



Enhancing Data Plane Security Against Cyber Attacks for Wired and Wireless Communication Software Defined Networks Based on Encrypted Algorithms

Fadia Noori Hummadi¹ • Mahmood Zaki Abdullah*²
Mohammed Ali Tawfeeq², and Ali Khalid Jassim³

¹Al-Khwarizmi Collage of Engineering, University of Baghdad, Baghdad, Iraq.

²Computer Engineering Department, Mustansiriyah University, Baghdad, Iraq.

³Electrical Engineering Department, Mustansiriyah University, Baghdad, Iraq.

Received: 01 02 2025; Accepted: 05 06 2025

Available: 30 04 2026

Abstract: Information technology is constantly evolving in terms of structure, hardware, and software, and one of the most important areas of information technology that is continually updated is networks, which take up a large area in terms of development and innovation in applications, services, tools, etc. Now is the time for Software Defined Networks (SDN) in wired and wireless topologies to emerge. SDN is a modern networking technology in which the data plane is separated from the control plane, and control functions are gathered in one device called a controller, which leads to many security problems and exposure to attacks by intruders. The aim of this article is to enhance data plane security against cyber-attacks for wired and wireless communication software-defined networks based on encrypted algorithms. In this article, a Developed Encryption Algorithm (DEA) is proposed to protect wired and wireless communication SDN against malicious and cyber-attacks. The proposed algorithm was tested and verified by sending and receiving ten text files of different sizes through three scenarios of SDN topology, and then calculating the encryption and decryption time. Finally, the National Institute of Standards

*Corresponding author.

E-mail address: drmzaali@uomustansiriyah.edu.iq (M.Z. Abdullah).

Peer Review under the responsibility of Universidad Nacional Autónoma de México.

and Technology (NIST) test was applied to verify the complexity and strength of the proposed algorithm, and the DEA algorithm gives an accuracy of 95.39% as compared with the SG algorithm, which gives an accuracy of 94.93%.

Keywords: SDN, control plane, data plane, DEA, NIST

1. Introduction

Many devices in technology are connected via networks. Networks consist of several devices, and the most important of these are the switch and Router, which handle traffic. In traditional networks, these devices consist of two important parts: the data plane and the control plane. The software-defined network worked on its hardware to separate the data plane from the control plane, with these devices responsible only for data traffic and the control functions combined into a single device called a controller. In addition to the two mentioned (plane) layers, it also contains a layer responsible for applications called (Application layer) and two interfaces (Southbound and Northbound), and their function is to link the SDN layers together and also has an OpenFlow protocol whose function is to measure the communication between the control plane and the nodes (Nunes et al., 2014). The main difference between the traditional network and Software-Defined Networks is that SDNs are characterized by a control plane and a data plane, whereas in traditional networks these two planes are combined into a single control and data plane. Figure 1 (Prajapati et al., 2018).

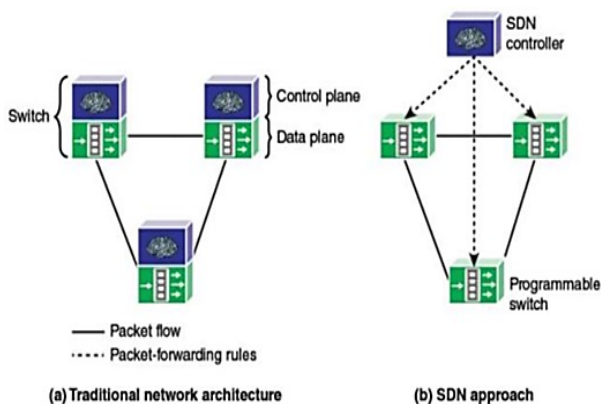


Figure 1. Traditional network vs. SDN network (Peterson et al., 2021).

Below, an architecture is briefly illustrated in Figure 2:

- **Application layer:** The top layer in SDN consists of applications and services provided by the network to the user. This layer communicates with the next layer (the control layer) via the Northbound interface (Nayyar et al., 2022).
- **Northbound Interfaces:** A software API that allows communication between high-level components and helps applications to access their functions and services at a control level without having information about the details of primary network switches (Kaur et al., 2024).
- **Control layer:** The control layer can be considered the layer responsible for all decisions to control the sending and receiving of information packets, in addition to its responsibilities in managing and controlling all types of traffic related to the passage of these packets, and that all of these activities are implemented through a program and not through physical electronic units (Kaur et al., 2024).
- **Open Flow** is one of the software-defined networking (SDN) standards. This enables the controller to communicate with the data plane (switches/routers), whether physical or virtual (Manasyan, 2022).
- **Southbound Interfaces:** It represents the link between the control plane and the routers and is used to define the set of instructions with the flow table (Nayyar et al., 2022).
- **Data Layer:** The third layer in the SDN structure, also known as the infrastructure layer, that defines the internetworking devices (routers and gateways) with all instructions that can be used via API (Kaur et al., 2024).

2. Literature Survey

Many researchers have made great efforts in this research field, the most prominent of whom are the following:

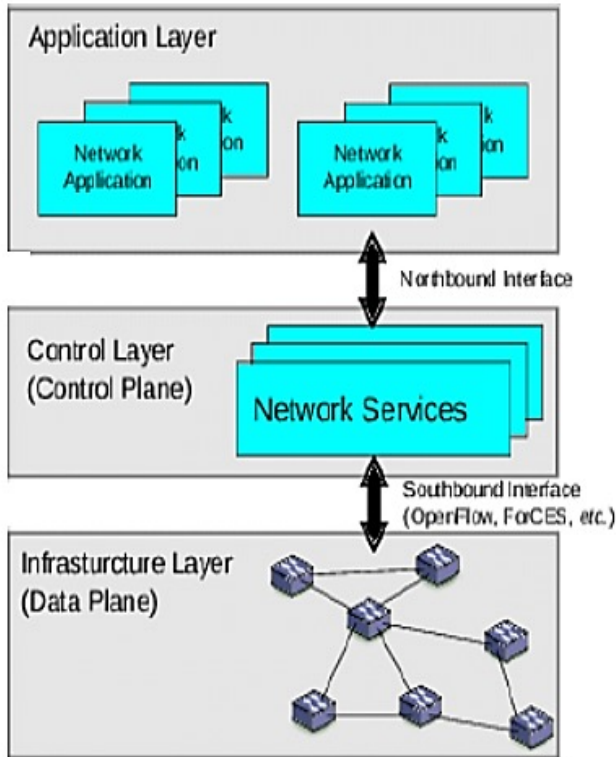


Figure 2. WCSDN Architecture (Braun & Menth, 2014).

Sandra et al. (2013), described the security challenges of the SDN network through two schools: the first because of the programmability and central control of this network can expose the network to many attacks, and the second deals with the two features mentioned above that can help the network to improve the safety of the network dramatically.

Asad et al. (2015) proposed creating a network security architecture defined by software, using a special type of controller (FlowVisor) and the AES encryption algorithm, and measuring the encryption and decryption times.)

Oluwasogo et al. (2015) developed an algorithm to prevent malicious attacks against the SDN, and this algorithm is a combination of source definition, authentication (using blacklisting and whitelisting) and content filtering (using word fragmentation and Bayes theory) and that malicious identification/authentication and packet aggregation, provides a solution It effectively identifies/authenticates legitimate/malicious mail, thus preventing the malicious attack from reaching their target host in the SDN. The result was 10% positive and false positive, and 90% positive and real positive.

Chao et al. (2016) studied three complementary methods of data plane security. These methods are promising solutions for localizing and mitigating malicious SDN keys: Active probing, Statistics checking, and Packet obfuscation. Firstly, is to detect errors that ensure the correct implementation of the flow rule by sending test packets, while secondly is a way to make the fast detection of malicious keys by checking the flow statistics that are consistent, and the third is a way of making of the minimum damage through the encryption process to the package's content.

Ertaul & Venkatachalam (2017) concluded that it is possible to provide suitable security using IPsec VPN in addition to using AES and TLS encryption. An SDN topology was created, and the network was attacked by DDoS and MITM, then the AES and TLS algorithms were used to secure communication between the switch and controller, and IPsec VPN was used to protect the connection between hosts, thus protection was provided against IP spoofing and data modifications.

Ragaharini et al. (2018) analyzed the real packet messages between the data plane and the control level by studying the performance of three types of controllers (ONOS, Open MUL and POX), and they were implemented through programming languages (Java, C, Python respectively) using packet analysis Mininet-Wireshark and comparing the performance of the three controllers with the performance of the measuring instrument ("cbench / which uses counterfeit control packets generated by switch instances") in terms of latency and throughput, the results showed that this reduces 96% of the controller arrival time and 98% of His speed.

Deepak et al. (2019) provided and covered SDN network model, challenges and solutions in terms of reliability that have a fundamental role in the development of software and inform the user in an emergency and work on the automatic solution and to increase the reliability of the SDN must be smart in the management of the network To prevent errors, and to extend the performance of the central console must support at least 100 switches, the SDN framework must coordinate multiple asynchronous events in the switches to perform specific duties, so open interfaces in this network lead to new types of attacks. Caused their performance to fail (for example, a DDoS attack (where the network has stopped working), so encryption methods must be developed to stop attacks or reduce their strength.

3. Methodology

The methodology for our proposed work will be implemented in an SDN environment through three scenarios, each with a distinct structure. In all three scenarios, the controller type is POX, with a specified number of switches and hosts, and the switches and hosts are connected to each other by multiple links to form (Single, Linear, and Tree) topologies. In each scenario, the process of sending and receiving messages or data is protected by a new encryption algorithm developed in this article, and this algorithm will be examined by NIST and is expected to pass the test. The proposed research work includes several steps that can be summarized as follows:

3.1 Research Environment

All the work of this paper has been done in a physical (HP) computer (Laptop) with 8 GB of Random-Access Memory (RAM), Processor Intel® Core™ i7-8550U CPU @ 1.8 GHz 2.00 GHz, and the installed Operating System is Windows 10 Pro 64-bit operating system. The used virtual machine was Oracle VMWare VirtualBox with Ubuntu version 15 (64-bit Operating System) with allocated base memory equal to 5000 MB, and a package of Mininet version 2.2.1 was installed over the Ubuntu operating system. The programming language used to execute all the results would be Python 3.7.

3.2 Scenarios for WCSDN Topology

In this research work, three scenarios would be achieved as listed below.

3.2.1 Single Topology

This scenario consists of one control unit (controller) connected to one switch, which in turn is connected to four hosts, as shown in Figure 3.

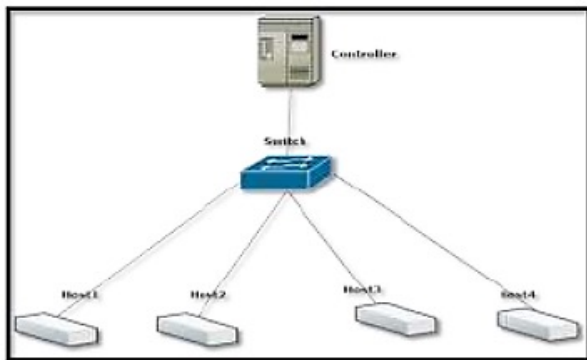


Figure 3. WCSDN single topology.

3.2.2 Linear Topology

This type of WCSDN scenario consists of one controller connected to four switches and four hosts, with each switch connected to one host in a linear fashion, as shown in Figure 4.

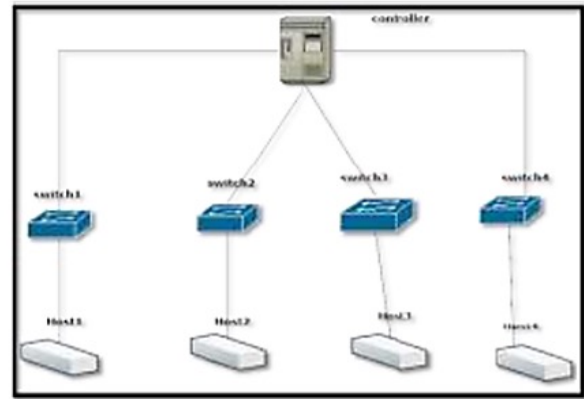


Figure 4. WCSDN linear topology.

3.2.3 Tree Topology

This type of WCSDN scenario consists of one controller connected to one switch, which is in turn connected to two switches. Each switch has three hosts, and the final connection topology forms a tree-like structure, as shown in Figure 5.

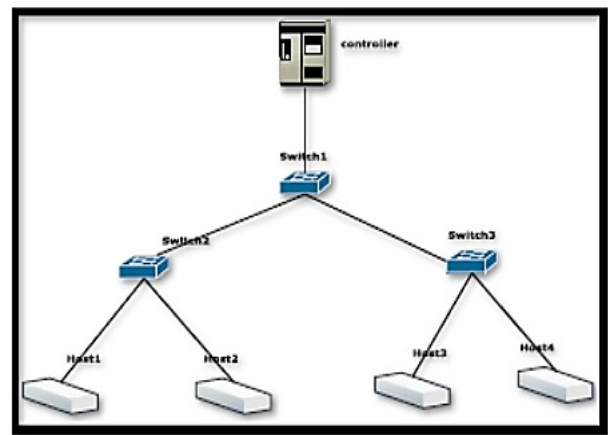


Figure 5. WCSDN tree topology.

3.3 Developed Encryption Algorithm System Design

It is a developed form of the (SG algorithm) derived from the replacement tables (S-BOX) for the two algorithms (Serpent and GOST), and it is applied to the WCSDN

network. This algorithm is designed to increase complexity, speed, and security. This newly developed algorithm is considered a block cipher that consists of three functions (new function M, new function F, and a nonlinear function based on the Serpent S-BOX) and many techniques (XOR operation, shifting the text block to the right by the number of ones, and shifting the text block to the left by the number of zeros). This algorithm consists of 512-bit (64-byte) segments. The encryption key consists of 512 bits for ten rounds, plus a primary key generator that is sufficient for 10 rounds with a length of 5120 in each round. A new 512-bit length is truncated from the key sequentially.

3.3.1 Developed Encryption Algorithm Encryption System

Encryption is the process of converting plain text into ciphertext, making it unavailable to anyone who reads or snoops on texts or messages sent from the sender to the recipient. Figure 6 shows the flowchart of the DEA encryption algorithm, with its steps illustrated.

The input consists of 512-bit text with a 5120-bit key; the input text is processed for ten rounds, starting with the entered text using a new function M. The input bits are divided into two parts (left and right), each consisting of 256 bits. They are processed by shifting them in the same direction (left or right) by the number from chaos. The key block is then divided into two 256-bit blocks (left and right). The XOR operation is applied between the outputs of the operations performed on the text block, which is divided into two blocks (left and right) (i.e., apply the XOR operation between the left part and between the right part), with the two blocks switched together, and then the new left part is addressed by a function The new F and the new right pane by a nonlinear function based on the S-Box serpent. The output of these two processes is combined, and the Transposition encoding with data compression process is applied to the output.

3.3.2 Build a New Function (M)

A 32-bit text is entered, and this function takes all these bits and divides them into 4-bit chunks. Operations are applied to these pieces (XOR, add, apply S-BOX, apply shift on some chunks and substitute with each other). This function is executed 16 times, 32-bit input for each execution process and 32-bit output. Inputs are separated and executed several times instead of being implemented once in order to minimize the execution time. Used S-Box of GOST algorithm, the input to the S-Box is 4 bits and the output is 4 bits too. The same function is used in decryption, but backward.

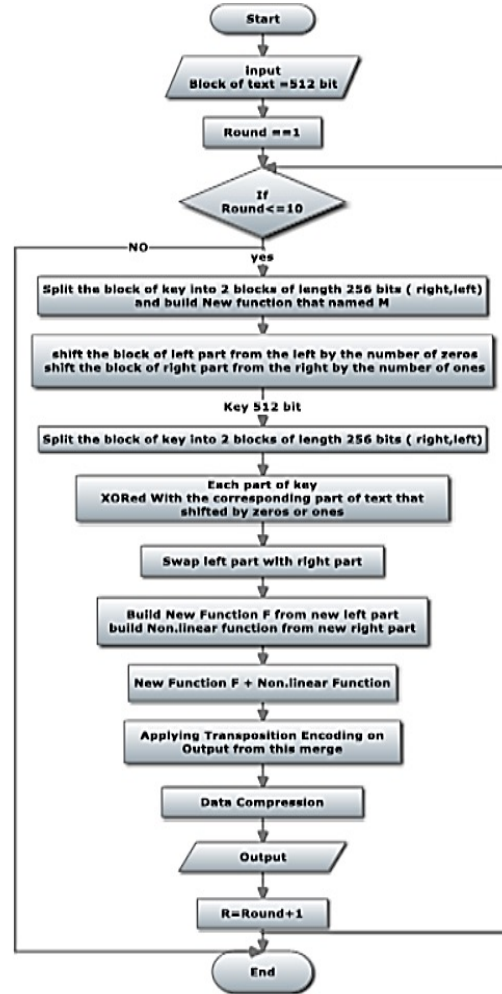


Figure 6. Flowchart of the encryption process in the DEA Algorithm.

3.3.3 Build a New Function (F)

The input consists of 64 bits. This function takes 64 bits of data. First, the data block is divided into 16-bit pieces; then the pieces are swapped, their DNA complements applied, and the results added and XORed with each other. This function is executed 4 times. Each execution takes 64 bits of input and produces 64 bits of output. The cross-over takes 16-bit input. First, the input is divided into 8-bit pieces, then each piece is split into two 4-bit parts. The first part of the first piece is swapped with the second part of the second piece, and the second part of the first piece is replaced with the first part of the second piece. The same function is used in decryption, but in reverse.

3.3.4 Non-Linear Function

32 bits can be entered to this function, begins with a data block is divided into 4-bit pieces, then these pieces are

applied with (XOR, S-BOX, and shift on some pieces). This function is executed eight times where each execution is 32-bit input and 32-bit output. S-Box of the Serpent algorithm, 4-bit input and 4-bit output. The function is used in reverse during the decryption process.

3.3.5 Developed Encryption Algorithm Decryption System

Decryption is the process of converting ciphertext to plaintext, transforming incomprehensible words and phrases into a conceptually meaningful form using a known algorithm between the parties (sender and recipient). The DEA decryption algorithm flowcharts are shown in Figure 7, and its steps are illustrated below.

256 bits, the left part would be processed by a new function F, while the right part would be processed via a non-linear function based on the S-Box that switched between the left and right panes. The 512-bit key block would be divided into two blocks (left and right) where each one of the blocks consists of 256 bits, the XOR operation would be applied to the left part of the key with the output from the right part of the text handled by the non-linear function, while the XOR operation would be applied to the right part of the key with the output from

the left part of the text handled by the new function F. The left part is then processed by shifting the block from the right, the right part is processed by shifting the block from the left and the output from these two processes can be combined and the output from the merger is processed by the new function M. Finally, after processing the transposition process will be applied and the output is the input to the next operation.

4. Implementation and Results

The proposed algorithms can be verified and tested through three scenarios of WCSDN topologies with their protection by the DEA algorithm, using ten text files in sizes (64B, 10KB, 20KB, 30KB, 40KB, 50KB, 60KB, 100KB, 160KB, 200 KB). Calculating the encryption and decryption time with the test DEA algorithm via the (NIST) test, and all the results are compared with the SG algorithm.

4.1 Single WCSDN Topology

The implementation results for encryption and decryption in this topology are shown in Figures 8 and 9, respectively.

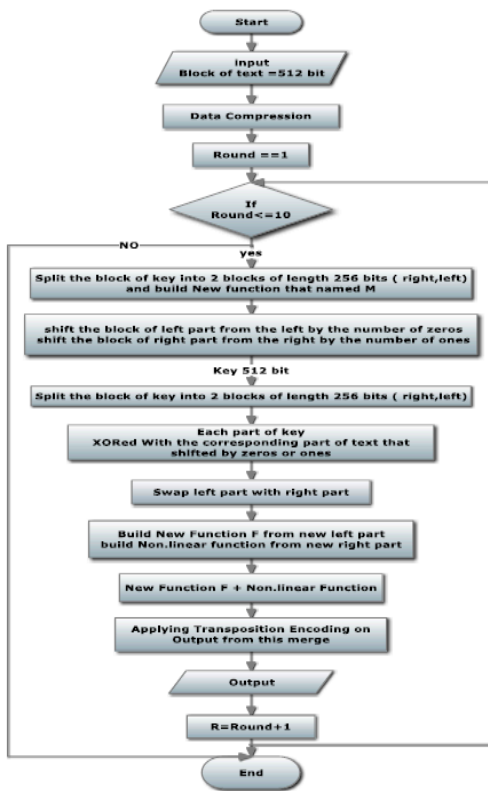


Figure 7. Flowchart of the decryption process in the DEA Algorithm.

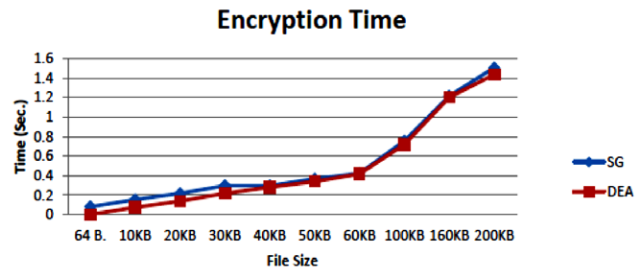


Figure 8. Encryption Running Time Between SG and DEA In Single Topology.

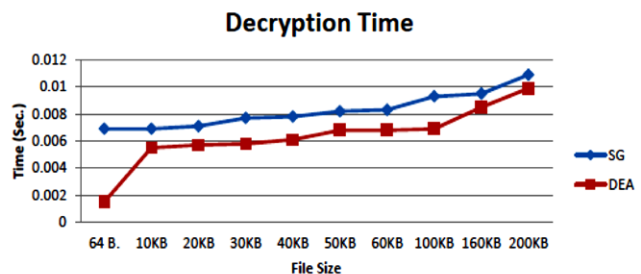


Figure 9. Decryption Running Time Between SG and DEA in a Single Topology.

4.2 Linear WCSDN Topology

The implementation results for encryption and decryption in this topology are shown in Figures 10 and 11, respectively.

4.3 Tree WCSDN Topology

The implementation results for encryption and decryption in this topology are shown in Figures 12 and 13, respectively.

It can be clearly seen that the results obtained from the decryption process of the ten files show that the DEA algorithm implemented the decoding process faster than the (SG) algorithm, as mentioned previously. When comparing, the results showed that all scenarios in the SDN network in which the developed encryption algorithm outperformed the new algorithm, but in a (dual-control) scenario, better results were achieved than in other scenarios by implementing the decoding process using the DEA algorithm on it.

4.4 National Institute of Standards and Technology (N.I.S.T.)

The N.I.S.T. is a statistical test that includes 16 tests and is frequently used; these tests can be applied to a variety of random processes, as shown in Tables 1 and 2.

Table 1. SG Algorithm N.I.S.T. results.

Testing Type	Value of P.	Outcome
1. Test of Frequency	0.2700	Rand.
2. Test of Frequency block	0.1890	Rand.
3. Test of Run	0.5830	Rand.
4. Long running of 1s in a block	0.1941	Rand.
5. Test of binary matrix ranking	0.4503	Rand.
6. Test of Discrete Fourier Transform (DFT)	0.6049	Rand.
7. Test of non-overlapping matching	0.9493	Rand.
8. Test of overlapping matching	0.6993	Rand.
9. Test of Maurer's statistical universal	-1.0000	Non-Rand.
10. Test of linear complexity	0.1939	Rand.
11. Test of serial process	0.3005	Rand.
12. Test of Approx. entropy	0.0001	Non-Rand.
13. Test of cumulative forward sums	0.0001	Non-Rand.
14. Test of cumulative reverse sums	0.0001	Non-Rand.
15. Test of random excursions	0.5245	Rand.
16. Test of rand. variant Excursions	0.1920	Rand.

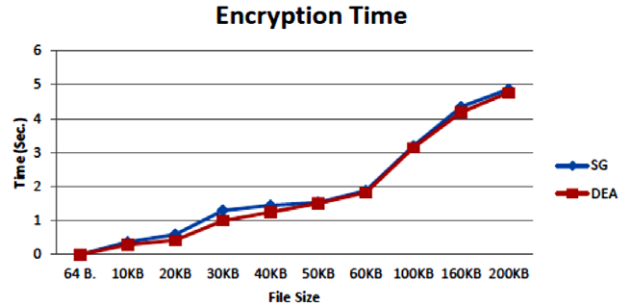


Figure 10. Encryption Running Time Between SG and DEA In Linear Topology.

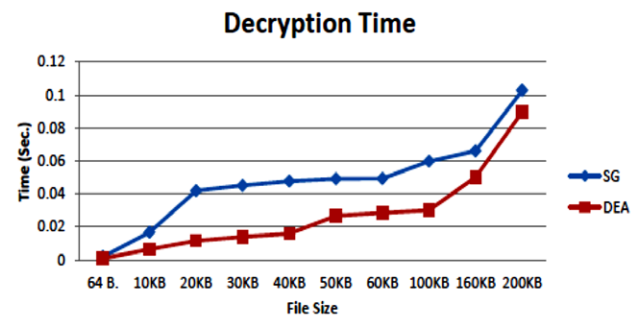


Figure 11. Decryption Running Time Between SG and DEA In Linear Topology.

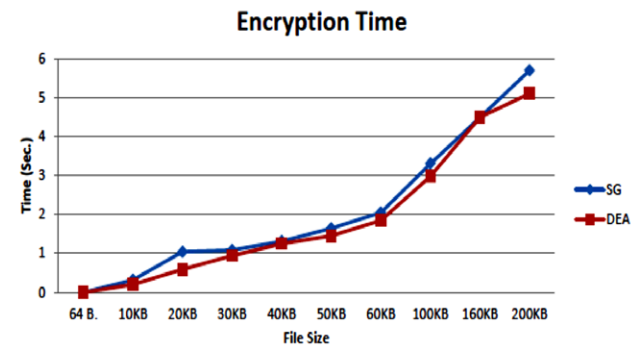


Figure 12. Encryption Running Time Between SG and DEA In Tree Topology.

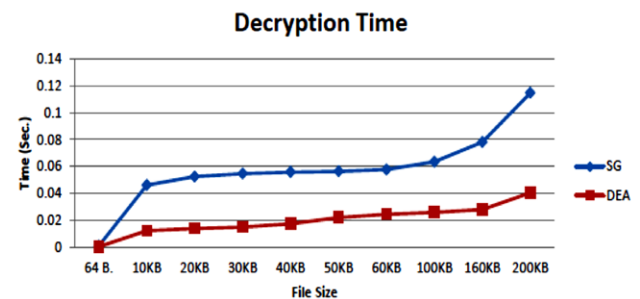


Figure 13. Decryption Running Time Between SG and DEA In Tree Topology.

Table 2. DEA Algorithm N.I.S.T. results.

Testing Type	Value of P.	Outcome
1. Test of Frequency	0.2700	Rand.
2. Test of Frequency block	0.1890	Rand.
3. Test of Run	0.7202	Rand.
4. Long running of 1s in a block	0.9539	Rand.
5. Test of binary matrix ranking	0.4503	Rand.
6. Test of Discrete Fourier Transform (DFT)	0.3334	Rand.
7. Test of non-overlapping matching	0.6641	Rand.
8. Test of overlapping matching	0.6993	Rand.
9. Test of Maurer’s statistical universal	-1.0000	Non-Rand.
10. Test of linear complexity	0.1436	Rand.
11. Test of serial process	0.1919	Rand.
12. Test of Approx. entropy	-1.0000	Non-Rand.
13. Test of cumulative forward sums	0.2450	Rand.
14. Test of cumulative reverse sums	0.4045	Rand.
15. Test of random excursions	0.5957	Rand.
16. Test of rand. variant Excursions	0.5319	Rand.

4. Concise comparison, analysis, and discussion of the Single, Linear, and Tree (WCSDN) Topologies

After applying, implementing, and testing the proposed algorithms in the three (WCSDN) topologies, the following comparison, analysis, and discussions can be obtained, which can be summarized in the following Table 3.

Table 3. Comparisons, Analyses, and Discussions Between WCSDN Applied Topologies.

Feature	Single (Star) WCSDN	Linear (Chain/Bus) WCSDN	Tree WCSDN
Core Concept	Central controller manages all sensor nodes.	Nodes connected sequentially; control can be distributed.	Hierarchical structure with coordinating nodes.
Control Plane	Centralized at the controller.	Can be distributed or centralized (less common).	Hierarchical; local control at coordinators, global at root.

Table 3. Continued

Feature	Single (Star) WCSDN	Linear (Chain/Bus) WCSDN	Tree WCSDN
Data Plane	Direct links between sensors and controller.	Data flows through intermediate nodes.	Data aggregated at coordinators, then to the root.
Scalability	Limited by central controller’s capacity.	Moderate; performance degrades as the number of hops/nodes increases.	Good; hierarchical structure manages complexity.
Reliability	Single point of failure (controller).	Failure of a node can disrupt the chain.	More fault-tolerant; branch failures isolated.
Complexity	Simple implementation and management.	Moderate; routing and collision management needed.	More complex; requires sophisticated control and routing.
Latency	Low for direct communication.	Higher due to multi-hop communication.	Moderate; depends on the tree depth.
Power Efficiency	Can be managed centrally; direct links can be efficient for short ranges.	Can be inefficient due to relaying; nodes consume power forwarding data.	Can be efficient with optimized aggregation and routing within sub-trees.
Cost	Generally lower initial cost for small deployments.	Moderate cost; can increase with complex routing.	Higher initial cost due to controllers/coordinators.
Best Use Cases	Small, localized areas; smart homes.	Linear deployments (e.g., pipelines); less dense areas.	Larger areas, multi-level structures, varying node density.
Analysis	Simple but vulnerable and less scalable.	Easy to extend linearly but prone to failures and latency.	Balances scalability and fault tolerance but more complex.
Discussion	Centralized control simplifies management but creates bottlenecks.	Distributed control offers resilience but complicates coordination.	Hierarchical control enables scalability and manageability but requires careful design.

Conclusions

In this article three different scenarios were created from the WCSDN network which is (Single, Linear, Tree) and the use of ten text files of different sizes to send over the network because of the central nature of the WCSDN and its programmability led to exposure to many attacks and try to penetrate it, so it was done developing an encryption algorithm to protect it called the DEA algorithm.

To evaluate the (SG) algorithm and the developed encryption algorithm (DEA), one test (N.I.S.T) is chosen to verify the outcomes of the (SG) and DEA algorithms and to compare their results. On the other hand, this article measures the encryption runtime for the SG and DEA algorithms by encrypting different data blocks and comparing the results. The results show that the developed encryption algorithm (DEA) has lower encryption/decryption run time compared with the (SG) algorithm. From the above, the practical benefit of using and applying the proposed algorithms is clear: they are fast and suitable for all software-defined network topologies, enabling encryption and decryption to protect information against cyber-attacks. As suggestions for future work, the researcher can evaluate the performance of other open-source SDN controllers, such as RunOS and Trema, to develop this system, and evaluate the performance of SDN controllers using different simulation tools, such as EstiNet and NS-3. An important point that all researchers in this field should consider is that it is very necessary to adopt information encryption and follow safe steps in achieving secure networking and protecting information servers by providing them with firewalls and the necessary electronic applications that protect them from cyber-attacks.

Acknowledgements

All authors would like to thank the presidency and professors of Mustansiriyah University, and the University of Baghdad for their support and assistance in completing this research.

Funding

None of the authors received any sponsorship to produce the research reported in this article.

Conflict of Interest

For this article the authors do not have any type of conflict of interest to declare.

References

- Adekunle, O. O., (2015), "A Security Architecture for Software Defined Networks (SDN) - ProQuest," vol. 13, no. 7, pp.56–62.
- Aula. Riyadh. Nasser, and Mahmood Zaki Abdullah (2023), "Optimizing DDoS Attack Classification in SDN Networking via Combining Features", IEEE Xplore: 21 August 2024, 2023 International Conference on Engineering Applied and Nano Sciences (ICEANS) 23-24, pp 114-120.
<https://doi.org/10.1109/ICEANS58413.2023.10629673>.
- Braun, W., & Menth, M. (2014). Software-defined networking using OpenFlow: Protocols, applications and architectural design choices. *Future Internet*, 6(2), 302-336.
<https://doi.org/10.3390/fi6020302>
- Chao, T. W., Ke, Y. M., Chen, B. H., Chen, J. L., Hsieh, C. J., Lee, S. C., & Hsiao, H. C. (2016). Securing data planes in software-defined networks. In *2016 IEEE NetSoft Conference and Workshops (NetSoft)* (pp. 465-470). IEEE.
<https://doi.org/10.1109/NETSOFT.2016.7502486>.
- Ertaul, L., & Venkatachalam, K. (2017). Security of software defined networks (SDN). In *Proceedings of the International Conference on Wireless Networks (ICWN)* (pp. 24-30). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
<https://www.proquest.com/docview/2139473729/abstract/B70267AD59A4415PQ/1?accountid=14598>
- Fadia Noori Al-Nuaimy, Ibrahim Amer Ibrahim, Ali Saadi Hamza, and Adile Faeq Naji, (2020), "Design and Implement Whitening Teeth Kit Using Normal Toothpaste", International Journal of Software & Hardware Research in Engineering, Vol.8, Issue 2, February 2020,
<https://doi.org/0.26821/IJSHRE.8.2.2020.8203>
- Fadia Noori Hummadi Al-Nuaimy, (2020), "A new eliminating EOG artifacts technique using combined decomposition methods with CCA and H.P.F. techniques", TELKOMNIKA Telecommunication, Computing, Electronics and Control, Vol. 18, No. 5,
<https://doi.org/10.12928/TELKOMNIKA.v18i5.14143>.
- Hasan Kamel, and Mahmood Zaki Abdullah (2022), "A new approach of extremely randomized trees for attacks detection in software defined network", Indonesian Journal of Electrical Engineering and Computer Science, Vol. 28, No. 3, pp. 1613~1620.
<http://doi.org/10.11591/ijeecs.v28.i3.pp1613-1620>.

- Hasan Kamel, and Mahmood Zaki Abdullah (2022), "Distributed denial of service attacks detection for software defined networks based on evolutionary decision tree model", Bulletin of Electrical Engineering and Informatics, Vol. 11, No. 4, pp. 2322~2330, August 2022.
<https://doi.org/10.11591/eei.v11i4.3835>.
- Husham Salam Saeed , Muhammad Hassan Fares (2025), "Enhancing Surveillance with Machine and Deep Learning-Based Facial Recognition Model: A Proposed Approach for Identification", Journal of Engineering and Sustainable Development, Vol. 29, No. 1,
<https://doi.org/10.31272/jeasd.2633>.
- Inas Jawad Kadhim, Tawfeeq E. Abdulabbas, Riyadh Ali, Ali F. Hassoon, Prashan Premaratne (2024), "A Enhanced Speech Command Recognition using Convolutional Neural Networks", Journal of Engineering and Sustainable Development, Vol. 28, No. 6,
<https://doi.org/10.31272/jeasd.28.6.8>.
- Irfa, A., and S. A. Mahm., (2015), A novel secured SDN based architecture for grid security, Pro. -15th IEEE Int. Con. Comp. In. Tech. CIT 2015, Auto.. Se, pp.762–769.
- Manasyan, A. D. (2022). Network Management Automation Through Virtualization. *Mathematical Problems of Computer Science*, 58, 91-98.
<https://doi.org/10.51408/1963-0096>.
- Jawaharan, R., P. M. Mohan, T. Das, and M. Gurusamy, (2018), "Empirical Evaluation of SDN Controllers Using Mininet/Wireshark and Comparison with Cbench," 2018 27th Int. Conf. Comput. Commun. Networks, pp. 1–2.
- Kaur, M., Jain, V., Nand, P., & Rakesh, N. (Eds.). (2024). *Software-Defined Network Frameworks: Security Issues and Use Cases*. CRC Press.
<https://doi.org/10.1201/9781040018323>
- Mahmood Zaki Abdullah, Ali Khalid Jassim, Fadia Noori Hummadi, and Mohammed Majid, (2024), "NEW STRATEGIES FOR IMPROVING NETWORK SECURITY AGAINST CYBER ATTACK BASED ON INTELLIGENT ALGORITHMS", Journal of Engineering and Sustainable Development, Vol. 28, No. 3.
<https://doi.org/10.31272/jeasd.28.3.4>.
- Maysaa Hameed Abdulameer, Mahmood Zaki Abdullah, Ali Khalid Jassim, and Mohammed Majid M. Al Khalidy (2024), "A Hybrid for Analyzing Text Streaming Using Data Mining and Machine Learning Techniques", Journal of Engineering and Sustainable Development, Vol. 28, No. 5, September 2024,
<https://doi.org/10.31272/jeasd.28.5.13>.
- Nayyar, A., Singla, B., & Nagrath, P. (Eds.). (2022). *Software Defined Networks: Architecture and Applications*. John Wiley & Sons.
<https://doi.org/10.1002/9781119857921>.
- Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turlatti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, 16(3), 1617-1634.
<https://doi.org/10.1109/SURV.2014.012214.00180>
- Peterson, L., Cascone, C., & Davie, B. (2021). *Software-defined networks: a systems approach*. Systems Approach, LLC.
<https://github.com/SystemsApproach/SDN>.
- Prajapati, A., Sakadasariya, A., & Patel, J. (2018). Software defined network: Future of networking. In *2018 2nd international conference on inventive systems and control (ICISC)* (pp. 1351-1354). IEEE.
<https://doi.org/10.1109/ICISC.2018.8399028>
- Python Software Foundation (2023). PySerial 3.5 documentation. pySerial' Available at:
<https://pythonhosted.org/pyserial/>.
- Rana, D. S., S. A. Dhondiyal, and S. K. Chamoli, (2019), "Software Defined Networking (SDN) Challenges, issues and Solution" , International Journal of Computer Sciences and Engineering, vol. 7, no. 1. pp. 884–889.
- Sabreen Waheed Kadhum, Mohammed Ali Tawfeeq (2025), "Improving Performance Classification in Wireless Body Area Sensor Networks Based on Machine Learning Techniques", Journal of Engineering and Sustainable Development, Vol. 29, No. 1,
<https://doi.org/10.31272/jeasd.2491>.
- Sco.Hawar, S., G. O'Callaghan, and S. Sezer, (2013), "SDN security: A survey," SDN4FNS 2013 - 2013 Work. Software Defi.. Networks Futur. Networks Serv.
- Taqwa Oday Fahad, Abbass Hussien Miry, Ammar Al-Gizi, Mohammed Hussein Miry, Ahmed Talib Razzooqee (2024), "Recognition of Underwater Acoustic Radar Signals Based on Multiresolution and Dense Convolutional Neural Network", Journal of Engineering and Sustainable Development, Vol. 28, No. 6.
<https://doi.org/10.31272/jeasd.28.6.12>.