJ to demonstrate that it is possible to achieve a security performance comparable with conventional FJ method without CSI. We modify the model to consider the practical challenge in real world systems of the bandwidth constrained feedback channel for providing CSI. Finally, the proposed security scheme can combine MIMO security and detection tasks into a single end-to-end estimation, feedback, encoding, and decoding process, which can be jointly optimized to maximize throughput and minimize block error rate for specific channel conditions.

Index Terms

Physical layer security, autoencoder, friendly jamming, wiretap channel, mutual information neural estimation.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. Introduction

The development of Internet-of-Things (IoT) in Industry 4.0 has brought breakthrough achievements in many areas, e.g., manufacturing, healthcare, and agriculture. Since the data exchange via IoT networks increases dramatically, it also raises many cybersecurity issues. Various approaches have been proposed to mitigate the damage caused by cyberattacks, such as deep learning based cybersecurity threats detection, blockchain based data integrity protection, and physical layer security (PLS) based communication security. In this paper, we consider PLS as an efficient solution for IoT security.

Conventional encryption based security requires infrastructure for distribution of keys and irreversibility of the underlying encrypted function. Recently, PLS has been well developed to also provide secure communication [1]. Confidential communication between legitimate users (*e.g.* between Alice as the transmitter and Bob as the receiver) is wiretapped by illegitimate users (*e.g.* Eve as the eavesdropper). Security is provided thanks to inherit random characteristics of the wireless medium. Compared to conventional encryption based security, PLS is considered as a lightweight solution for secure communication.

Main PLS approaches based on signal-to-noise ratio (SNR), instead of using keys, as reviewed in [2], include: channel coding, channel-based adaptation, artificial noise. Among of them, the use of artificial noise, also called friendly jamming (FJ), to degrade the wiretap channel is more practical thanks to its low computational complexity at the receiver. This approach can be applied to IoT wherein communication devices (sensors, actuators) are desired to be lightweight and low-cost.

In one of the first methods on FJ based security, proposed by Goel and Nagi [3], [4], Alice creates a jamming signal (TxFJ) via precoding, assuming that Alice has the channel state information (CSI), *i.e.* the Alice-Bob channel. Such a precoded FJ signal lies in the null space of Alice-Bob channel, and hence does not affect Bob's reception while degrading Alice-Eve channel. Receiver based friendly jamming (RxFJ) was proposed in a multi-user broadcast channel [5], where Bob is equipped with in-band full-duplex (FD) capabilities. In this work, a non-zero average rate of secrecy can be guaranteed, regardless of the eavesdropper position. Also, the power constraints for the information signal, the TxFJ, and RxFJ signals were investigated in [6]. Further, in scenarios that have interference multiple wiretap channels (MIMO/ multiple links) with distributed computation and limited co-ordinations, a non-cooperative game for modeling the power control problem was proposed in [7].

All the aforementioned methods require that full CSI at the transmitter of their legitimate channel is available at the transmitter to construct FJ signals. Regarding imperfect CSI approach, such as a beam-forming based FJ approach was proposed in [8], requires a high computational capability at Bob to achieve a non-zero average rate of secrecy along with desired BER performance.

In this work, we focus on dealing with the problem related to imperfect CSI, whether being imperfect or unknown, by incorporating a well-known neural network (NN) architecture called variational autoencoder (AE), in the FJ approach for PLS. AE aims to learn a representation (*i.e.* encoding) for a set of data and, from the encoding, reconstructs a representation at the output as close as possible to its original input, by simultaneously optimizing the encoding and decoding functions.

When applied to for PLS, an AE-based method over the Gaussian wiretap channel was proposed in [9]. In this method, Eve is assumed to be equipped with a NN that can cluster the constellation points with high probability. A coding scheme was proposed to make Eve suffer a high block error rate (BLER). However, this method will be undermined when the noise level at Bob is higher than that at Eve. In [10], using AEs to design finite block length wiretap codes was proposed. Also, a multi-objective programming function was proposed to simultaneously minimize the leakage of information to Eve and the BER at Bob. This method has high complexity because the number of needed parameters grows with the code parameters. Using AE for the MIMO Gaussian wiretap channel was demonstrated in [11]. However, the channel matrices at Bob and Eve, and the number of antennas at Eve must be known as well. Besides, the learning in the above AE-based methods for security must take into account all the communication blocks (encoder, channel, decoder, etc.) from the input to the output (*i.e.* end-to-end learning). If there is a way to apply the AE for some but not all blocks, the training time can be shortened and, especially, the design of the method can be more flexible when the end-to-end learning is difficult.

Secrecy capacity is defined as the difference of mutual information (MI) between the legitimate and illegitimate channels [1]. Hence, estimating and optimizing MI can enhance security performance. Recently, mutual information neural estimation (MINE) was proposed by Belghazi *et al.* in [12] and proved to very efficiently optimize MI. From this seminal work, MINE was applied to channel coding on the autoencoder channel [13] by creating a feedback from the output of the channel back to the input of the encoder. The performance of this method is comparable with that of end-to-end learning.

Inspired by the work in [8], we propose a new FJ method for PLS using AE in case of imperfect CSI. However, unlike the approaches based on CSI or channel estimation's error to cancel out FJ at receiver, ours allows the intended transmitter (Tx) and receiver (Rx) learning to null out the FJ signals while only degrade the eavesdropper channel. Further, we leverage MINE to secure the AE based communications when the channel's distribution is unknown. This method can provide comparable secrecy performance to that in [4]. Our work has three main contributions as follows:

- We exploit the generalization capability of neural networks to develop the robust MIMO FJ scheme with imperfect channel knowledge due to the practical issues such as the time-varying nature of channels and limited number of reference signals [14]. Since DL based communications frameworks have powerful generalization ability with respect to the input data sets. We show that it is possible to rely on this characteristic to construct a robust FJ method with imperfect CSI. In other word, our proposed method show the better secrecy capacity compared to the conventional ones when the channel varies or with CSI errors. The great benefits here is we can still maintain secrecy but do not need to sacrifice the capacity and power for channel estimation (for transmitting pilots).
- We develop a new security scheme in which the secrecy optimization can be embedded into the learning process in cases of imperfect CSI. In the first case, only channel distribution is required for the training process. Second, we modify the model to consider the practical challenge in real world systems of the bandwidth constrained feedback channel for providing CSI. This is the case where a compact q-bit representation of the

CSI is available at the transmitter instead of the perfect CSI. The results show that the quantization providing the comparable performance of our system for certain values of q.

- We leverage MINE based FJ to demonstrate that it is possible to achieve a security performance comparable with conventional FJ method without CSI. The training process is performed at the transmitter side to maximize the secrecy capacity between the sampled Tx and Rx signals. Compared to end-to-end learning like AE-based FJ, this security solution not only saves computation resource but also saves energy consumption at the receiver as well. This will be applicable for IoT devices which facing resource constrains.
- Finally, the proposed security scheme can combine MIMO security and detection tasks into a single end-to-end estimation, feedback, encoding, and decoding process, which can be jointly optimized to maximize throughput and minimize block/symbol error rate for specific channel conditions.

The remaining of this paper is organized as follows. In Section II, we briefly introduce the conventional FJ-based PLS method, as the benchmark for our proposed method. In Section III-A1, we propose the AE-based FJ method to achieve secrecy capacity without full CSI. In Section IV, we propose the model which based on MINE for securing communications. The simulation results of secrecy and BLER rate are given and discussed in Section V.

Notation: Vectors and matrices are denoted by bold lowercase and uppercase letters. The absolute value of a real number, the magnitude of a complex number and the complex conjugate transpose are, respectively, denoted by $\|\cdot\|$, $|\cdot|$ and $(\cdot)^{\dagger}$. A complex Gaussian random variable with mean μ and variance σ^2 is denoted by $\mathcal{CN}(\mu, \sigma^2)$. The expectation of a random variable X is denoted by $E[X]$.

II. Friendly jamming with full CSI

We first introduce the background of FJ which is a base line in this work. Further, in this section, the MIMO communication is included as fundamental part for our proposed method.

A. Multiple-input-One-output (MISO) FJ Scheme

The conventional FJ model shows a system with N_T and N_R antennas at the Tx (Alice, A) and Rx (Bob, B), respectively. The eavesdropper (Eve, E) with N_E antennas is also considered. A MISO communication scheme is considered in the IoT scenario where Alice is a controller or gateway and Bob is an IoT device with resource constraints described in Figure 1.

1) System Model: In this model, $N_T = 2, N_R = 1$ and $N_E = 1$, the Tx uses a part of the available power to transmit artificial noise, called a FJ signal. Since it is generated by the Tx, it can be designed in such a way that it only degrades the eavesdropper channel. The conventional works [3], [4] assume that the CSI of the Alice-Bob channel is available at Alice but not that of Eve. The block fading is assumed which means that channel gains are constants in a block and independent distributed in different blocks. At the time k, the channel gain vectors from Alice to Bob and Eve are given by h_k and g_k respectively. Due

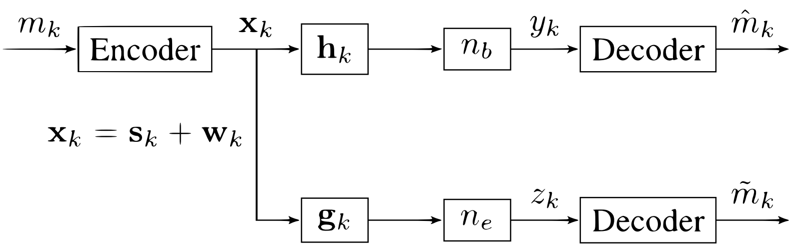


Fig. 1: FJ security model.

to the assumption of block fading, h_k and g_k are constant over a block of a large number of symbols and independent in different blocks. The transmitted signal x_k and the received signals at Bob and Eve are respectively given by

$$\begin{aligned} y_k &= \mathbf{h}_k^\dagger \mathbf{x}_k + n_b, \\ z_k &= \mathbf{g}_k^\dagger \mathbf{x}_k + n_e. \end{aligned} \tag{1}$$

To secure the communication, the Tx chooses the precoding scheme such that the transmitted signal is given by

$$\mathbf{x}_k = \mathbf{s}_k + \mathbf{w}_k = \mathbf{p}_k u_k + \mathbf{w}_k,$$

where the u_k denotes the Gaussian distributed information bearing signal, $\mathbf{w}_k$ is i.i.d. Gaussian FJ. To guarantee secrecy, $\mathbf{w}_k$ is chosen such that $\mathbf{h}_k^\dagger \mathbf{w}_k = 0$ and $\mathbf{h}_k^\dagger \mathbf{p}_k \neq 0$. Hence, $\mathbf{w}_k$ lies in the null space of $\mathbf{h}_k^\dagger$ and thus is canceled out at Bob. The received signals at Bob and Eve are given by

$$\begin{aligned} y_k &= \mathbf{h}_k^\dagger \mathbf{p}_k u_k + n_b, \\ z_k &= \mathbf{g}_k^\dagger \mathbf{p}_k u_k + \mathbf{g}_k^\dagger \mathbf{w}_k + n_e, \end{aligned}$$

where $u_k \sim \mathcal{CN}(0, \sigma_u^2)$, $n_b \sim \mathcal{CN}(0, \sigma_b^2)$, and $n_e \sim \mathcal{CN}(0, \sigma_e^2)$.

2) Security Performance: The performance of this security model is evaluated via the secrecy rate C_s, which is the mutual information difference between the Alice-Bob and Alice-Eve [15], presented as:

$$\begin{aligned} C_s &\doteq I(A,B) - I(A,E) \\ &= \log(1 + \mathrm{SNR_B}) - \log(1 + \mathrm{SNR_E}) \\ &= \log\left(1 + \frac{\left|\mathbf{h}_k^\dagger \mathbf{p}_k\right|^2}{\sigma_b^2}\right) - \log\left(1 + \frac{\left|\mathbf{g}_k^\dagger \mathbf{p}_k\right|^2}{E[\left|\mathbf{g}_k^\dagger \mathbf{w}_k\right|]^2 + \sigma_e^2}\right), \end{aligned} \tag{2}$$

where $I(A,B)$ and $I(A,E)$ are the mutual information between Alice and Bob and Alice and Eve, respectively. Since C_s is a random variable, the average secrecy rate will be examined and our objective function will be

$$\overline{C}_s \doteq \max_{E[x_k {x_k}^\dagger] \leqslant P} E[C_s], \tag{3}$$

where P is the power constraint of the communication system. To solve (3), the authors in [4] chooses $\mathbf{p}_k$ such that $\mathbf{p}_k = \mathbf{h}_k / \left\|\mathbf{h}_k\right\|$. The nonzero secrecy rate is achieved with the assumption of the additive white Gaussian noise (AWGN) on both channel with $\sigma_b^2 = \sigma_e^2$.

The simulation results in [4] proves that the average secrecy rate increases with N_T. This security scheme is established based on the perfect knowledge of CSI and statistics of Eve channel, *e.g.* σ_e^2. In Section III-A1, we propose an FJ method based on the AE that will relax the requirement of full CSI.

B. Multiple-input-Multiple-output (MIMO) FJ Scheme

In this scenario we examine the the generalized model where both the receiver and eavesdropper equipped with multiple antennas. The assumption of $N_T > N_R,\ N_E$ can be seen in IoT scenario similar to MISO case.

1) System Model: In the MIMO FJ scennario the number of antennas at Bob and Eve are N_T and N_E respectively are greater than 1. The channel matrices at time k on Alice-Bob and Alice-Eve channel are $\mathbf{H}_k$ and $\mathbf{G}_k$ respectively. The elements of $\mathbf{H}_k$ and $\mathbf{G}_k$ are assumed to be i.i.d. and independent of each other and unchanged over a block of large number of symbols. The received signals at Bob and Eve are presented as

$$\begin{aligned} \mathbf{y}_k &= \mathbf{H}_k \mathbf{x}_k + \mathbf{n}_b \qquad (4) \\ &= \mathbf{H}_k \mathbf{s}_k + \mathbf{H}_k \mathbf{w}_k + \mathbf{n}_b, \\ \mathbf{z}_k &= \mathbf{G}_k \mathbf{s}_k + \mathbf{G}_k^\dagger \mathbf{w}_k + \mathbf{n}_e. \qquad (5) \end{aligned}$$

Conventional method assumes $\mathbf{H}_k$ is perfectly known at the Tx so the FJ signal is chosen as $\mathbf{H}_k^\dagger \mathbf{w}_k = 0$ then $\mathbf{w}_k = \mathbf{Z}_k \mathbf{v}_k$. Further, if $\mathbf{w}_k$ was chosen fixed, the artificial noise seen by the eavesdropper would be small if $\mathbf{G}_k$ is small. To avoid this possibility, the sequence of $\mathbf{w}_k$ is chosen to be complex Gaussian random vectors in the null space of $\mathbf{H}_k$. In particular, the Tx chooses elements of $\mathbf{v}_k$ to be i.i.d. complex Gaussian random variables with variance σ_v^2, and independent in time as well. It follows that the elements of $\mathbf{w}_k$ are also Gaussian distributed.

2) Security Performance: The covariance of noise at Eve is calculated as

$$\mathbf{K} = (\mathbf{G}_k \mathbf{Z}_k \mathbf{Z}_k^\dagger \mathbf{G}_k)\sigma_v^2 + \mathbf{I}\sigma_e^2. \qquad (6)$$

Then the secrecy capacity C_s is presented as

$$\begin{aligned} C_s &\doteq I(A,B) - I(A,E) \\ &= \log(1 + \mathrm{SNR_B}) - \log(1 + \mathrm{SNR_E}) \\ &= \log\left(det(\mathbf{I} + \mathbf{H}_k \mathbf{Q}_s \mathbf{H}_k^\dagger\right) - \log\left(\frac{det(\mathbf{K} + \mathbf{G}_k \mathbf{Q}_s \mathbf{G}_k^\dagger)}{det(\mathbf{K})}\right), \qquad (7) \end{aligned}$$

where $\mathbf{Q}_s = \mathbf{E}[\mathbf{s}_k \mathbf{s}_k^\dagger]$. Since the the Tx has no information about Alice-Eve's channel so it first chooses Q_s to maximize the capacity of the link to the receiver by using eigenvector transmission. To maximize the secrecy rate, the first term in (7) or the capcity on the legitimate channel is maximized by SVD based method. $\mathbf{H}_k$ is composited as

$$\mathbf{H}_k = \mathbf{U}_k \mathbf{\Gamma}_k \mathbf{V}_k^\dagger.$$

The Tx chooses $\mathbf{S}_k = \mathbf{V}_k \mathbf{r}_k$ and the Rx processes the received signal ($\mathbf{y}_k$) by multiplying it by $\mathbf{U}_k^\dagger$. Then, the equivalent channel to the Rx becomes

$$\tilde{\mathbf{y}}_k = \mathbf{\Gamma}_k^\dagger \mathbf{r}_k + \tilde{\mathbf{n}}_b.$$

then the transmitter chooses $\mathbf{Q}_r$ as

$$\mathbf{Q}_r = \mathbf{E}[\mathbf{r}_k\mathbf{r}_k^\dagger] = diag(\sigma_{r,1}^2, \sigma_{r,2}^2, ..., \sigma_{r,N_T}^2).$$

The $\sigma_{r,i}^2$ is founded by the water filling solution with power constraint $\mathbf{P}_{info} \leqslant \mathbf{P}$ corresponding to the largest singular values of H_k. Then, the minimum guaranteed secrecy capacity is given by

$$C_s = \log\left(det(\mathbf{I} + \mathbf{\Gamma}_k\mathbf{Q}_r\mathbf{\Gamma}_k^\dagger)\right) - \log\left(\frac{det(\mathbf{K}+\mathbf{F})}{det(\mathbf{K})}\right), \tag{8}$$

where $\mathbf{F} = \mathbf{G}_k\mathbf{V}_k\mathbf{Q}_r\mathbf{V}_k^\dagger\mathbf{G}_k^\dagger$. Since C_s is a random variable, the average secrecy capacity C_{sav} and the outage probability can be computed, using Monte Carlo simulations. The objective function is

$$\begin{aligned}\overline{C}_s \doteq \max_{\mathbf{E}[\mathbf{x}_k\mathbf{x}_k^\dagger]\leqslant P} & E[\log\left(det(\mathbf{I} + \mathbf{\Gamma}_k\mathbf{Q}_r\mathbf{\Gamma}_k^\dagger)\right) - \\ & \log\left(\frac{det(\mathbf{K}+\mathbf{F})}{det(\mathbf{K})}\right)], \end{aligned} \tag{9}$$

where the power constraint $\mathbf{E}[\mathbf{x}_k\mathbf{x}_k^\dagger] \leqslant P$ can be rewrite as trace$(\mathbf{V}_k\mathbf{Q}_r\mathbf{V}_k^\dagger + N_{FJ}\sigma_v^2) \leqslant P$, and N_{FJ} denotes the number of antennas used for FJ transmitting. From Equation (6), it can be seen that to guarantee $det(\mathbf{G}_k\mathbf{Z}_k\mathbf{Z}_k^\dagger\mathbf{G}_k)\sigma_v^2 \neq 0$, the transmitter must use at least N_E antennas for FJ signals while the remaining ones can be used for transmitting information signals.

III. MIMO Autoencoder Based Friendly Jamming

We use AE based communications for the goal that is learning to guarantee secrecy. For the communication purpose, the learning process actually is optimization process in which the reconstruction error of the inputs is minimized. In this section, we will examine the secrecy capacity in the practical scenarios as follows:

1) Unknown static CSI, $\mathbf{H}_k$, with the block channel fading assumption.
2) The channel changes and the varies of channel $\mathbf{\Delta H}_k$ modeled through independent identical distributed complex Gaussian distribution with zero mean and scaled identity covariance matrix
3) Learning process with a limited feedback

A. Autoencoder based MIMO Communications scheme

1) Re-parameter conversion: The use of complex valued signals is not available in NN networks. Thus, we re-parameterize the problem using real valued vectors and one-hot mappings as follows:

$$\begin{aligned} \hat{\mathbf{x}}_\mathrm{k} &= \begin{bmatrix} \mathrm{R_e}(\mathbf{x}_\mathrm{k}) \\ \mathrm{I}_m(\mathbf{x}_\mathrm{k}) \end{bmatrix}, \ \hat{\mathbf{x}}_\mathrm{k} = \begin{bmatrix} \mathrm{R_e}(\mathbf{y}_\mathrm{k}) \\ \mathrm{I}_m(\mathbf{y}_\mathrm{k}) \end{bmatrix} \\ \hat{\mathbf{n}}_b &= \begin{bmatrix} \mathrm{R_e}(\mathbf{n}_b) \\ \mathrm{I}_m(\mathbf{n}_b) \end{bmatrix}, \ \hat{\mathbf{H}}_\mathrm{k} = \begin{bmatrix} \mathrm{R_e}(\mathbf{H}_\mathrm{k}) & -\mathrm{I}_m(\mathbf{H}_\mathrm{k}) \\ \mathrm{I}_m(\mathbf{H}_\mathrm{k}) & \mathrm{R_e}(\mathbf{H}_\mathrm{k}) \end{bmatrix}, \end{aligned} \tag{10}$$

where $\hat{\mathbf{x}} \in R^{2N_T}$, $\hat{\mathbf{y}} \in R^{2N_R}$ and $\hat{\mathbf{H}} \in R^{2N_R \times 2N_T}$ are the transmitted received vectors, and the

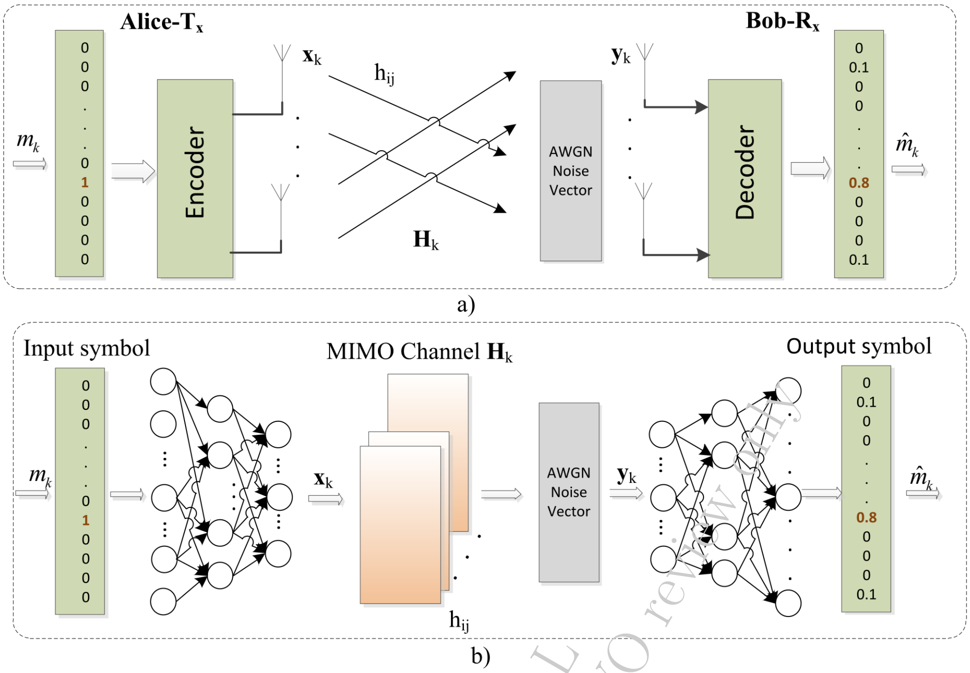


Fig. 2: AE communication scheme for PLS.

channel matrix respectively with the real elements. The relationship between input and out put in (4) becomes [16]:

$$\hat{\mathbf{y}}_k = \hat{\mathbf{H}}_k^{\dagger}\hat{\mathbf{x}}_k + \hat{\mathbf{n}}_b. \tag{11}$$

2) Autoencoder based MIMO communication: The conventional MIMO and MIMO based on AE communication models are presented in Figure 2. To simulate the MIMO channel, we will set up the channel layer in the network illustrated in Figure 2b as the MIMO channel $\mathbf{H}_k$ in Figure 2a. The flat Rayleigh fading as the channel distribution is used in our implementation. At time k, the message $m_k \in M = \{1, 2, \dots, M\}$ is encoded into the transmitted vector s_k. The power constraint is guaranteed by the normalization layer. The receiver blocks at Bob are based on the model in [10] with the last layer using the softmax function. This function gives a probability distribution $\hat{1}_m \in (0,1)^{\mathrm{card}(M)}$ over all of messages (card denotes cardinality), which is fed into the cross-entropy loss function. Then the maximum likelihood is used for estimation of the sent signal [17]. To do it, the cross-entropy loss function is chosen to optimize signals reconstruction error [18]. Hence, the index of the element of $\hat{1}_m$ with the highest probability will be the decoded symbol.

B. Proposed Security scheme

The objective of PLS is to guarantee that no information leakage to Eve while Bob can recover the message without errors [15]. Our model aims to not only guarantee secrecy data capability but also achieve desired BLER.

The proposed AE-based FJ communication and security scheme is illustrated in Figure 3 based on the work in [10], which aims to copy the input message m of the network to

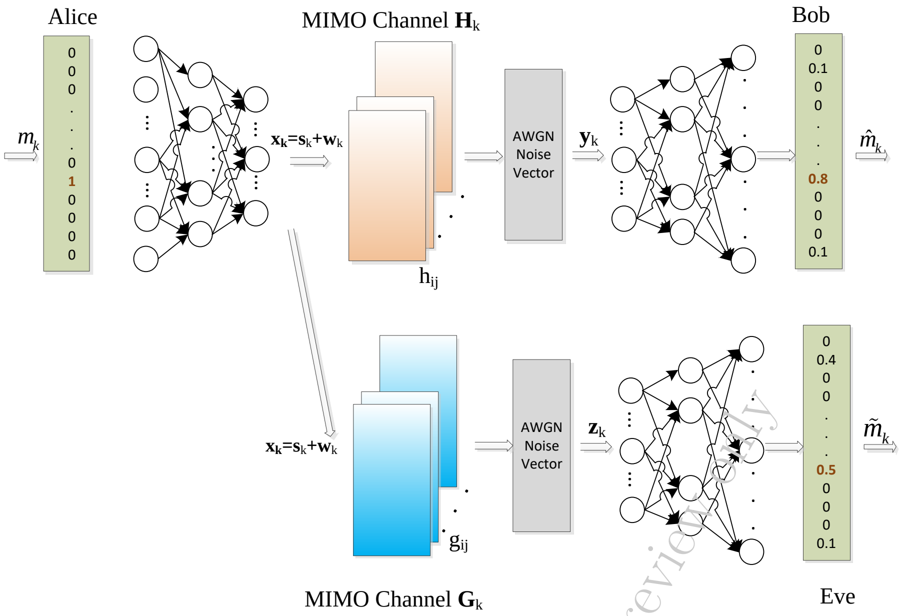


Fig. 3: Autoencoder based Friendly jamming with an example of error decoding at Eve.

its output. The messages are embedded into one-hot vectors 1_m, then encoded to practical modulated information signal, *e.g.* BPSK, by dense layers. The FJ signal is injected into the information signal via the FJ generator layer making the final transmitted signals $\mathbf{x}_k$.

The principle of security by FJ discussed in Section II is using a precoding technique to make the FJ signal orthogonal with the perfectly known channel $\mathbf{H}_k$. However, since $\mathbf{H}_k$ is unknown in our case, we leverage the idea in [8] such that the FJ signal is designed to be orthogonal to the information bearing signal $\mathbf{s}_k$. Once the transmitter and receiver learn to maximize the secrecy capacity it will also minimize BLER. Note that the parameters in the fading layers of Alice-Bob and Alice-Eve, are respectively i.i.d. Alice and Bob communicate and learn the injected FJ signals while Eve tries to decode the message at the same time. Next, we will consider secrecy rate optimization, and security loss function to secure the communication.

C. Problem Formulation

1) Unknown static CSI: In this scenario, the channel coefficients of both Bob's and Eve's channels are unknown remain constant for a coherence interval of transmit symbol periods. The conversion in (10) is used for the case of the received signals $\mathbf{z}_k$ and the channel matrix $\mathbf{G}_k$ at Eve. Similar to conventional FJ method, the transmitted signals $\hat{\mathbf{x}}_k = \hat{\mathbf{s}}_k + \hat{\mathbf{w}}_k$, where $\hat{\mathbf{w}}_k$, and $\hat{\mathbf{s}}_k$ are FJ and information bearing signals individually. From (11), the received signals received at Bob and Eve respectively are

$$\begin{aligned}\hat{\mathbf{y}}_k &= \hat{\mathbf{H}}_k^{\dagger}\mathbf{s}_k + \hat{\mathbf{H}}_k^{\dagger}\hat{\mathbf{w}}_k + \hat{\mathbf{n}}_b,\\ \hat{\mathbf{z}}_k &= \hat{\mathbf{G}}_k^{\dagger}\mathbf{s}_k + \hat{\mathbf{G}}_k^{\dagger}\hat{\mathbf{w}}_k + \hat{\mathbf{n}}_e.\end{aligned} \tag{12}$$

The transmitted signal, $\hat{\mathbf{x}}_k$, satisfies the following power constraint:

$$E[\hat{\mathbf{x}}_k^\dagger \hat{\mathbf{x}}_k] = E[\hat{\mathbf{s}}_k^\dagger \hat{\mathbf{s}}_k] + E[\hat{\mathbf{w}}_k^\dagger \hat{\mathbf{w}}_k] \leqslant P.$$

FJ generation: We implement FJ scheme by setting parameters in the FJ generator layer, which is a shown in Figure 3. As proposed above, the FJ signal $\hat{\mathbf{w}}_k$ is orthogonal with the information bearing signals $\hat{\mathbf{s}}_k$. That design of the FJ generator layer in the autoencoder network is demonstrated in Figure 4. We choose $\hat{\mathbf{s}}_k = \hat{\mathbf{q}}_k \hat{\mathbf{u}}_k$, and $\hat{\mathbf{w}}_k = \hat{\mathbf{v}}_k \hat{\mathbf{d}}_k$, where $\hat{\mathbf{u}}_k$ denotes the information signals. The parameter $\hat{\mathbf{q}}_k$ and $\hat{\mathbf{v}}_k$ are the non-trainable weight and bias in the layer and orthogonal with each other. The elements of $\hat{\mathbf{d}}_k$ is chosen as i.i.d. Gaussian random variable.

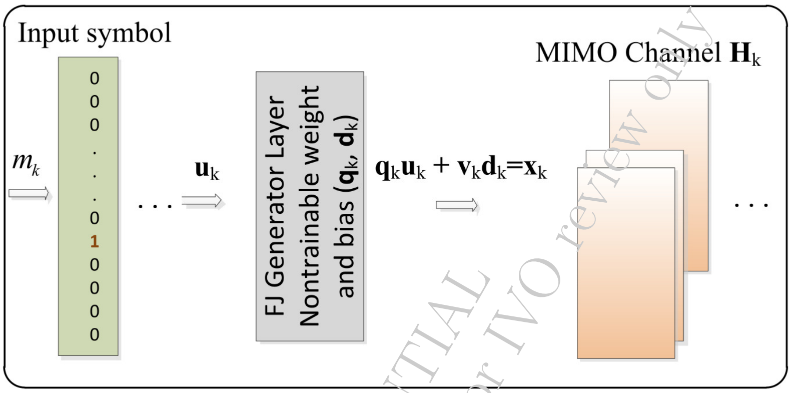

Input symbol
m_k
0
0
0
.
.
.
0
1
0
0
0
0
. . .
$\mathbf{u}_\mathrm{k}$
FJ Generator Layer
Nontrainable weight
and bias ($\mathbf{q}_\mathrm{k}$, $\mathbf{d}_\mathrm{k}$)
$\mathbf{q}_\mathrm{k}\mathbf{u}_\mathrm{k} + \mathbf{v}_\mathrm{k}\mathbf{d}_\mathrm{k} = \mathbf{x}_\mathrm{k}$
MIMO Channel $\mathbf{H}_\mathrm{k}$
. . .

Fig. 4: FJ generator scheme

Secrecy capacity optimization based on AE: A straight-forward approach to optimize the secrecy capacity is computed via the difference of mutual information between Bob's and Eve's channel, as given in (9). However this would be a non-trivial task due to the unknown underlying channel distribution. In this method, we use the multiple cross-entropy loss function, proposed in [9], as given by

$$\begin{aligned} L &= (1-\alpha) H(p_A(s_k), p_B(s_k)) + \alpha H(p_A(s_k), p_E(s_k)) \\ &= (\alpha - 1) \sum_{i=1}^{M} s_{ik} \log \hat{s}_{ik} - \alpha \sum_{i=1}^{M} s_{ik} \log \tilde{s}_{ik}, \end{aligned} \tag{13}$$

where p_A is the probability mass function of the data at Alice, p_B and p_E are the resulting probability mass functions from the softmax function output in the decoders at Bob and Eve, H denotes cross-entropy, and α is a parameter that controls the trade-off between the classification errors at Eve and the communication rate over Bob channel. In other words, minimizing $H(p_A(s_k), p_B(s_k)$ trains the encoder to maximize the output probability of symbol s_{ik}, and thereby reducing the output probability of all other symbols at Bob. In contrast, maximizing $H(p_A(s_k), p_E(s_k))$ forces the system to reduce the output probability of the symbol s_{ik} and therefore randomly forces a higher probability on other symbols $s_{i \neq j}$. Hence, we can gain optimal secrecy capacity by the neural optimization. More specifically, after the training process the secrecy capacity is evaluated by both equation below

$$C_s = I(X_n, Y_n) - I(X_n, Z_n), \tag{14}$$

$$C_s = I(m, \hat{m}) - I(m, \tilde{m}), \tag{15}$$

where X_n, Y_n, and Z_n is the signals samples at transmitter, receiver and eavesdropper, and $m, \hat{m}$, and $\tilde{m}$ are transmitted message at Alice, predicted message at Bob and Eve with the trained model respectively. The secrecy capacity will based on MINE, which is described in detail in Section IV.

2) CSI changes: Next, we consider the estimated channel state information (CSI) at the transmitter side can not be perfect in general. For purposes of our analysis, we denote $\mathbf{H}_k$ to be the imperfect CSI at Tx and the mathematical expression is given by

$$\tilde{\mathbf{H}}_k{=}\mathbf{H}_k + \Delta\mathbf{H}_k, \tag{16}$$

where $\Delta\mathbf{H}_k$ is an i.i.d. complex Gaussian distribution with zero mean and scaled identity covariance matrix given as $\Delta\mathbf{H}_k \sim \mathcal{CN}(0, \rho_e^2 \mathrm{I}_{N_R})$, and $\rho_e^2 = \frac{N_T}{N_P E_P}$ with N_p and $E_p$representing the number and the power of pilot symbols respectively [19]. Unlike conventional optimization method mentioned in Equation (9) based on the SVD of channel, our assumption is that only statistical information of the channel and CSI errors is available at Alice as seen in Figure 5. Comparing to the cases of random static channel and perfect CSI, there are two factors that contribute to the degradation of SNR at Bob leading to the decrease of secrecy capacity as

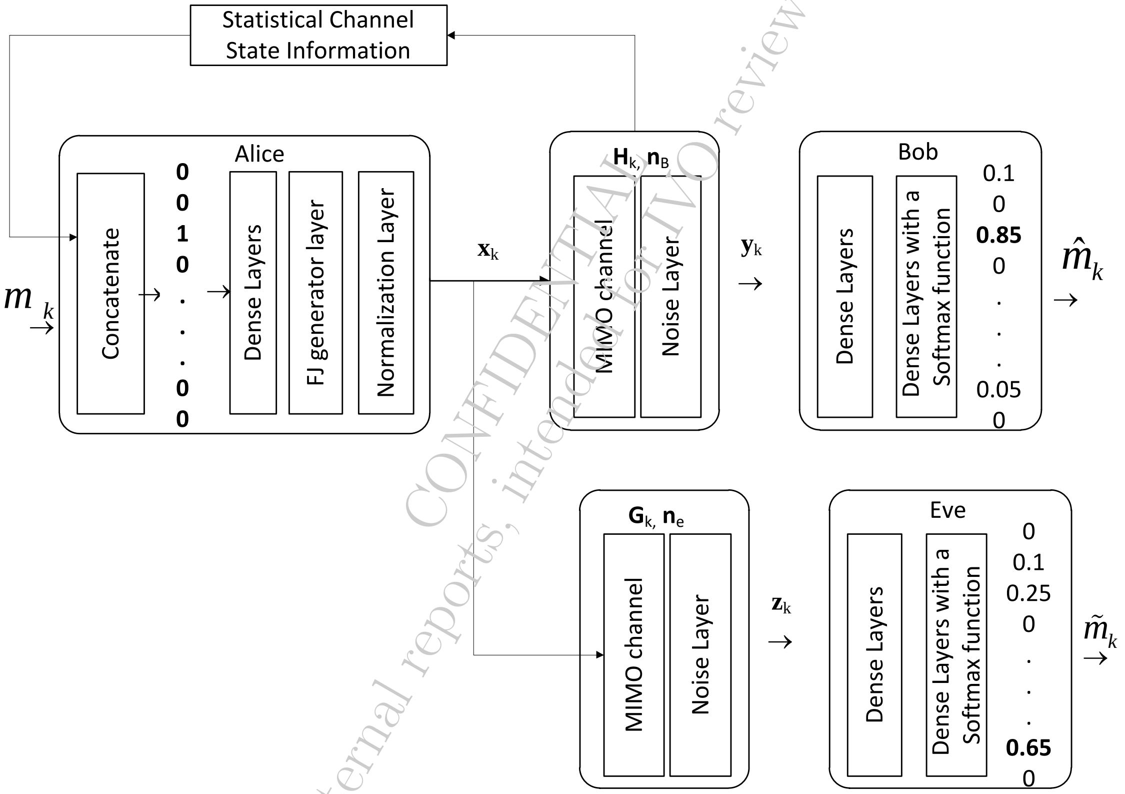


Fig. 5: Learning based FJ with statistical CSI

i) The information bearing signal will leakage to the Eve's channel.
ii) The friendly jamming signals will interfere to the Bob's channel

The impacts of the imperfect CSI on secrecy capacities in the traditional method can be seen in the equation (7) where the error is taken into account as

$$\overline{C}_s \doteq \max_{\mathbf{E}[\mathbf{x}_k\mathbf{x}_k^{\dagger}]\leqslant P} \log\left(det(\mathbf{I}+(\mathbf{H}_k+\Delta\mathbf{H}_k)\mathbf{Q}_s(\mathbf{H}_k+\Delta\mathbf{H}_k)^{\dagger}\right) - \log\left(\frac{det(\mathbf{K}+\mathbf{G}_k\mathbf{Q}_s\mathbf{G}_k^{\dagger})}{det(\mathbf{K})}\right), \tag{17}$$

Secrecy capacity optimization based on AE: The optimization in (17) can be resolved by partitioning the SVD of $\mathbf{H}_k$ and second order perturbation analysis [8]. However, solution requires exponential complexity when the number of antennas increase. Thus, we leverage the non-convex optimization capability provided by NNs, we can directly solve problem with sufficient training data. The architectures of our NN wil be as

- *Input Layer*: Concatenation of $\mathbf{x}_k$ and $\tilde{\mathbf{H}}_k$ which have been converted to real domain from complex domain
- *Hidden Layers*: Simulate the Alice-Bob channel that estimate the mapping function: $h(\mathbf{x}_k, \tilde{\mathbf{H}}_k)$
- *Output Layer*: $\hat{\mathbf{x}}_k$, and the activation function is soft-max for a reconstruction problem
- *Optimizer*: adam is chosen as the optimizer
- *Loss Function*: the Equation (13)

By consider the CSI error as an input for training our network will be train to maximize secrecy capacity and reconstruct the signals as well.

IV. MINE Based Friendly Jamming

1) MINE Preliminary: We first study how MINE [12] to estimate the mutual information between the legitimate users. The mutual information between the two random variable X and Y is given as

$$I(X,Y) = D_{KL}(P(X,Y) \parallel P(X) \otimes P(Y)), \tag{18}$$

which is equivalent to the Kullback-Leibler (KL)divergence, D_{KL}, between the joint probability $P(X,Y)$ and the product of the marginals $P(X) \otimes P(Y)$. In [12], Donsker-Varadhan representation was applied to represent the KL divergence as follows:

$$D_{KL}(P \parallel Q) = \sup_{f:\Omega\rightarrow R} E_P[T] - \log(E_Q[e^T]), \tag{19}$$

where the supremum is taken over all classes of the function f such that the expectation is finite. By choosing the function class, the term on the right hand side of (19) yields an optimal lower bound on the KL-divergence. In MINE, a deep neural network, called statistics network, is chosen as the function family $T_\theta : X \times Y \rightarrow \mathrm{R}$ with parameters $\theta \in \Theta$. We then have the following lower bound on KL-divergence:

$$D_{KL}(P \parallel Q) \geqslant \sup_{T\in F} E_P[T] - \log(E_Q[e^T]). \tag{20}$$

By using the inequality $I(X,Y) \geqslant I_\Theta(X,Y)$ [12], where $I_\Theta(X,Y)$ denotes the mutual information measure defined as

$$I_\Theta(X,Y) = \sup_{\theta\in\Theta} E_{P_{XY}}[T] - \log(E_{P_X \otimes P_Y}[e^T]), \tag{21}$$

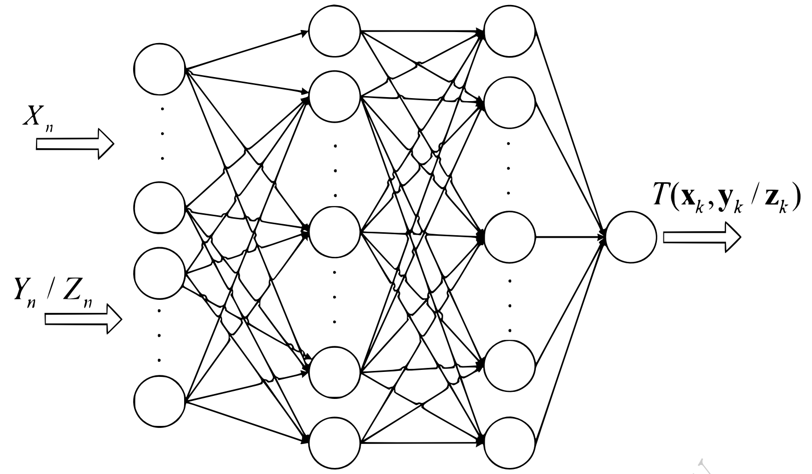


Fig. 6: MI estimator.

we can estimate the mutual information by maximizing $I_\Theta(X, Y)$ in (21). The MI neural estimator T_θ includes two fully connected hidden layers each has 10 nodes, and a linear output node as shown in Figure 6. Where the inputs are the samples from the joint distribution of $P(X_n, Y_n)$ or $P(X_n, Y_n)$, and marginal distributions $P(X)$, $P(Y)$ or $P(Z)$ respectively. We take the samples of these distributions and approximate the expectations by the sample average. The marginal distribution of the input can be derived by shuffling the joint distribution along with the batch axis [12]. Hence, the MI estimation for N samples is given as follows [13]:

$$I_\Theta(X_n, Y_n) = \frac{1}{N}\sum_{i=1}^{N}[T_\theta(\mathbf{x}_i^n, \mathbf{y}_i^n)] - \log\frac{1}{N}\sum_{i=1}^{N}[e^{T_\theta(\mathbf{x}_i^n, \bar{\mathbf{y}}_i^n)}].$$

Similarly, the MI between Alice and Eve or the leakage information to Eve is estimated as

$$I_\Theta(X_n, Z_n) = \frac{1}{N}\sum_{i=1}^{N}[T_\theta(\mathbf{x}_i^n, \mathbf{z}_i^n)] - \log\frac{1}{N}\sum_{i=1}^{N}[e^{T_\theta \mathbf{x}_i^n, \bar{\mathbf{z}}_i^n)}].$$

Then the secrecy rate is then given by

$$C_s^{MINE} = I_\Theta(X_n, Y_n) - I_\Theta(X_n, Z_n). \tag{22}$$

2) MINE-based FJ: In this section, we leverage MINE to optimize channel capacity and secrecy capacity simultaneously. The structures of encoder and MI estimator, I_Θ network, remain unchanged as described in Section III and IV. Regarding security purpose, we will shown that MINE can be used to train the channel with the FJ set up proposed above in Section II, by alternating the maximization of the estimated mutual information over the estimator weights and the encoder weights. The main advantage of that is the training works without explicit knowledge of the channel density function and rather approximates a function/distribution of the channel as AE based FJ. Our security model is presented in Figure 7. To achieve the security requirement and reliable transmission, we use a new security loss function based on MI with the control coefficient β as follows:

$$L_{\text{MINE}} = \beta I_\Theta(X_n, Y_n) + (1-\beta) I_\Theta(X_n, Z_n), \tag{23}$$

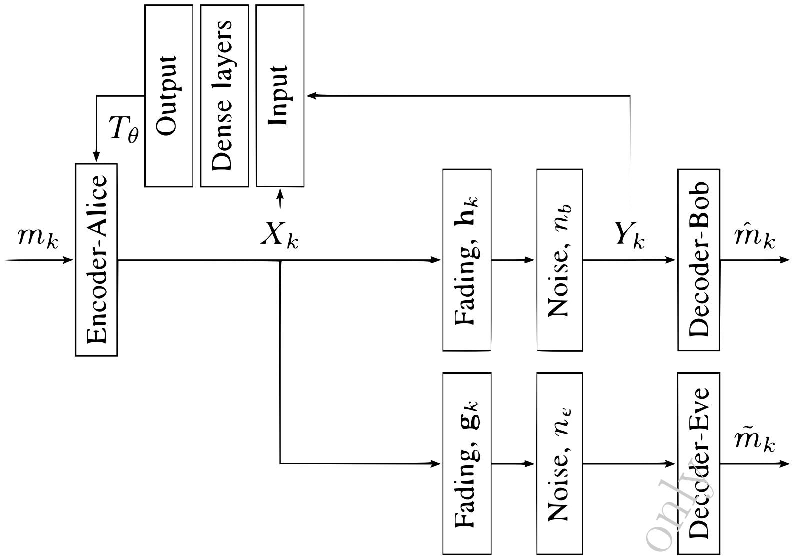


Fig. 7: MI based FJ

From the equation (22) the loss function can be presented as

$$L_{\text{MINE}} = \frac{\beta}{N}\sum_{i=1}^{N}[T_\theta(\mathbf{x}_i^n, \mathbf{y}_i^n)] - \beta \log \frac{1}{N}\sum_{i=1}^{N}[e^{T_\theta(\mathbf{x}_i^n, \bar{\mathbf{y}}_i^n)}]$$
$$-\frac{(1-\beta)}{N}\sum_{i=1}^{N}[T_\theta(\mathbf{x}_i^n, \bar{\mathrm{z}}_i^n)] - (1-\beta)\log \frac{1}{N}\sum_{i=1}^{N}[e^{T_\theta(\mathbf{x}_i^n, \bar{\mathrm{z}}_i^n)}] \tag{24}$$

where the coefficient β represents the trade-off between the communication rate and secrecy rate.

As mentioned above, to avoid approximating the channel probability distribution itself, we will approximate the mutual information between the samples of the channel input and output and optimize the encoder weights, by maximizing the mutual information between them, see Figure 8. For that, we utilize a recent NN estimator of the mutual information [8] and integrate it in our security framework. We are, therefore, independent of the decoder and can reliably train our encoding function using only channel samples.

V. PERFORMANCE EVALUATION

A. Simulation and Neural network architecture

MISO-AEFJ: We use the state-of-the-art deep learning library Tensor Flow with Adam optimizer as tools for the training process. To yield the BLER and expected secrecy capacity $\overline{C}$, we use Monte-Carlo simulation with the flat Rayleigh fading channel [20], [21] by including a fading layer right after the normalization layer. The components of $\mathbf{h}_k$ and $\mathbf{g}_k$ are assumed to be i.i.d. flat Rayleigh with $E(|\mathbf{h}_i|^2) = E(|\mathbf{g}_i|^2) = 1$. Further, we assume that the power constraint P has been normalized by the power of AWGN noise variables n_b and n_e. For this simulation, we have taken a direct SNR of 7 dB in both links during the training phase. For the AEFJ training process, we construct the AE on the Alice-Bob

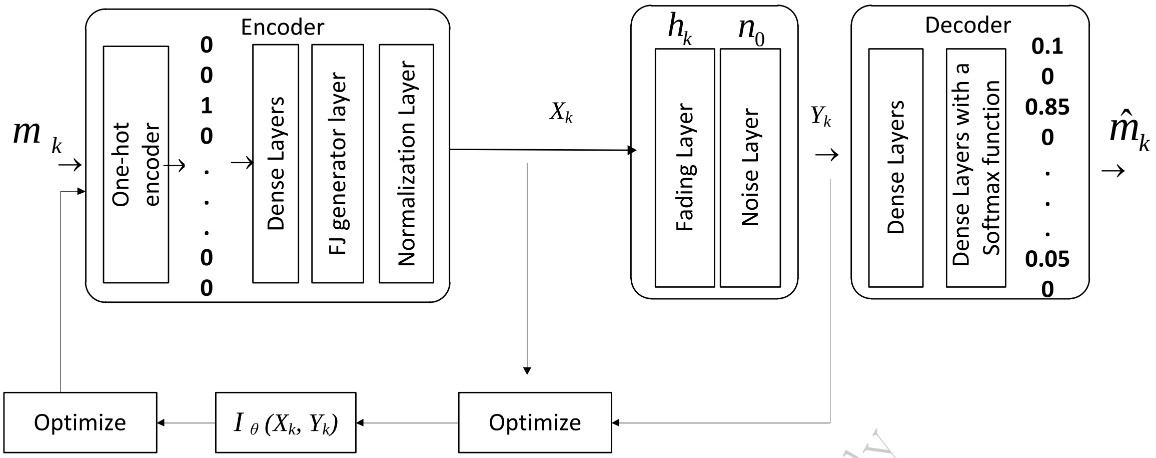


Fig. 8: Optimization at the encoder

link with the layers as in the Figure 3 while Eve has the same neural decoder as that of Bob. The input and output layer has 16 neurons, which represent a symbol when transmit a block of 4 bit. The channel layer includes N_T neurons width representing the time instants on the channel. In the fading layer, the two channel layers are concatenated. The AE at Alice-Bob channel and NN at Eve are trained at the same time as a one-input-two-output NN network with the security loss function (13). Regarding MINE based security approach, the structure of our model is unchanged with the difference in the MINE security function (9). Figure 9 demonstrates the secrecy rate with two different values of α in (13), given the number of transmit antennas $N_T = 2$, each is with 300 iterations and the batch size of $20,000$. We observe that the higher the value of α, the higher the secrecy rate. This means that there is a trade-off between the communication rate and the secrecy rate due to the influence of the FJ signal. Figure 10 shows BLER at Bob and Eve, before and after a secure communication by AEFJ, for some SNR values. We observed a significant increase in BLER of Eve by using AEFJ compared to the one without AEFJ. Meanwhile, Bob's BLER change before and after applying AEFJ is negligible. This means the proposed method shows high performance against the physical-layer security thread by eavesdropping. Figure 11 compares the mutual information at Bob and Eve using the security model in Figure 7. We can see that the information leakage to Eve decreases significantly when using FJ, and a non-zero average secrecy rate can be achieved. Figure 12 demonstrates the performance of the MINE model to deal with the eavesdropping thread in wireless communications. We observe that the BLER performance using MINE is competitive with that in AEFJ with the cross-entropy loss function. Also, regarding the relationship between the average secrecy rate and BLER of the receiver at Bob, Figure 13 shows a decrease in the average secrecy rate when BLER increases. **MIMO-AEFJ:** In this part we compare the secrecy capacity achieved in case of using AEFJ and conventional methods in MIMO channels.

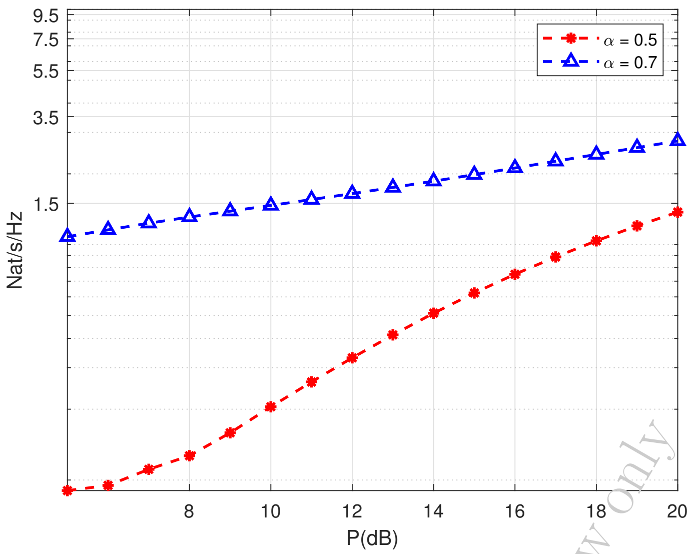


Fig. 9: Average secrecy capacity with different values of α.

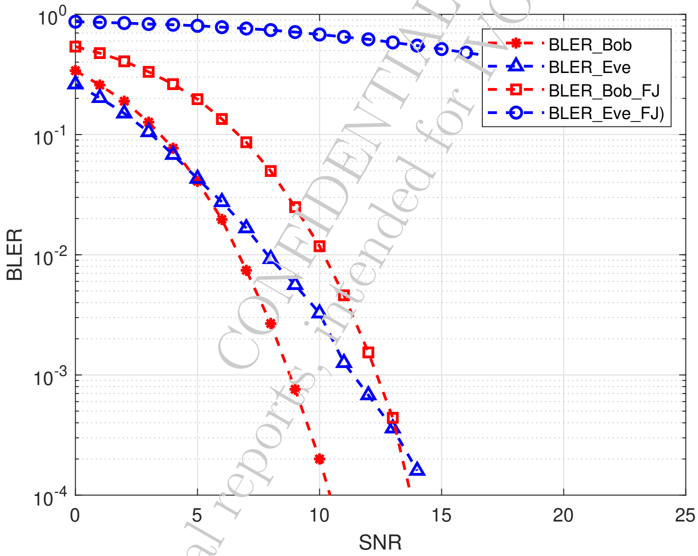


Fig. 10: The block error rate for Autoencoder based FJ.

VI. Conclusion

In this paper, we have presented a new deep learning based friendly jamming approach to deal with the eavesdropping issues in wireless communication. By using the autoencoder at both transmitter and receiver, we have shown that the communication secrecy and the reliability can be achieved simultaneously compared to the conventional model. Further, we leverage the mutual information neural estimator to optimize the security scheme. This modification shows comparable security performance as compared to the autoencoder with the cross-entropy security loss function. In addition, using the mutual information neural estimator, we can optimize the autoencoder independently at the transmitter and the receiver, which is a shortage of the conventional autoencoder model. This method is promising for applications that require

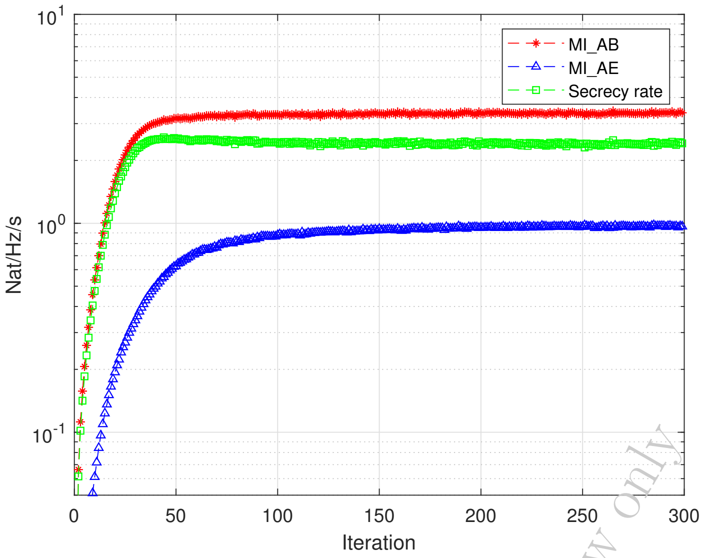


Fig. 11: Mutual information between Alice-Bob and Alice-Eve.

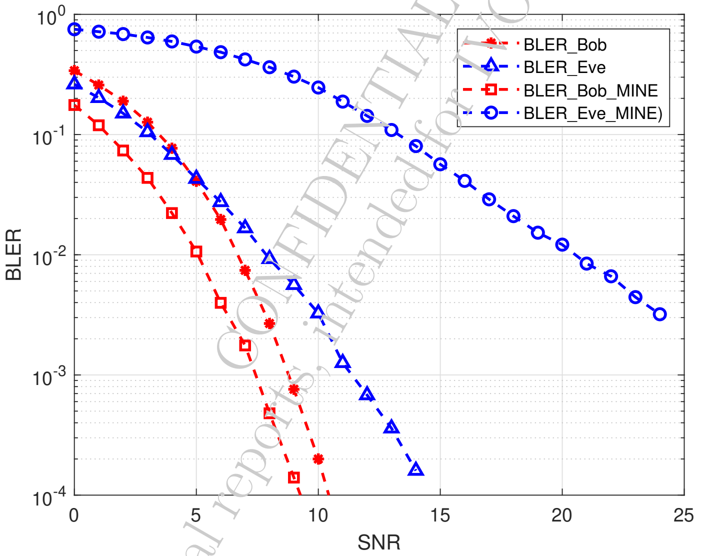


Fig. 12: BLER at Bob and Eve using MINE security.

fast deployment and lightweight security, such as IoT networks.

References

[1] M. Bloch and J. Barros, *Physical-layer security: from information theory to security engineering*. Cambridge University Press, 2011.

[2] J. M. Hamamreh, H. M. Furqan, and H. Arslan, "Classifications and applications of physical layer security techniques for confidentiality: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1773–1828, 2018.

[3] R. Negi and S. Goel, "Secret communication using artificial noise," in *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference*, vol. 3. IEEE, 2005, pp. 1906–1910.

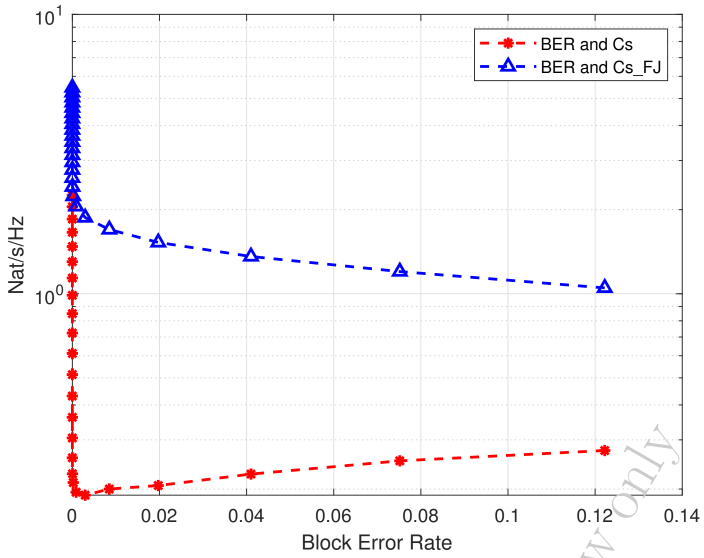


Fig. 13: Secrecy rate and BLER.

[4] S. Goel and R. Negi, "Guaranteeing secrecy using artificial noise," *IEEE transactions on wireless communications*, vol. 7, no. 6, pp. 2180–2189, 2008.

[5] B. Akgun, O. O. Koyluoglu, and M. Krunz, "Exploiting full-duplex receivers for achieving secret communications in multiuser miso networks," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 956–968, 2016.

[6] Y. Liu, J. Li, and A. P. Petropulu, "Destination assisted cooperative jamming for wireless physical-layer security," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 682–694, 2013.

[7] P. Siyari, M. Krunz, and D. N. Nguyen, "Friendly jamming in a mimo wiretap interference network: A nonconvex game approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 601–614, 2017.

[8] A. Mukherjee and A. L. Swindlehurst, "Robust beamforming for security in mimo wiretap channels with imperfect csi," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 351–361, 2010.

[9] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for the gaussian wiretap channel," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[10] K.-L. Besser, C. R. Janda, P.-H. Lin, and E. A. Jorswieck, "Flexible design of finite blocklength wiretap codes by autoencoders," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 2512–2516.

[11] X. Zhang and M. Vaezi, "Deep learning based precoding for the mimo gaussian wiretap channel," *arXiv preprint arXiv:1909.07963*, 2019.

[12] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm, "Mine: mutual information neural estimation," *arXiv preprint arXiv:1801.04062*, 2018.

[13] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," *arXiv preprint arXiv:1903.02865*, 2019.

[14] C. Wang, E. K. S. Au, R. D. Murch, W. H. Mow, R. S. Cheng, and V. Lau, "On the performance of the mimo zero-forcing receiver in the presence of channel estimation error," *IEEE Transactions on Wireless Communications*, vol. 6, no. 3, pp. 805–810, 2007.

[15] C. E. Shannon, "Communication theory of secrecy systems," *Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[16] E. Telatar, "Capacity of multi-antenna gaussian channels," *European transactions on telecommunications*, vol. 10, no. 6, pp. 585–595, 1999.

[17] Y. Bengio, I. Goodfellow, and A. Courville, *Deep learning*. MIT Proess, 2017, vol. 1.

[18] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[19] F. Jiang, C. Li, and Z. Gong, "Accurate analytical ber performance for zf receivers under imperfect channel in low-snr region for large receiving antennas," *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1246–1250, 2018.

[20] B. Sklar, "Rayleigh fading channels in mobile digital communication systems. ii. mitigation," *IEEE Communications magazine*, vol. 35, no. 7, pp. 102–109, 1997.

[21] H. Harada and R. Prasad, *Simulation and software radio for mobile communications*. Artech House, 2002.

Vietnam National University, Hanoi
University of Engineering and Technology

Advanced Institute of Engineering and Technology

Technical Report

UET-AVITECH-2020007 November, 2020

Implementation of a blockchain-based testbed for smart grids

Do Hai Son, Tran Thi Thuy Quynh, Tran Viet Khoa,
Dinh Thai Hoang, Nguyen Linh Trung, Dusit Niyato,
Diep N. Nguyen, Nguyen Viet Ha, Eryk Dutkiewicz

Hanoi, Vietnam

Contents

Implementation of a blockchain based testbed for smart grids

Do Hai Son[1], Tran Thi Thuy Quynh[1], Tran Viet Khoa[1],
Dinh Thai Hoang[2], Nguyen Linh Trung[1], Dusit Niyato[3],
Diep N. Nguyen[2], Nguyen Viet Ha[1], and Eryk Dutkiewicz[2]
[1] AVITECH, VNU University of Engineering and Technology, Hanoi, Vietnam
[2] Electrical and Data Engineering, University of Technology Sydney, Australia
[3] Computer Science and Engineering, Nanyang Technological University, Singapore

November, 2020

Abstract

Blockchain technology has a huge impact on important areas, such as healthcare, finance, vehicles, agriculture, and Internet of Things networks. It ensures the security, integrity and performance of the network. In this project, a smart grid testbed is built to model a blockchain based application. The work includes two phases. In phase 1, we implement the system based on Ethereum network 1.0, using the Proof-of-Work (PoW) consensus mechanism. The network is tested to resist common cyber-attacks such as DDoS and 51% (Double-spending). The experiment is shown in detail. Phase 2 is expected to develop the core network of the system in Ethereum 2.0, using the Proof-of-State consensus mechanism.

Index Terms

Cybersecurity, blockchain, smart grid.

This work is the output of the ASEAN IVO http://www.nict.go.jp/en/asean_ivo/index.html project Cyber-Attack Detection and Information Security for Industry 4.0 and financially supported by NICT http://www.nict.go.jp/en/index.html.

I. Introduction

In Industry 4.0, a growing number of cyber-attack incidents occurred at central servers with serious effects. Hence, data security and integrity have always been attracting great attention. A potential solution is blockchain technology [1] that stores data on a number of network nodes and a consensus mechanism among the nodes to avoid the stored data being manipulated maliciously.

In 2009, Satoshi Nakamoto's development of Bitcoin, which hailed as a radical development in money and currency, being the first example of a digital asset. After Bitcoin, blockchain technologies are blooming with a famous project named "Ethereum". This technology has Bitcoin's valuable characteristics such as decentralization, transparency, immutability, and security-and-privacy. Moreover, it has some great improvements like **smart contract** and **GHOST** (Greedy Heaviest Observed Subtree) protocol [2]. It takes only 15 seconds to confirm a new block (0.25% of Bitcoin) [3], [4].

The Ethereum network is applied in many important applications, such as:

- Smart agriculture: the Ethereum network is used in large farms [5]; it guarantees the safety and transparency of data, e.g. fertilizer, irrigation [6].
- Internet of vehicles: The decentralized properties of th Ethereum technology has been brought safety, privacy, and security to information of the driver [7].
- Healthcare: Blockchain technologies have been adopted by many healthcare systems to enhance the privacy of patient data [8], improve interoperability across devices [9], and maintain an immutable decentralized database of medical records.

In a smart city, the sensor networks are connected together and data is collected based these networks in real-time. A smart grid [10] in Fig. 1 is necessary for controlling the electric

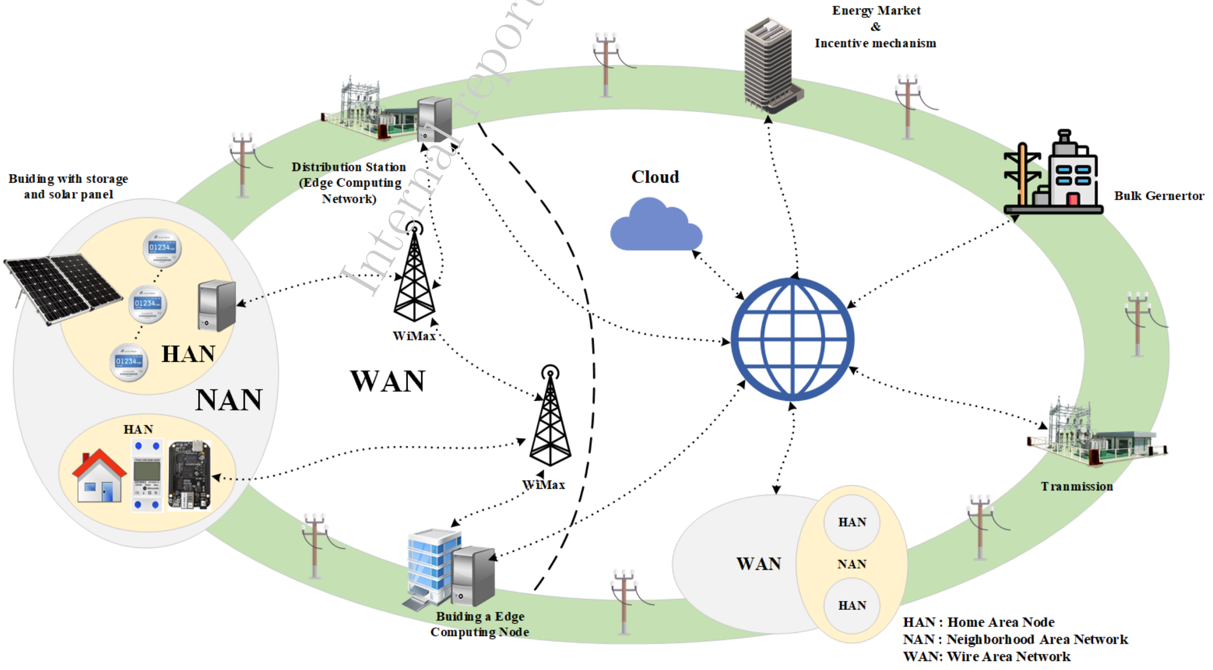


Fig. 1: Model of a smart grid in a smart city.

system of the city. The smart grid can exploit, store, and display the amount of electricity consumption and production. After that, the money is calculated to pay the bill.

Zhuang et al. [11] exploited the blockchain technology and showed the architecture and development platforms of a blockchain-based smart grid for cybersecurity. Huang et al. [12] presented mechanisms of a smart grid in theory and the implementation of the communication system using Sigfox devices but has not applied blockchain into a complete system yet. Gao et al. [13] even finished building a smart contract for their smart grid, but the experimental results have been incomplete.

As shown in Fig. 1, the smart grid is separated into threes layers: Home Area Node (HAN home electrical of the customer), Neighborhood Area Network (NAN - electrical network of several HANs), Wire Area Network (WAN - Connects NANs together). In this report, we implemented a test-bed at HAN level because this layer is easily attacked, hence affecting the customers directly. After that, the DDoS and 51% attacks were verified on the proposed system.

The rest of the report is organized as follows. Section II provides an overview of the Ethereum blockchain technology. Section III describes information about how to implement a smart grid on the Ethereum network. Section IV verifies the ability of the cyber-attack resistance of the test-bed. Finally, Section V summarizes the report.

II. Overview of the Ethereum network

In 2009, Bitcoin is the first example of a digital asset with none of backing, centralized issuer and controller. The underlying blockchain technology attracted a huge attention. In 2013, Ethereum was proposed by Vitalik Buterin, a cryptocurrency researcher and programmer. It provides a blockchain network with a built-in fully fledged Turing-complete programming language, used to create “contracts”. The contracts are used to encode arbitrary state transition functions, allowing users to create any of the systems described above, as well as many others, simply by writing up the logic in a few lines of code [14].

The Ethereum network is divided into seven protocol layers [15]: Storage, Data, Network, Protocol, Consensus, Contract, Application. These protocol layers are described by the structure given in Fig. 2 [15].

Among them, the contract layer and the GHOST protocol will be introduced primarily, because they make up the preeminence between Ethereum over Bitcoin and make Ethereum chosen for the experimental purposes of this report.

Smart contract in the Contract layer is the first highlight of Ethereum, that is simply a piece of code that is running on Ethereum. It is called a “contract” because code that runs on Ethereum can control valuable things like ETH or other digital assets. A smart contract can be built with Solidity language as shown in Fig. 3. The Solidity Compiler will compile the smart contract into Bytecode and Application Binary Interface (ABI). Both of them are packaged into a transaction and deployed into the Ethereum network. Bytecode is an executable code on Ethereum Virtual Machine (EVM) and Contract ABI is an interface to interact with EVM Bytecode. Web3 [16] is a tool provided for users to interact with smart contracts. With the

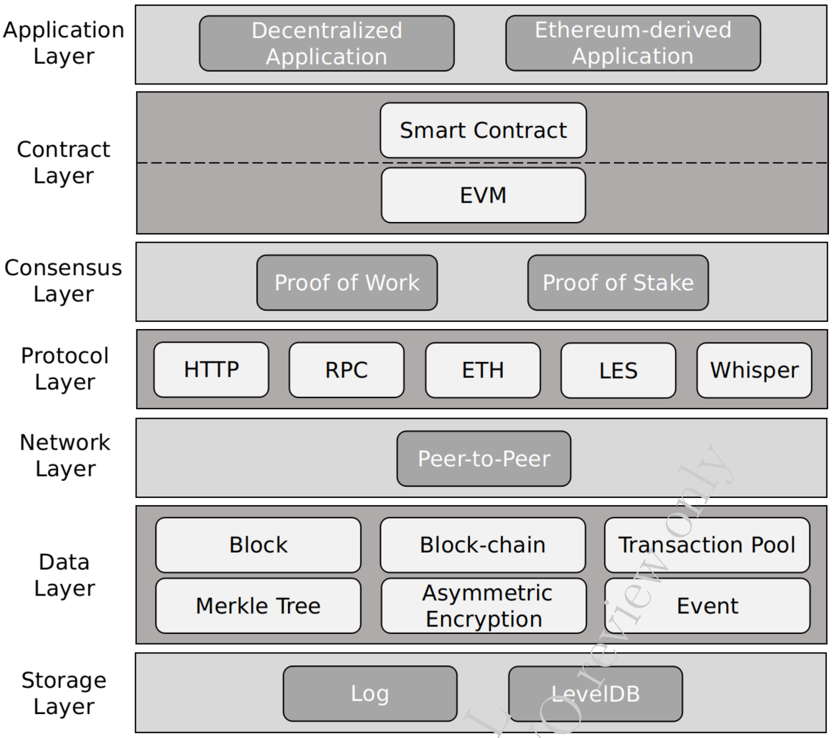


Fig. 2: Ethereum network layer classification [15].

address of smart contract and its ABI, Web3 allows the user to call functions and collect data from the smart contract for their intentions.

GHOST [2] is a Proof-of-Work (PoW) blockchain protocol much like Bitcoin's, except in how it resolves the correct blockchain. As the name entails, instead of the longest chain consensus rule, GHOST follows the path of the subtree with the combined hardest proof of work/difficulty. This can be succinctly visualized as the path of the largest subtree by cardinality and refer to that for simplicity, however, the consensus rule, similar to Bitcoin, is based on aggregate computational power/hashes but of subtrees instead of single links. This protocol makes Ethereum significantly faster than Bitcoin's block confirmation times [14]. In fact, through GHOST, Ethereum's block confirmation time has been significantly improved, as compared to Bitcoin, as shown in Table I.

Table I: Block confirmation time comparisons.

	Bitcoin	**Ethereum**
Type	PoW	PoW
Proposer selection	Base on hash rate	Base on hash rate
Hardware requirement	High	High
Average transaction mining time	Avg around: 98 minutes [4]	Avg around: 15 seconds [3]
Finality time	After 6 new blocks are added Avg around: 10 hours	After 7 new blocks are added Avg around: 2 minutes

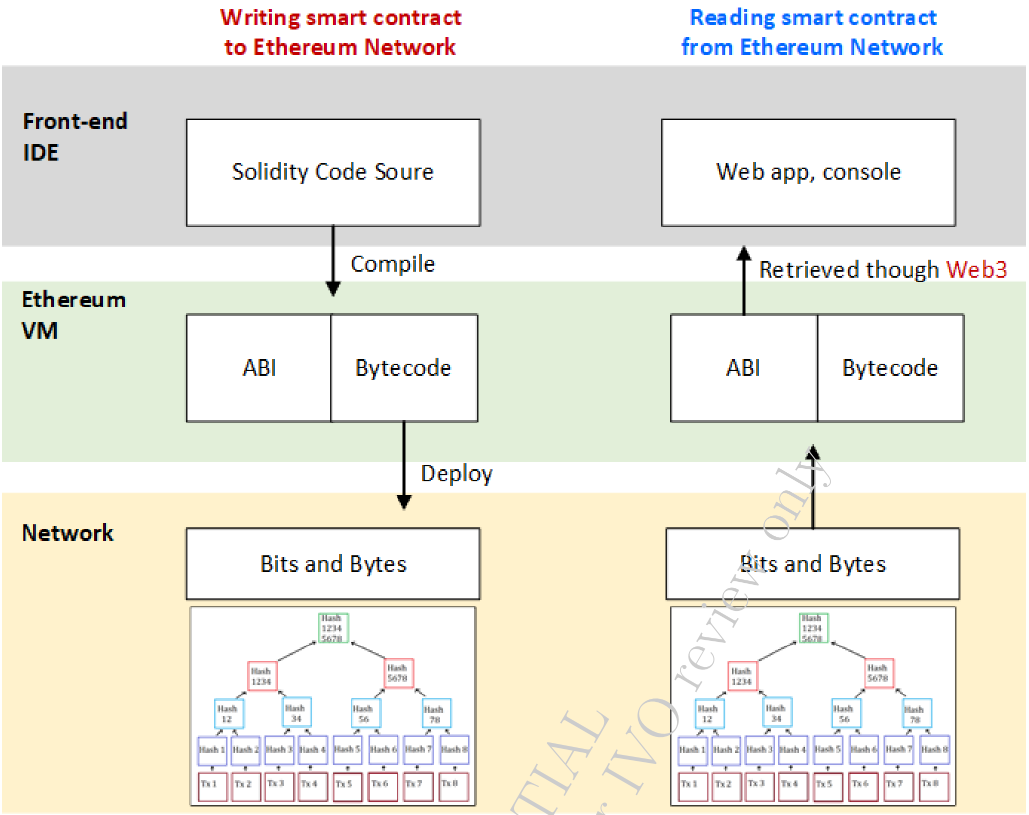


Fig. 3: Smart contract of the Ethereum network.

III. Development of a blockchain-based smart grid

This section presents how to build a smart grid based on both the public Ethereum network and the private Ethereum network.

A. System model

A simple system model of the test-bed is shown in Fig. 4. It includes an electrical load, a smart meter, a BeagleBone Black, a laptop and the Ethereum network.

Smart meter is the next generation of electricity meter. It measures how much electricity has been used, as well as displays this one on a handy in-home display. Furthermore, data collected from the smart meter can be exploited by other IoT devices which use Modbus-RTU protocol. In detail, a smart meter model XTM35sc is used, the parameters displayed on its screen are: voltage, current, active power, consumed power, power factor, frequency. Some specifications are: operating voltage from 161 V to 300 V AC, operating temperature between –10°C and 50°C, operational current range from 0.25A to 50A, operational frequency range is 50/60 Hz. In this proposal, data collected from the smart meter will be exploited, decoded, and transmitted through BeagleBone Black. For the sake of simplicity, only consumed energy data will be collected.

BeagleBone Black (BBB) is a low-cost, community-supported development platform for developers and hobbyists. It is also known as a mini PC suitable for IoT applications. It

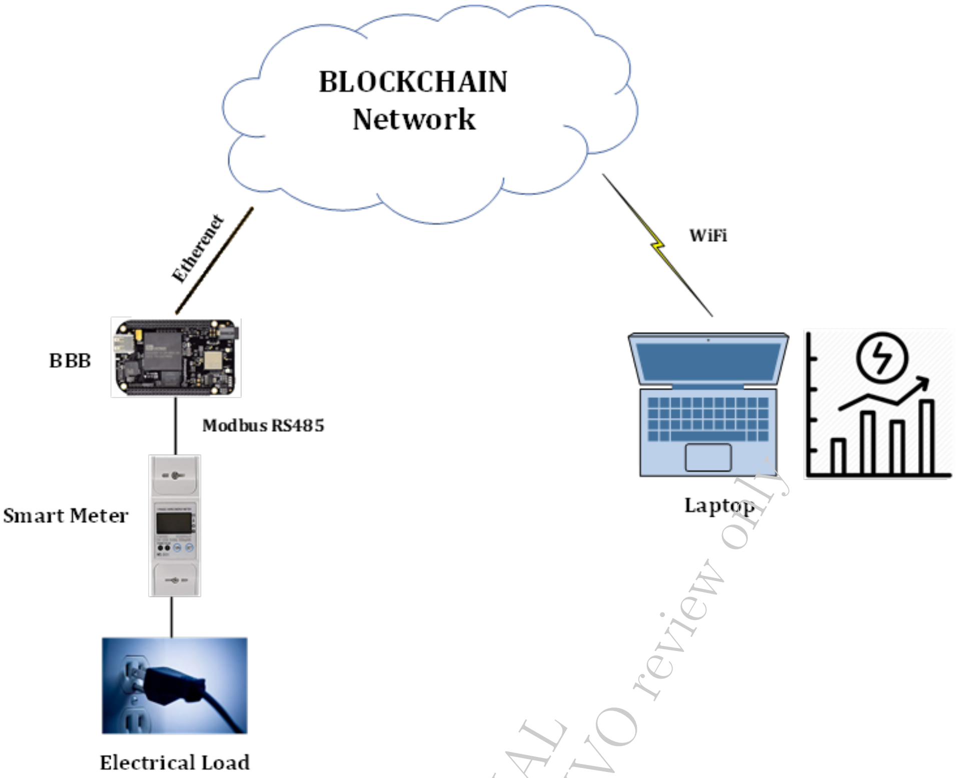


Fig. 4: System model.

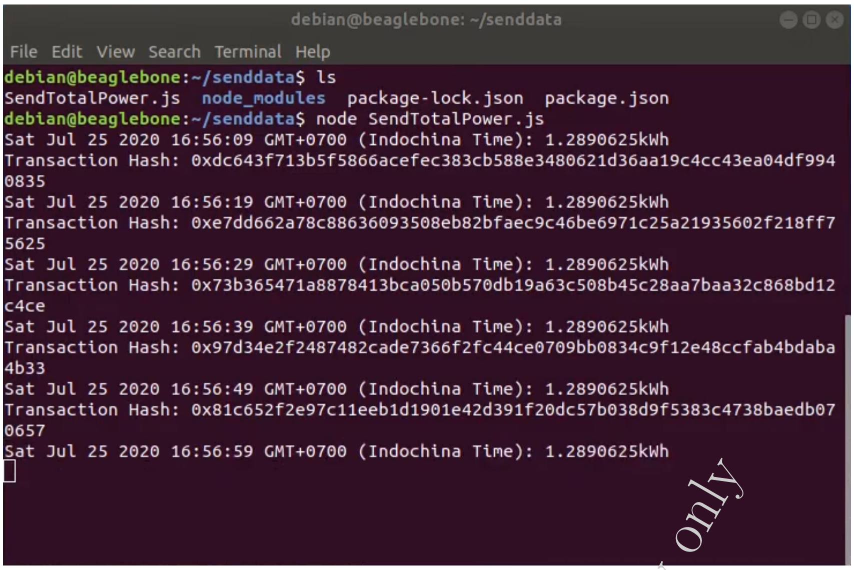
also brings back many advantages such as runs on Linux, integrates both USB and Ethernet ports, and has a lot of expansion pins available to plug the sensors,... In our testbed, BBB will be used to extract data from a smart meter. The extracted data will be packed according to the ABI of the smart contract into a transaction. And then that transaction will be sent to the Ethereum network. In detail, BBB reads the value of the registers in the smart meter, determines what is the consumed energy register, and decodes it into a float32 value following the “big_endian” standard. Having obtained the value of consumed energy, Web3.js is used to pack this one to a transaction. Lastly, Ethereumjs-tx [17] is used to sign that transaction by “private_key” and send it to the network. Fig. 5 shows a screenshot when BBB was sending transactions to the Ethereum network.

A laptop is used to track the amount of electricity comsumption. It is connected to the Ethereum network to take and show data via a dashboard interface.

The Ethereum network is the core of the testbed and is presented next.

B. Ethereum network

Ether is a digital currency with a huge price. At the time of writing this report 1 Ether is approximately $350, so if the testbed is deployed in the main Ethereum network, it will be expensive even though the transactions do not include Ether, but the transaction fee will still be charged, and as mentioned above 10 seconds per transaction will be a big problem. Thus, two ways to deploy the system and send transactions for free in the Ethereum network are as follows.



debian@beaglebone: ~/senddata
File Edit View Search Terminal Help

```
debian@beaglebone:~/senddata$ ls
SendTotalPower.js  node_modules  package-lock.json  package.json
debian@beaglebone:~/senddata$ node SendTotalPower.js
Sat Jul 25 2020 16:56:09 GMT+0700 (Indochina Time): 1.2890625kWh
Transaction Hash: 0xdc643f713b5f5866acefec383cb588e3480621d36aa19c4cc43ea04df994
0835
Sat Jul 25 2020 16:56:19 GMT+0700 (Indochina Time): 1.2890625kWh
Transaction Hash: 0xe7dd662a78c88636093508eb82bfaec9c46be6971c25a21935602f218ff7
5625
Sat Jul 25 2020 16:56:29 GMT+0700 (Indochina Time): 1.2890625kWh
Transaction Hash: 0x73b365471a8878413bca050b570db19a63c508b45c28aa7baa32c868bd12
c4ce
Sat Jul 25 2020 16:56:39 GMT+0700 (Indochina Time): 1.2890625kWh
Transaction Hash: 0x97d34e2f2487482cade7366f2fc44ce0709bb0834c9f12e48ccfab4bdaba
4b33
Sat Jul 25 2020 16:56:49 GMT+0700 (Indochina Time): 1.2890625kWh
Transaction Hash: 0x81c652f2e97c11eeb1d1901e42d391f20dc57b038d9f5383c4738baedb07
0657
Sat Jul 25 2020 16:56:59 GMT+0700 (Indochina Time): 1.2890625kWh
```

Fig. 5: BBB collects data from smart meter and sent to Ethereum network.

1) Public Ethereum testnet: The first way is to use the Ethereum testnet, which is quite popular. It still has the same protocol as the main network, but on the testnet, all Ethers are valueless and users can get them in some faucets [18] on the Internet. Nowadays, there are many Ethereum testnets that have already been deployed, such as Rinkeby [19], Kovan [20], Goerli [21]. In this report, the Rinkeby testnet is used. The easiest way to interact with the Rinkeby network is to use a third-party application, Infura [22], which provides APIs for Web3 and connects to Rinkeby.

Using Infura and testnet are extremely simple and handy. All of the functions are almost the same as those in the main network. However, this is also the reason why they are not suitable for special purposes like the testbed. The functions such as miners, nodes, and other users cannot be controlled. Moreover, Infura also limits the number of API calls per day, so if the smart grid is large, it is necessary to create a private network.

2) Private Ethereum testnet: In order to create a private network, Ethereum's developer team provides an open-source software named Go-ethereum [23] (Geth). Geth is a powerful software, with a lot of functions, for examples: create private nodes, create new accounts, run miner inside nodes, p2p connections, control client for nodes, ...

First of all, initialization parameters for a node must be defined. The work is completed by creating a "genesis.json", this file includes several parameters, for instance: "chainid", "difficulty", "gas limit", "homesteadBlock". Then, the following two commands are for creating and running a private Ethereum node.

```
// To create a new node by genesis file:
$ geth –datadir ./node1 init ./genesis.json
// Run this node with options:
$ geth –datadir ./ –rpc –rpcaddr 172.17.0.2 –rpcport "8545"–rpccorsdomain "*" –allow-insecure-unlock console
```

where –datadir points to where the local node located; –rpc is JSON-RPC [24], that is an address which use to connect with other nodes; –allow-insecure-unlock will unlock permission of accounts that allows them to send transactions. And the rest of the options can be found in [25].

C. Smart contract

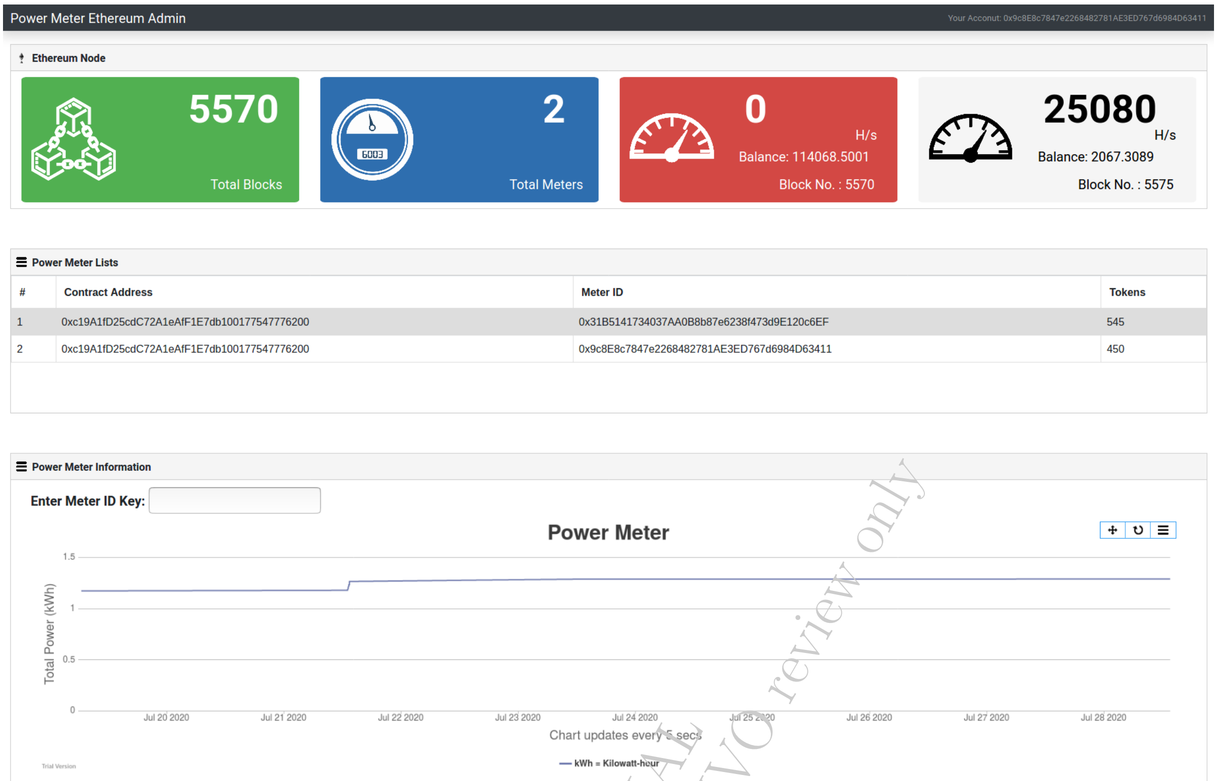
After the Ethereum network has been running, a simple smart contract will be deployed. This contract only collects energy and time-stamps this energy, marks them in sequence, and creates events for other devices that can read data. The smart contract is shown in the following table.

Smart Contract: Smart Meter

```
contract Smartmeter {
   uint public taskCount = 0
   struct Meter {
     uint id;
     string time;
     string energy;
   }
   mapping(uint => Meter) public meter;
   event taskCreated(
     uint id,
     string time,
     string energy
   );
   constructor() public{
     createMeter("0" , "0.0");
   };
   function createMeter(string memory _time, string memory _energy) public {
     ++taskCount;
     meter[taskCount] = Meter(taskCount, _time, _energy);
     emit taskCreated(taskCount, _time, _energy);
   }
}
```

D. Data Monitoring

In our testbed, data monitoring is presented in a dashboard, created by ReactJS [26] and linked to the Ethereum network by json-rpc (default: 127.0.0.1:8545). Web3 collects data from the blockchain network and transfer them to the front-end website. Parameters are selected to display, such as total blocks in the blockchain, number of smart meters, the hash rate of miners, number of tokens in a small table, and a chart of consumed energy by time. All of them are updated manually every 10 seconds. The obtained results are shown in Fig. 6.



Power Meter Ethereum Admin

Your Acconut: 0x9c8E8c7847e2268482781AE3ED767d6984D63411

Ethereum Node

5570 Total Blocks

2 Total Meters

0 H/s
Balance: 114068.5001
Block No. : 5570

25080 H/s
Balance: 2067.3089
Block No. : 5575

Power Meter Lists

#	Contract Address	Meter ID	Tokens
1	0xc19A1fD25cdC72A1eAfF1E7db100177547776200	0x31B5141734037AA0B8b87e6238f473d9E120c6EF	545
2	0xc19A1fD25cdC72A1eAfF1E7db100177547776200	0x9c8E8c7847e2268482781AE3ED767d6984D63411	450

Power Meter Information

Enter Meter ID Key:

Fig. 6: Dashboard of the testbed using private testnet.

IV. Cyber-attacks on the developed blockchain testbed

Assuming a hacker would attack the blockchain network, this section will check the performance of the designed Ethereum network against two types of attacks: Distributed Denial-of-Service (DDoS) attack to disrupt the network service, and 51% (Double-spending) attack to steal cryptocurrency.

A. *Distributed Denial-of-Service (DDoS) attack*

DDoS is a common cyber-attack that causes a machine or a network resource unavailable to its intended users by temporarily or indefinitely disrupting the connection between a host and network. Denial-of-service typically takes place when the targeted machine or resource is flooded with a huge number of IP packet requests in an attempt to overload the network and prevent some or all legitimate requests from being fulfilled.

To implement this attack on the testbed, 4 nodes have been created by the method create private node as mentioned in Section III-B2, and each node resides in a virtual machine created by using Docker [27]. After they were all running, in console of Geth, run the following command to connect them:

```
> admin.addPeer(enode)
```

where enode is a parameter that defines a specific node and can be obtained by the command: “admin.nodeInfo.enode”.

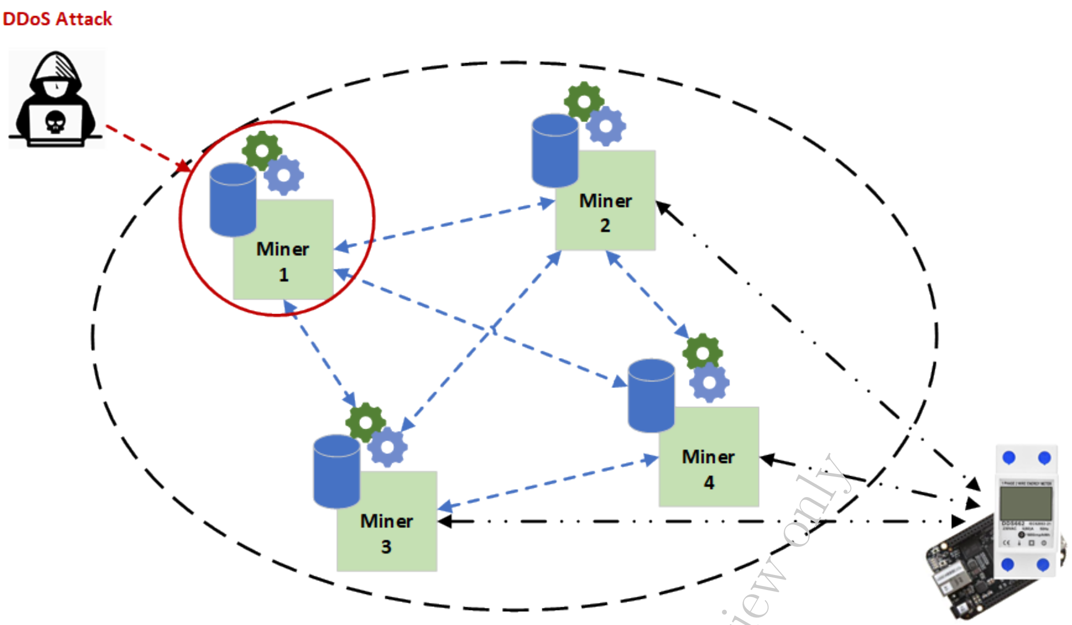


Fig. 7: DDoS attack to the Ethereum network.

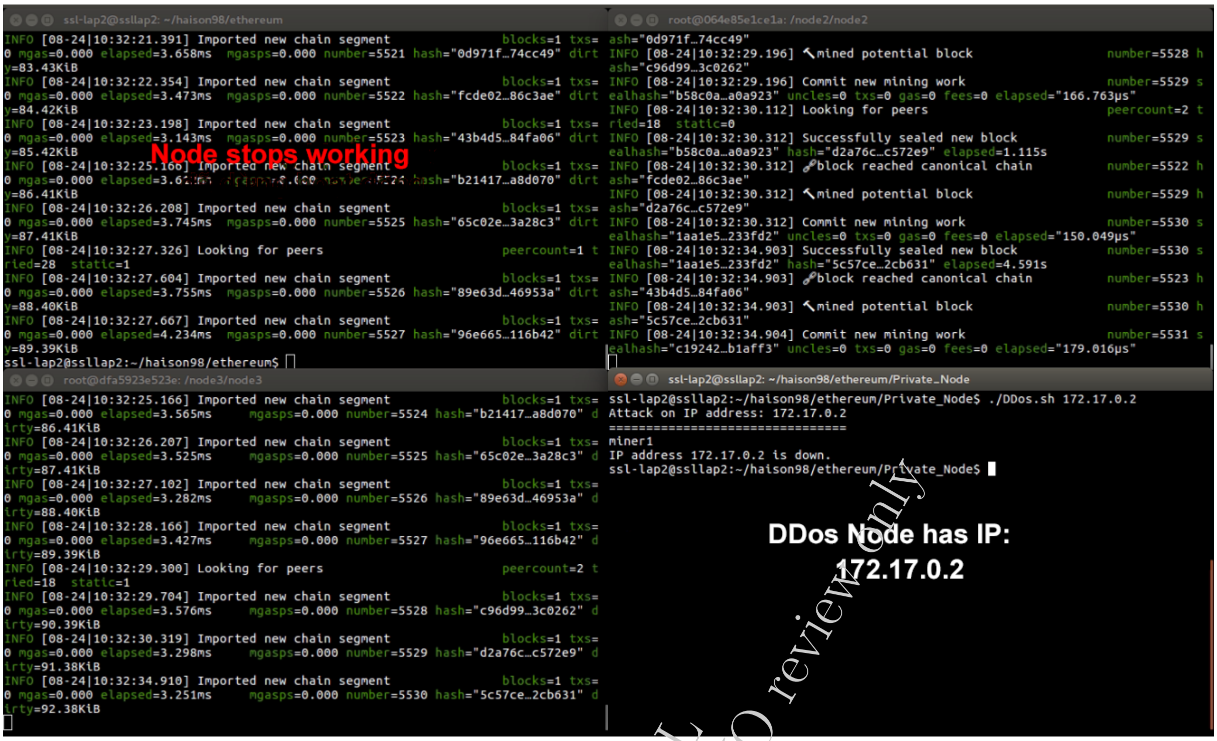
Fig. 7 illustrates a DDoS attack to Ethereum network.

Because each node runs in a virtual machine, turning off a node by its IP address has the same result as a DDoS attack at the node. To do that, a short script is written, and results, tested in the testbed, are shown in Fig. 8. The attacked node in the first terminal stoped working, but the other nodes was still working fine without interruption. In conclusion, due to the decentralization of the blockchain network, the system on the testbed was not affected by the DDoS attack.

B. 51% attack

Page 3 of Satoshi Nakamoto's whitepaper [1] (on "Bitcoin: A Peer-to-Peer Electronic Cash System") states the following: "If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains.". However, in small to medium Ethereum networks, there is a risk that an attacker can control majority of CPU power, that can be a 51% attack (Double-spend). To illustrate, in early 2009, there was the theft of 219500 ETC ($1.1 M) at Coinbase [28].

In our study, to implement the attack, a new Token is created in ERC 20 Token standard (EIP-20) [29], that could present for money in the main network. This work divides these Tokens into meters A and B. Fig. 9 shows the attack, where Pools A and B are two nodes including miner. Thus, meter A sends 50 Tokens to meter B but Pool A fakes this transaction from 50 to 100 Tokens. Because Pools A and B do not connect to each other, they will confirm new blocks by their own chains. And according to the assumption, Pool A has 51% total hashrate of the network. So, Pool A will confirm new blocks faster, and in Fig. 10 assume that the chain in Pool A longer than Pool B is five blocks. The longest chain is always considered the "true" chain, therefore when Pools A and B connect to each other, the chain of Pool B with honest transaction will be rejected.



ssl-lap2@ssllap2: ~/haison98/ethereum

```
INFO [08-24|10:32:21.391] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.658ms  mgasps=0.000 number=5521 hash="0d971f…74cc49" dirt
y=83.43KiB
INFO [08-24|10:32:22.354] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.473ms  mgasps=0.000 number=5522 hash="fcde02…86c3ae" dirt
y=84.42KiB
INFO [08-24|10:32:23.198] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.143ms  mgasps=0.000 number=5523 hash="43b4d5…84fa06" dirt
y=85.42KiB
INFO [08-24|10:32:25.166] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.6[illegible] hash="b21417…a8d070" dirt
y=86.41KiB
INFO [08-24|10:32:26.208] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.745ms  mgasps=0.000 number=5525 hash="65c02e…3a28c3" dirt
y=87.41KiB
INFO [08-24|10:32:27.326] Looking for peers                        peercount=1 t
ried=28  static=1
INFO [08-24|10:32:27.604] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.755ms  mgasps=0.000 number=5526 hash="89e63d…46953a" dirt
y=88.40KiB
INFO [08-24|10:32:27.667] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=4.234ms  mgasps=0.000 number=5527 hash="96e665…116b42" dirt
y=89.39KiB
ssl-lap2@ssllap2:~/haison98/ethereum$ 
```

root@064e85e1ce1a: /node2/node2

```
ash="0d971f…74cc49"
INFO [08-24|10:32:29.196] ⛏mined potential block                  number=5528 h
ash="c96d99…3c0262"
INFO [08-24|10:32:29.196] Commit new mining work                   number=5529 s
ealhash="b58c0a…a0a923" uncles=0 txs=0 gas=0 fees=0 elapsed="166.763µs"
INFO [08-24|10:32:30.112] Looking for peers                        peercount=2 t
ried=18  static=0
INFO [08-24|10:32:30.312] Successfully sealed new block            number=5529 s
ealhash="b58c0a…a0a923" hash="d2a76c…c572e9" elapsed=1.115s
INFO [08-24|10:32:30.312] 🔗block reached canonical chain           number=5522 h
ash="fcde02…86c3ae"
INFO [08-24|10:32:30.312] ⛏mined potential block                  number=5529 h
ash="d2a76c…c572e9"
INFO [08-24|10:32:30.312] Commit new mining work                   number=5530 s
ealhash="1aa1e5…233fd2" uncles=0 txs=0 gas=0 fees=0 elapsed="150.049µs"
INFO [08-24|10:32:34.903] Successfully sealed new block            number=5530 s
ealhash="1aa1e5…233fd2" hash="5c57ce…2cb631" elapsed=4.591s
INFO [08-24|10:32:34.903] 🔗block reached canonical chain           number=5523 h
ash="43b4d5…84fa06"
INFO [08-24|10:32:34.903] ⛏mined potential block                  number=5530 h
ash="5c57ce…2cb631"
INFO [08-24|10:32:34.904] Commit new mining work                   number=5531 s
ealhash="c19242…b1aff3" uncles=0 txs=0 gas=0 fees=0 elapsed="179.016µs"
```

root@dfa5923e523e: /node3/node3

```
INFO [08-24|10:32:25.166] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.565ms     mgasps=0.000 number=5524 hash="b21417…a8d070" d
irty=86.41KiB
INFO [08-24|10:32:26.207] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.525ms     mgasps=0.000 number=5525 hash="65c02e…3a28c3" d
irty=87.41KiB
INFO [08-24|10:32:27.102] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.282ms     mgasps=0.000 number=5526 hash="89e63d…46953a" d
irty=88.40KiB
INFO [08-24|10:32:28.166] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.427ms     mgasps=0.000 number=5527 hash="96e665…116b42" d
irty=89.39KiB
INFO [08-24|10:32:29.300] Looking for peers                        peercount=2 t
ried=18  static=1
INFO [08-24|10:32:29.704] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.576ms     mgasps=0.000 number=5528 hash="c96d99…3c0262" d
irty=90.39KiB
INFO [08-24|10:32:30.319] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.298ms     mgasps=0.000 number=5529 hash="d2a76c…c572e9" d
irty=91.38KiB
INFO [08-24|10:32:34.910] Imported new chain segment               blocks=1 txs=
0 mgas=0.000 elapsed=3.251ms     mgasps=0.000 number=5530 hash="5c57ce…2cb631" d
irty=92.38KiB
```

ssl-lap2@ssllap2: ~/haison98/ethereum/Private_Node

```
ssl-lap2@ssllap2:~/haison98/ethereum/Private_Node$ ./DDos.sh 172.17.0.2
Attack on IP address: 172.17.0.2
================================
miner1
IP address 172.17.0.2 is down.
ssl-lap2@ssllap2:~/haison98/ethereum/Private_Node$ 
```

Fig. 8: DDoS attack result.

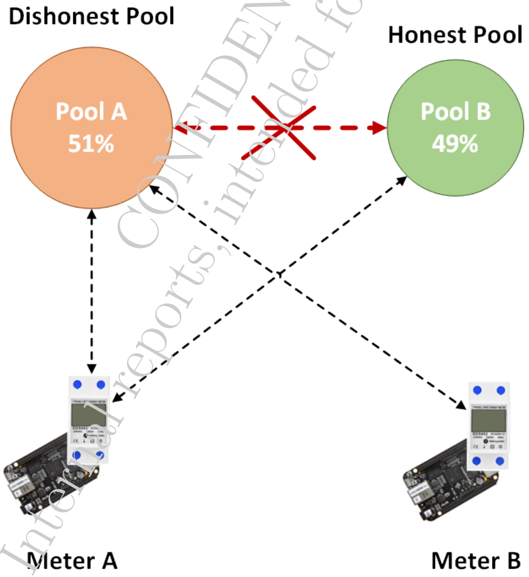


Fig. 9: 51% attack (double spend).

As a result, this is really a risk and exists alongside PoW. For example, the top five mining pools control up to 55.9% [30] total hashrate of the Bitcoin network. This is the most serious issue of PoW-based blockchain networks because it is against the decentralized spirit of blockchain technology.

There are two ways to solve this issue. Firstly, the organizations that run large pools need to be certificated. Secondly, change the core of Ethereum from Proof-of-Work to Proof-of-Stake [31].

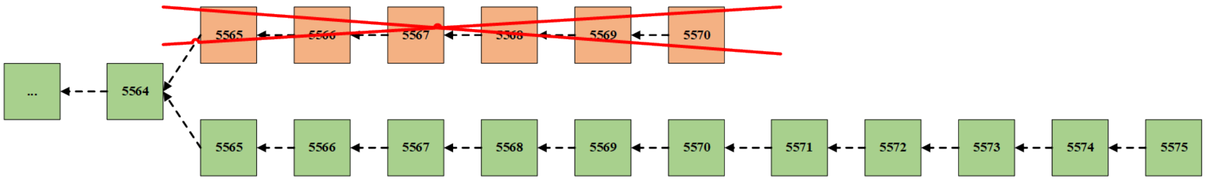


Fig. 10: Result of 51% attack.

V. Conclusion

In this report, we have developed a simple testbed to provide experiences of blockchain technology which built in smart grid application. The test-bed shown the storage and display data of the smart grid from end to end. Two cyber-attacks (DDoS and 51%) were also verified on the testbed.

For future work, the system model will be expanded by using several IoT gateways. The core of Euthereum 2.0 is considered implementing in phase 2.

Above, we have designed a simple blockchain network for the purpose of enhancing security of a practical IoT application (smart grid) using an existing blockchain method. The focus is on the implementation of the blockchain technology.

We would like to extend it to a more complex and practical scenario of a smart factory. Such a complex scenario is to experiment different types of security attacks on different IoT devices in practice. In addition, security methods developed in this project, such as federated transfer learning for cyberattack detection can be implemented in this configuration. This scenario is depicted in Figure 11.

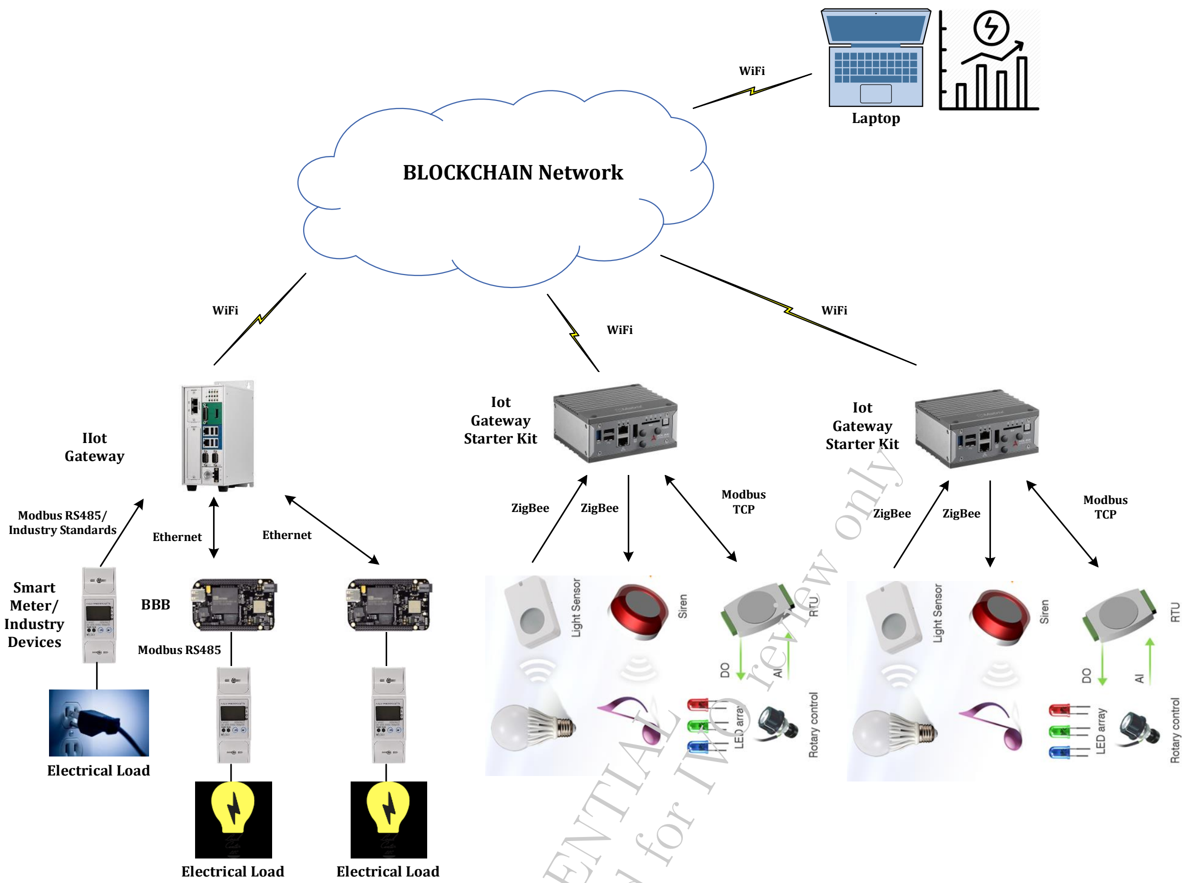


Fig. 11: Complex system model (for future work).

.

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System." [Online]. Available: bitcoin.org/bitcoin.pdf

[2] Y. Sompolinsky and A. Zohar, "Secure High-Rate Transaction Processing in."

[3] "Ethereum Average Block Time Chart." [Online]. Available: https://etherscan.io/chart/blocktime

[4] "Blockchain Charts: Average Confirmation Time." [Online]. Available: https://www.blockchain.com/charts/avg-confirmation-time

[5] G. Mirabelli and V. Solina, "Blockchain and agricultural supply chains traceability: Research trends and future challenges," *Procedia Manufacturing*, vol. 42, no. 2019, pp. 414–421, 2020. [Online]. Available: https://doi.org/10.1016/j.promfg.2020.02.054

[6] M. Shyamala Devi, R. Suguna, A. S. Joshi, and R. A. Bagate, "Design of IoT Blockchain Based Smart Agriculture for Enlightening Safety and Security," *Communications in Computer and Information Science*, vol. 985, no. May, pp. 7–19, 2019.

[7] R. Jabbar, M. Kharbeche, K. Al-Khalifa, M. Krichen, and K. Barkaoui, "Blockchain for the internet of vehicles: A decentralized IoT solution for vehicles communication using ethereum," *Sensors (Switzerland)*, vol. 20, no. 14, pp. 1–27, 2020.

[8] Drew Ivan, "Blockchain-based Method for Secure Storage of Patient Records," *ONC/NIST Use of Blockchain for Healthcare and Research Workshop*, no. August, p. 11, 2016. [Online]. Available: https://www.healthit.gov/sites/default/files/9-16-drew{_}ivan{_}20160804{_}blockchain{_}for{_}healthcare{_}final.pdf

[9] W. R. Hersh, A. M. Totten, K. B. Eden, B. Devine, S. Z. Kassakian, S. S. Woods, M. Daeges, M. Pappas, S. Mcdonagh, W. R. Hersh, X. Yue, H. Wang, D. Jin, M. Li, W. Jiang, B. Almoaber, K. Edward, K. Edward, M. Azarm-daigle, C. Kuziemsky, L. Peyton, C. Wood, B. Winton, K. Carter, S. Benkert, L. Dodd, J. Bradley, M. Ave, A. Mcclung, N. Archer, K. Peterson, R. Deeduvanu, P. Kanjamala, and K. Boles, "Health Information Dissemination from Hospital To Community Care : Current State And Next Steps In Ontario," *Journal of Medical Systems*, vol. 63, no. 50, pp. 425–432, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.08.363{%}5Cnhttp://dx.doi.org/10.1007/s10916-016-0574-6

[10] M. E. El-Hawary, "The smart grid - State-of-the-art and future trends," *Electric Power Components and Systems*, vol. 42, no. 3-4, pp. 239–250, 2014.

[11] P. Zhuang, T. Zamir, and H. Liang, "Blockchain for Cyber Security in Smart Grid: A Comprehensive Survey," *IEEE Transactions on Industrial Informatics*, vol. 3203, no. c, pp. 1–1, 2020.

[12] Z. Huang, K. Suankaewmanee, J. Kang, D. Niyato, and N. P. Sin, "Development of reliable wireless communication system for secure blockchain-based energy trading," *JCSSE 2019 - 16th International Joint Conference on Computer Science and Software Engineering: Knowledge Evolution Towards Singularity of Man-Machine Intelligence*, pp. 126–130, 2019.

[13] J. Gao, K. O. Asamoah, E. B. Sifah, A. Smahi, Q. Xia, H. Xia, X. Zhang, and G. Dong, "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access*, vol. 6, no. c, pp. 9917–9925, 2018.

[14] Vitalik Buterin, "Ethereum Whitepaper." [Online]. Available: https://ethereum.org/en/whitepaper/

[15] H. Zhao, X. Bai, S. Zheng, and L. Wang, "RZcoin: Ethereum-based decentralized payment with optional privacy service," *Entropy*, vol. 22, no. 7, pp. 1–28, 2020.

[16] "Web3.js - Ethereum JavaScript API," 2018. [Online]. Available: https://web3js.readthedocs.io

[17] "Ethereumjs-tx." [Online]. Available: https://github.com/ethereumjs/ethereumjs-tx

[18] "Rinkeby Authenticated Faucet." [Online]. Available: https://www.rinkeby.io/{#}faucet

[19] "TESTNET Rinkeby (ETH)." [Online]. Available: https://rinkeby.etherscan.io/

[20] "TESTNET Kovan (KETH)." [Online]. Available: https://kovan.etherscan.io/

[21] "TESTNET Goerli (GTH)." [Online]. Available: https://goerli.etherscan.io/

[22] Infura, "Ethereum API — IPFS API Gateway — ETH Nodes as a Service." [Online]. Available: https://infura.io/

[23] Ethereum, "Official Go implementation of the Ethereum protocol." [Online]. Available: https://github.com/ethereum/go-ethereum

[24] "JSON-Remote Procedure Call (RPC)." [Online]. Available: https://eth.wiki/json-rpc/API

[25] Ethereum, "Go Ethereum: Command-line Options." [Online]. Available: https://geth.ethereum.org/docs/interface/command-line-options

[26] "React – A JavaScript library for building user interfaces." [Online]. Available: https://reactjs.org/

[27] "Empowering App Development for Developers — Docker." [Online]. Available: https://www.docker.com/

[28] "Deep Chain Reorganization Detected on Ethereum Classic (ETC)." [Online]. Available: https://blog.coinbase.com/

[29] Ethereum/EIPs, "ERC-20 Token." [Online]. Available: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md

[30] Blockchain, "Hashrate Distribution and Estimation of Hashrate Distribution Amongst the Largest Mining Pools." [Online]. Available: https://www.blockchain.com/pools

[31] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, "Proof-of-Stake Consensus Mechanisms for Future Blockchain Networks: Fundamentals, Applications and Opportunities," *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.