VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

Improvement of the CurveCP Cryptography for Enhancing the Secure of Internet of Things

Nguyen Van Tanh[1], Ngo Quang Tri[2,*], Nguyen Linh Giang[2],
Nguyen Anh Tuan[1], Nguyen Van Ngo[3]

[1]VNU International School, 144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
[2]Hanoi University of Science and Technology, Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam
[3]Cortek Co., Ltd, 378 Lac Long Quan, Xuan La, Tay Ho, Hanoi, Vietnam

Received 14 December 2020
Revised 08 March 2021; Accepted 12 May 2021

Abstract: With the increase of the threats at information security in Internet of Things (IoT), there are more and more security solutions have been designing. Unlike the traditional security, these solutions need to adapt with IoT Platform because of the difference about complex communication protocol, low energy, processing ability and limited memory. Our research team, after had been under a long process of analyzing theoretical documents and operating simulated experiments, improved, and implemented CurveCP which is one of these Lightweight cryptographies in the Wireless sensor Networks (WSN) to enhance data secure and information security of IoT System. This study briefly describes the improvement of CurveCP Lightweight cryptography by reducing length of cryptographic key as well as implement in IoT System. It also includes the simulated experiments, solutions evaluation, conclusion, and future development.

Keywords: Internet of Things (IoTs), Wireless Sensor Network (WSN), CurveCP, Cryptographic Box.

1. Introduction

Internet of Things (IoT) is expanding its popularity in human life. To ensure the data safety of IoT, the scientists designed the security solutions to each component of its and one of them is CurveCP Lightweight cryptography. CurveCP Lightweight cryptography was introduced in 2011 [1] and then was improved and completed. However, this protocol has been never installed completely in reality network model. The reason is all addition mechanisms always consume resource of IoT network. Especially, the resource consumption of the cryptographies likes CurveCP is extremely tremendous, and its operation might affect negatively to IoT System operation and, in some case, make it exhaust.

[*] Corresponding author.
E-mail address: tri.ngoquang@sie.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.282

Our research team, from the above overview, improved the CurveCP Lightweight Cryptography and implemented its source codes in IoT simulated experiment in Contiki Operation System. We also compare performance of installed-CurveCP WSN with normal WSN and thus, evaluated the possibility of the CurveCP in the IoT.

The study has 2 Chapters: Chapter 1: CurveCP Lightweight Cryptography which introduces definition, operating mechanism, and critical feature of the CurveCP as well as describes our improvement about it; Chapter II: Experiment which indicates process from design to implement of improved CurveCP in experiments simulating Wireless Sensor Network (WSN) Operation. This chapter also analyzes the results of these experiments; Chapter III: Evaluation, Conclusion and Future Development.

2. CurveCP Lightweight Cryptography

In this chapter, we describe all problems of information security in IoT currently and thus, demonstrates the importance of the CurveCP. After that, we introduce some critical features of the CurveCP and finally, describe some suitable improvements to fit it in IoT platform.

2.1. Information Security and Data Secure in IoT Network

The IoT Network is extremely vulnerable against threat about Information Security and Data Secure. In FAIR 2020 [2], we introduce the reason of IoT weakness including: poor protection in IoTs standards such as 6LoWPAN and Zigbee [3] as well as limited resource in Wireless Sensor Network (WSN) - a component of IoT System. Therefore, the security solutions in IoT platform have been designing to provide the strong protection of IoT Network and CurveCP Lightweight Cryptography is one of these solutions.

2.2. Introduce to CurveCP

CurveCP is the security solution using Elliptic Curve Cryptography [4] to ensure the flexibility as well as secure level in IoT Platform. From the first publishing in 27th Chaos Communication Congress on December 28th in 2010, the CurveCP has only developed in experiments for different scientific projects but without any commercialization. Figure 1 describes the location of the CurveCP.

In Figure 1, CurveCP has 2 versions: one is called Curve Server for Sink node and the other is called CurveCP Client for Sensor Node. The CurveCP monitor transmit lines between IoT nodes, and all messages is encrypted before transmission and decrypted before encapsulated in nodes.

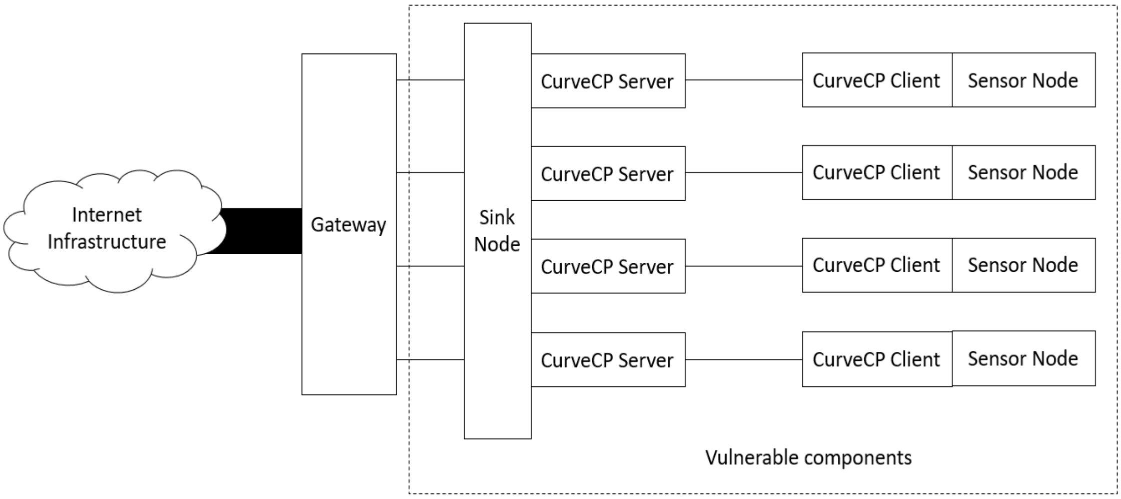

Figure 1. Location of the CurveCP.

2.3. Cryptographic Mechanism in CurveCP

The CurveCP uses the stream cipher mechanism in Application Layer in IoT. The CurveCP has 4 critical features is in the below list [5]:

Firstly, the CurveCP uses Client - Server Architecture and concentrate to protect Server.

Secondly, the CurveCP uses asymmetric cryptographic mechanism including Public key and Secret key. However, the CurveCP uses Cryptographic Box which encrypts and decrypts by separate keys. Thus, the CurveCP allows to combine between both authentication and encryption in the same function and reduce operation costs.

Thirdly, the CurveCP uses classified key distribution including Short-term key and Long-term key. Long-term key has long lifespan but low using frequency. In contrast, the Short-term key is spawned from Long-term key, is used with high frequency so has higher risk at being exposed and at the result, its lifespan is shorter. The distribution operation is decided by Nonce. In almost data transmission, all packets must be encrypted by Short-term Public key of Server and Client, expect the Nonce and Identification can be transmitted in plain text. In addition, Cryptographic Box of the CurveCP can encrypted by Long-term key but decrypted by Short-term key or oppositely, encrypted by Short-term key but decrypted by Long-term key.

Finally, the CurveCP uses Cookies which can be accessed by CurveCP Server. The CurveCP Server also owns a symmetric key called "*Super-secret key*" to encrypt and decrypt this Cookies.

In cryptographic mechanism of the CurveCP, the key distribution protocol is the essential key of CurveCP strength. This protocol has 2 stages: Initiation and Transmission. Figure 2 describes generally this key distribution protocol of the CurveCP:

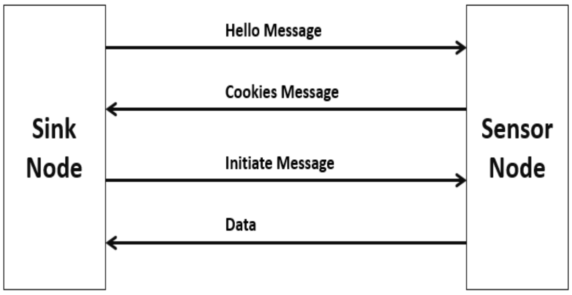

Figure 2. Key distribution protocol of the CurveCP Lightweight Cryptography.

About contents of each message in both 2 stages of the CurveCP, there is some notations are listed below. Client owns Long-term Public key C(P), Long-term Secret key C(S), Short-term Public key C'(P) and Short-term Secret key C'(S) while the Server owns Long-term Public key S(P), Long-term Secret key S(S), Short-term Public key S'(P) and Short-term Secret key S'(S); Nonce is N(x) while x is authenticating key. B(x) is cipher text which is encrypted from plain text x by Cryptographic Box B. s(x) is cipher text which is encrypted from plain text x by Super-secret key of Server. If the size of x is too high, x becomes,... And B(x) or s(x) becomes B(…) or s(…). 0 means all bit in messages is zero bit while Data means message data. Table 1 describes contents of messages in Stage 1: Initiation:

In Stage 2: Transmission, Client sends to Server: CMessage(Data) and N(C'(P)) while Server sends to Client: SMessage(Data), N(S'(P)).

Table 2 describes operations of Cryptographic Boxes in the CurveCP:

Table 1. Contents of messages in Stage 1: Initiation in the CurveCP Lightweight Cryptography

Message	Sender	Receiver	Content
Hello	Client	Server	Hello (0), C'(P), N(C'(P))
Cookies	Server	Client	Cookies(s(C'(P), N(C'(P)), S'(S), N(S'(S))), S'(P)), N(S'(P))
Initiate	Client	Server	s(C'(P), N(C'(P)), S'(S), N(S'(S)), Initiate (Vouch(C'(P), S'(P)), C(P)), N(C'(P))

Table 2. Operations of Cryptographic Boxes in the CurveCP

Box	Side	Crypt	Decrypt	Content Plain Text
Hello	Client	C’(S)	S(P)	0
Cookies	Server	S’(S)	C’(P)	s(…), S’(S), N(S’(P)
Vouch	Client	C(S)	S’(P)	C’(P), S’(P)
Initiate	Client	C’(P)	S’(S)	V(…), C(P)
CMessage	Client	C’(P)	S’(S)	Data
SMessage	Server	S’(P)	C’(S)	Data

The CurveCP Lightweight Cryptography follows a critical principle: “A node must not own both the decryption key and the encryption key”. It means the Box only encrypts by key from sender and decrypts by key from receiver. The Cryptographic Box decrypts using Short-term Secret key of Sender, so it ensures the secure of data. Meanwhile, only the receiver owns Short-term Public key of sender so only the receiver can decrypt the message from the sender. From this above feature, it is well-founded to claim: the principle of the CurveCP can support nodes to both secure and authenticate data. As the result, the volume of calculation is declined but the CurveCP still ensure high performance to prevent against threat from sniffing attacks and spoofing attacks. With this prevention, the Integrity and the Confidentiality of IoT System is highly secured.

2.4. Improvement of CurveCP Lightweight Cryptography

As the above mention, the CurveCP brings the disadvantages of cryptographic mechanism likes large resource consumption and need to be improved to decrease resource consumption and eliminate its negative effect in operation of IoT Network. All improvements focus on reducing the length of key because this change creates less uncontrolled side-effects but helps to reduce resource consumption in case of high cryptographic frequency. We improved all 3 critical messages including Hello message, Cookie’s message, and Initiate message by reducing 8 keys (notations of these keys shown in Table 1) including C(P), C(S), C’(P), C’(S) in Client and S(P), S(S), S’(P) S’(S) in Sever. The length decreases from 32 bit to 8 bits. When the key length decreases, the security level of the CurveCP absolutely decreases. However, the protection of this solution lost its value if its operation affects negatively the IoT activities, so this improvement is necessary. In addition, the rate of decrease ensures the balance between the resource consumption and the security level of the CurveCP.

Despite reducing resource consumption, these improvements cause a side-effect to be reducing the security level because the lower the key length is, the higher risk at brute force attacks will be. Nevertheless, the mission of the CurveCP providing the stability of IoT operation so this mission is clearly failing if the CurveCP consume much resource and make IoT operation unstable. Therefore, the decline of energy consumption is more urgent, and we must accept to reducing the secure level to keep the stability in operation of IoT Network.

3. Simulated Experiment of the CurveCP Lightweight Cryptography

As the below mention, the target of this study is proofing the possibility and the efficiency in implementation of the improved CurveCP in the simulated IoT Network in Contiki-OS. This chapter describes process for implementing the improved CurveCP, experiments script, measuring criterions and results:

3.1. Introduction of Contiki Operating System

Contiki Operating System (Contiki OS) is open-source software designed to simulate models of WSN. The Contiki OS is introduced by Adam Dunkels in 2002 [6] to meet the high demand for developing IoT Technology. The advantages of the Contiki OS conclude open-source code providing flexibility, the close visualization of simulation and friendly interface. The simulated experiments can be designed easily and run via Cooja simulation tools in Java platform [7].

3.2. Implement the Improved CurveCP Lightweight Cryptography

Source code library "*curvecp*" about the CurveCP in the Contiki OS is developed by Jan Mojzís in a project for developing TinySSH Security Protocol for IoT Network [14] and inherited source code about security protocol in Networking and Cryptography library (NaCl) of Daniel J. Bernstein [15]. Because the "*curvecp*" uses MUSL library developed by Rich Felker in 2011 [16], the first stage of the implementation of the CurveCP is installing it as well as setting its reference in the Contiki OS including downloading "*musl*" source code and configuring it in Contiki OS by the below code:

```
$ ./configure && make install
```

After that, the second stage of the implementation is downloading simulated library named "*contiki-master*" including scripts simulating WSN nodes and IoT operation. In "*contiki-master*", the folder "*apps*" is assigned to store additional mechanism so the additional cryptographic mechanism "*curvecp*" locate in this folder. In addition, the target is comparing improved CurveCP and normal CurveCP so the library "*curvecp*" must be duplicated and one of them will be rename to "*improved-curvecp*" and reducing cryptographic key as the mention in Part 2.4. The work including change configuration argument in file "*crypypto-block.h*" in folder "*src*":

```
// #define crypto_block_KEYBYTES 32
#define crypto_block_KEYBYTES 8
```

It is noted that the code line with mark "//" before is the code in normal "*curvecp*" and was deactivated.

The last stage in the CurveCP implementation is setting a reference by adding below code line in file "*project-conf.h*" in folder "*rpl-udp*" with link "*contiki-master/examples/ipv6/*".

In case of simulated experiments for the normal CurveCP:

```
APPS += curvecp
```

In case of simulated experiments for the improved CurveCP:

```
APPS += improved-curvecp
```

In addition, the CurveCP has 2 separate version: CurveCP Server for the Sink node and CurveCP Client for the Sensor Node so we must use a function declaration from "*curvecp*" or "*improved-curvecp*" to activate CurveCP

In file "*sink-node.c*" the function declaration for activate CurveCP Server:

```
Initiate-server-curvecp()
```

In file "*sensor-node.c*" the function declaration for activate CurveCP Client:

```
Initiate-client-curvecp()
```

After these 3 stages, the implementation of CurveCP and improved CurveCP was completed. The next part describes the design of simulated WSN topology and experiments testcase.

3.3. Design of Simulated WSN Topology and Experiments Testcase

Topology of simulated WSN is 5x5 grid which is widely used in IoT System because the grid topology support multi routing perfectly. Figure 3 describes this topology:

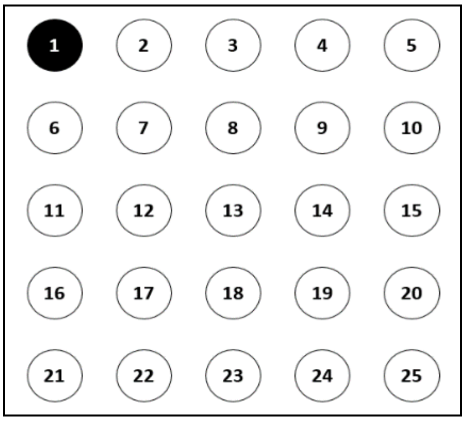

Figure 3. Topology of simulated WSN.

In Figure 3, a Sink node has black background color and white number color while Sensor nodes have white background color and black number color. In experiments, the Sensor nodes send data message to the Sink node with fixed frequency. Total time of each experiments in each testcase is 5 minutes.

As the above mention, the target of experiments is proofing the possibility for implementing improved CurveCP Lightweight Cryptography in simulated IoT Network by proofing the operation of IoT running improved CurveCP is stable. In addition, to proofing the advantages of our improvement, we run couple of experiments in the same condition: one installed normal CurveCP and the other installed improved CurveCP, thus we compare performance via measure some criterion and compare and evaluate our improvement.

3.4. Measuring Criterions

Three measuring criterions concludes Packet Delivery Ratio (PDR), Latency and Energy Consumption:

PDR is rate between the number of received packets and the number of sent packets. The unit of PDR is percent (%). Formula (1) calculates PDR:

$$PDR = \frac{R}{S} x100 \quad (1)$$

In Formula (1), S is the number of packets the calculating node sent while R is the number of packets the other nodes received from calculating node. PDR represents the reliability of transmission, the higher PDR is, the higher successful rate of transmission is [9]. According to Mansfield from Cengage Center, PDR from higher 95% to ensure the WSN has stable operation [10].

Latency is the average time a packet between departing from sender (calculating node) and arriving to receiver. The basic unit of Latency is milliseconds (ms). Formula (2) calculates Latency:

$$\text{Latency} = \frac{\sum_{i}^{n} (t(R)_i - t(S)_i)}{n} \quad (2)$$

In Formula (2), n is number of successful transmission packets, i is the index of packet, T(S)i is the time the calculating node sent packet index i while T(R)i is the time the receiver received packet index i. The Latency represents the quality of transmission, the higher the Latency is, the longer time for transmitting is [9]. According to SAS Information Technology Service Center in the United Kingdom, the Latency must be lower than 800ms to ensure the WSN has stable operation [11].

Energy Consumption is the abstract criterion represent to which amount of energy is consumed in different simulation activities. In Contiki, the energy consumption is calculated by the rate between the time node for different tasks (sending packets, receiving packets) and total time of simulation. However, Sourceforge proposed the Formula (3) to calculate energy consumption measured by milli Joule (mJ) from the abstract value [8].

$$E = (Txx19.5 + Rxx21.8 + CPU \text{ x } 1.8 + LPM \text{ x } 0{,}545) \text{ x } \frac{3}{32768} \quad (3)$$

In Formula (3), Tx is the rate between time a node uses to send packets and total simulation time while Rx is the rate between time a node uses to receive packets and total simulation time. CPU is energy consumption of CPU for simulation (different kind of node has different CPU value) and LPM is the rate between the time a node uses for basic tasks of node and total simulation time. The kind of node in simulation is Tmote Sky which require energy consumption lower than 12.6 mW [12] in a hour, 1.05 mW in 5-minutes simulation. It means the energy consumption must be lower than 315 mJ [13] to to ensure the WSN has stable operation.

3.5. Results and Evaluation

In total WSN, we will measure three criterions and take the average value of all nodes in WSN. Result is indicated in Table 3.

Table 3. Results of experiments

Testcase	PDR (%)	Latency (ms)	Energy Consumption (mJ)
No installed CurveCP	98.67	543.98	119.21
Installed normal CurveCP	92.13	893.24	287.90
Installed improved CurveCP	96.04	662.71	190.08

This is evaluation fron results of experiments indicated in Table 3.

Firstly, WSN using the normal CurveCP (not any improvement) consume a plenty of resource and its criterions do not reach their thresholds of stable operation. PDR decreased rapidly under 95% [10] while Latency increased rapidly to above 800 ms [11]. About energy consumption, despite reaching their thresholds of stable operation, it gets as twice as this criterion of WSN not using CurveCP.

Secondly, WSN using improved CurveCP consume litter resource than it is using normal CurveCP so its criterions reach their thresholds of stable operation. The energy consumption is higher than WSN not using CurveCP but litter than WSN using CurveCP without improvement.

To conclude, the CurveCP Lightweight Cryptographic, after the process of improvement, can be implemented in WSN and its operation do not affect negatively to WSN operation.

4. Conclusion

In this study, we described some basic problems of IoT platform likes the lack of security mechanisms and limited resource and it makes IoT System becomes extremely vulnerable. From this premise, we improved and implemented the Curve Lightweight Cryptographic to protect security and data secure IoT System as well as run experiment about simulated WSN with 25-nodes 5x5-grid topology. The CurveCP contains Cryptographic Box which is special algorithm combining asymmetric and symmetric cryptography so it can decrease volume of calculation and thus, reduce energy consumption. However, from results in experiment with WSN using the CurveCP, we recognize the resource saving from using Cryptographic Box is not enough to prevent the WSN operation being unstable by out of resource. When comparing criterions including PDR, latency, and energy consumption, the WSN using CurveCP consume as much as energy so PDR and Latency do not reach stable threshold. Therefore, we must reduce resource consumption by reducing key length of Cryptographic Box from 32 bit to 8 bit and accept its side-effect likes to decline of secure level. Experiments with WSN using the improved CurveCP proofing the rationality of out improvement when PDR and latency reached the stable threshold.

In the future, our team will combine CurveCP with our proposed mechanism like Overhearing preventing Denial of Service Attack by Botnet [17, 18] to protect the availability of IoT System because the CurveCP protect only the confidentiality and integrity. This combination creates a comprehensive security solution with 3 security characteristics of CIA Triangle is protected [19].

Acknowledgements

Our team sends the gratefulness to tutors and the colleagues for providing adequate support about infrastructure and advice during the researching process. They work in Bach Khoa Cipher Security Center and other faculties in School of Information Communication Technology, Hanoi University of Science and Technology as well as Vietnam National University. Support by Project QG.19.58 of VNU.

References

[1] D. J. Bernstein, CurveCP: Usable Security for the Internet, University of Illinois at Chicago, January 22, 2017.

[2] N. V. Tanh, N. Q. Tri, N. L. Giang, N. A. Tuan, Design of Comprehensive Security Solution on Internet of Things with Improved DTLS Protocol and Overhearing Mechanism, Fundamental and Applied Information Technology Research (FAIR), Nha Trang, Vietnam, 2020.

[3] D. X. Li, W. He, S. Li, Internet of Things in Industries: A Survey, IEEE Transactions on Industrial Informatics, Vol. 10, No. 4, 2014, pp. 2233-2243.

[4] D. Hankerson, A. J. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer Publishing, 2003.

[5] T. Pauly, C. Perkins, K. Rose, C. Wood, A Survey of Transport Security Protocols, University of Glasgow, September 6, 2018.

[6] A. Dunkels, Contiki: Bringing IP to Sensor Networks, ERCIM News Journal, European Union, 2009.

[7] Pedro, Mugdhe, Samarth, Cooja Simulator - Contiki Tutorials, Autonomous Networks Research Group, University of South California, 2016.

[8] M. Abdellatif, Contiki Developer, Power Consumption, July 17, 2017.

[9] D. Culler, D. Estrin, M. Srivastava, Guest Editors' Introduction: Overview of Sensor Networks, 2004, pp. 41-49, https://doi.org/10.1109/mc.2004.93.

[10] K. C. Mansfield, J. L. Antonakos, Computer Networking from LANs to WANs: Hardware, Software, and Security, Boston Cengage Learning, 2010, pp. 501.

[11] SAS Team, What is Network Latency (and How Do You Use a Latency Calculator to Calculate Throughput)?, The SAS Group of Companies Limited, April, 2019.

[12] M. Johnson, M. Healy, P. V. D. Ven, J. H. Martin, J. Nelson, T. Newe, E. Lewis, A Comparative Review of Wireless Sensor Network Mote Technologies, IEEE SENSORS Conference, 2009, pp. 1439-1442.

[13] D. Hofstrand, Energy Measurements and Conversions, Iowa State University Extension and Outreach, 2007.

[14] J. Mojzís, tinyssh, The SSH Library, 2018.

[15] D. J. Bernstein, T. Lange, P. Schwabe, NaCl: Networking and Cryptography library, University of Illinois at Chicago, 2016.

[16] R. Felker, Musl 1.1.24 Reference Manual, License of Massachusetts Institute of Technology, 2011.

[17] N. V. Tanh, N. Q. Tri, N. G. Tuyen, T. Q. Duc, T. H. Anh, B. T. Tung, The Flooding Attack in Low Power and Lossy Networks: A Case Study, The 7 th IEEE International Conference on Smart Communications in Network Technologies (SaCoNet 2018), El Oued, Algeria, 2018.

[18] N. V. Tanh, N. Q. Tri, N. G. Tuyen, N. L. Giang, N. V. Tien, Design a Security System for Internet of Things with Detectinng and Author Proceedings of Eliminating Denial of Service Attack Based on Overhearing Mechanism, the 3rd Symposium of Information Security (SoIS 2018), Da Nang, Vietnam, 2018.

[19] S. Samonas, D. Coss, The CIA Strikes Back: Redefining Confidentiality, Integrity and Availability in Security, Journal of Information System Security, Vol. 10, No. 3, 2014, pp. 21-45.