Enhanced Security of Software-defined Network and Network Slice Through Hybrid Quantum Key Distribution Protocol

Suadad S. Mahdi[1,2] and Alharith A. Abdullah[1]

***Abstract*—Software-defined networking (SDN) has revolutionized the world of technology as networks have become more flexible, dynamic and programmable. The ability to conduct network slicing in 5G networks is one of the most crucial features of SDN implementation. Although network programming provides new security solutions of traditional networks, SDN and network slicing also have security issues, an important one being the weaknesses related to openflow channel between the data plane and controller as the network can be attacked via the openflow channel and exploit communications with the control plane. Our work proposes a solution to provide adequate security for openflow messages through using a hybrid key consisting of classical and quantum key distribution protocols to provide double security depending on the computational complexity and physical properties of quantum. To achieve this goal, the hybrid key used with transport layer security protocol to provide confidentiality, integrity and quantum authentication to secure openflow channel. We experimentally based on the SDN-testbed and network slicing to show the workflow of exchanging quantum and classical keys between the control plane and data plane and our results showed the effectiveness of the hybrid key to enhance the security of the transport layer security protocol. Thereby achieving adequate security for openflow channel against classical and quantum computer attacks.**

***Index Terms*—hybrid key, openflow protocol, quantum key distribution, software-defined networking, network slicing, transport layer security.**

I. INTRODUTION

Software-defined networking (SDN) is an emerging and rapidly growing technology that separates the control plane from the network devices in order to give more flexibility to control the network, based on specific policies and security enforcements [1]. The advantages of SDN and virtualization have inspired the creation of network slicing (NS) and contributed to build the infrastructure of 5G networks [2].

NS came up to address the problem of growing network services [3]. Previously, the prevailing concept in networks was "one size fits all", but this concept does not apply to the fifth-generation networks and beyond, the reason is due to different network requirements of heterogeneous applications.

[1] University of Babylon, Babil, Iraq;
E-mail: {suadadsafaa, alharith}@itnet.uobabylon.edu.iq
[2] Al-Mustaqbal University College, Babil, Iraq;
E-mail: suadad.safaa@mustaqbal-college.edu.iq

DOI: 10.36244/ICJ.2022.3.2

Today, with network virtualization technology and software-defined networks, and the ability to abstract resources, the concept of network slicing is ready to create programmable network slices isolated from each other and release them to the real world.

The concept of network slicing is depicted in Figure 1, which allows for the establishment of logical networks for various types of services.

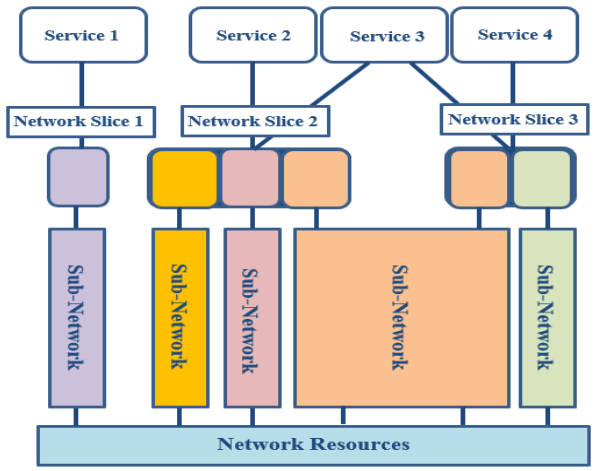

Fig. 1. Network slicing concept.

Despite the flexibility of SDNs, it has properties that can be considered as traps for network attackers such as the network information's centralization in the controller [4]. As a result, anyone can reach the servers hosting the control software, also, centralization means the SDN architecture has a one point of failure that makes the attackers direct their attacks to the controller. For instance, a malicious application (or controller) can be employed to reprogram the whole network to purposes of data stealing from the data center.

The controller has been able to enhance the network security by taking advantage of the global network view feature by running security applications in the controller in order to detect attacks and thus address them in addition to helping to understand the nature of the network in various threats, incidents and security vulnerabilities [5]. The controller uses the information collected and analyzed to enforce the appropriate security policies and thereby improve data plane security. But the idea of the SDN that stems from the decouple of control part from network devices has led to

the need to add new components to the network, namely the controller and communication channels between the planes, which include many challenges that pose security problems that deserve attention.

The Fig. 2 illustrates some critical security threats in SDN. Some of them are popular in the present networks and some other threats are more specific in SDN [6]. But the most dangerous attack is the one which exploits any vulnerability to access the controller and thus destroys the entire network.

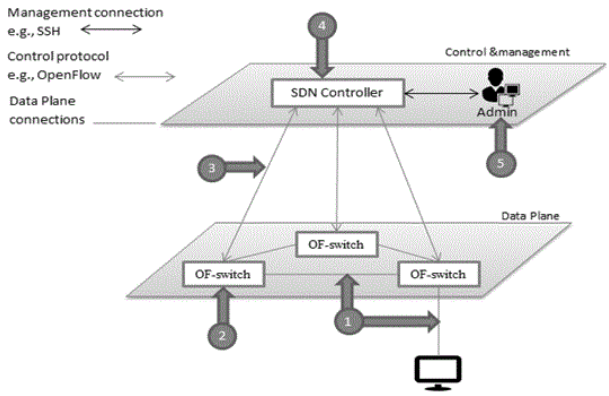

Fig. 2. Security threats in SDN architectures.

As shown in the figure, the threat 1, 2 and 5 are already present in traditional networks. On the contrary, from the threats 3 and 4 are specific to SDN and that stem from decoupling the controller from the network devices and making them centralized. Third threat is seen as the most dangerous, because the network process can be compromised. This type of threat is specific to SDN networks. The attacks focus on control plane communications (such as openflow channel) through which denial of service attacks or data theft are generated.

The danger of this attack depends on the access to the control plane, then it is able to collect enough power (in terms of the number of SDN switches[1] under its control) to launch distributed denial-of-service (DDoS) attacks, in additional to Man-in-the-Middle (MitM) attacks. This is due to a lack of authentication between the controller and SDN switches that makes it easier to create a virtual black hole network allowing data to leak during normal production flows. In essence, Transport Layer Security (TLS) protocol is necessary to protect connections in the SDN. However, relying on TLS use alone is not enough and as is well known in the security community, the use of SSL / TLS does not guarantee a secure connection especially with the advent of quantum computers [7][8], which can perform calculations very quickly, and have significantly affected classic security protocols.

The progress in quantum physics has led to thinking of new ways to ensure security in communication [9]. Quantum key distribution (QKD) is a suitable technology for securing network communication channels, where a single or entangled quantum state is transferred between two parties [10]. Each of parties has two channels: the quantum channel for the exchange of photons and the classic public channel to check for eavesdropping [11]. If a third party makes measurement of the transferred quantum, both of party will discover an eavesdropper presence on the communication channel based on the rules of the mechanics of quantum and the no-cloning theorem [12]. Several researchers have presented work on using quantum protocols to achieve key distribution instead of using traditional cryptographic methods (e.g., RSA), where Czermann, Márton et al. [13] demonstrated the successful distribution of quantum keys using the BB84 protocol in practice on a fiber-optic system.

This work explained how to use hybrid key [14] for key exchange and thus secure openflow channel via quantum TLS (QTLS) in SDN and NS. This paper is an extension of the work in [15] but in this paper the methodology is proposed on the NS environment. In particular, the hybrid key is considered the best solution to achieve authentication between the two parties based on quantum properties in addition to providing a strong key to supply double security based on mathematical complexity of classical method and physical properties of quantum protocol.

This paper is organized as follows. Section II clarifies related work and reveals its boundaries. Section III explains the steps and workflow used for securing openflow channel. The implementation, along with some of our test findings and evaluations, are presented in Section IV. Section V discusses security analysis, while this paper concludes in Section VI.

II. RELATED WORK

In this section, previous studies concerning openflow security in SDN and NS were shown. In [16] authors proposed a quantum key distribution (QKD) and encryption algorithm one time pad (OTP) to encrypt openflow protocol messages, which is named QKDFlow. This scheme is considered a solution that aims to block the MitM attack and thus secure openflow messages in SDN. While the researchers at [17] suggested an identity-based cryptography protocol (IBC) to secure software-defined networks connections, especially for controller communications with network devices within the data plane. Where they suggested that the role of private key generator (PKGs) be transferred to the controller to create the private keys for the network devices and thus reduce the load of PKGs managing the controller.

The authors in [18] discussed security for openflow communication protocol in SDNs by using the security protocol TLS and discussing the security loopholes in TLS. As a result, they proposed a change in TLS handshake protocol to achieve authentication between the parties by adding messages containing the random number, timestamp and hello message ID to revalidation of client and server status before sending finished messages. Therefore, based on this, the MitM attacker is prevented because of the timestamp of the response since the time taken by the attacker to decrypt a random number would certainly exceed the timeframe of the client's response to the server request. While, authors explain in [19] a method for detecting DDoS attacks in the SDN depended on the quantum parameters of QKD system, such as the secret key rate (SKR) and quantum bit error rate (QBER). Where the controller monitors the QBER, if the rate exceeds the

[1] Term SDN switch is example of data plane devices.

threshold limit, the controller makes a decision to change the path and thus mitigate DDoS.

The authors in [20] offered a solution for achieving effective and secure service-oriented authentication for 5G IoT applications, including network slicing and fog computing, to assure anonymity, user credibility, and service data confidentiality. Users are authenticated by utilizing access credentials produced by the IoT server, which allow them to access the IoT service. Otherwise, the attacker would be unable to do so without a legitimate access credential. While authors in [21] developed a hybrid strategy to protect communications between 5G network slices in distinct public cryptosystems, and two heterogeneous cipher schemes to achieve reciprocal communications between the public key infrastructure (PKI) and Certificate Less Public Key Cryptography (CLC) environments.

The authors in [22] introduced a security solution to address security problems related to data exchange in software-defined networks. Where proposed that the TLS use of protocol between the SDN nodes to provide adequate security for communication channel. In addition to use an integrated security module to enhance the security of communications through the application of the access control list (ACL), Strengthening of the TLS protocol configuration and contribute to minimizing the impact of private key hijacking.

Authors in [23] propose a key-distribution scheme suitable for the network slicing architecture when the slices are accessed by third-party applications. The proposed scheme consists of two technologies, the first is Shamir's secret sharing to distribute and rebuild private key shares, and the second technique is ElGamal cryptosystem to encrypt and decrypt the separator keys.

The authors in [24] focused on exploring the concerns of a distributed denial of service attack on a network slicing and presented a model based on deep learning to create a robust network slicing framework to proactively combat DDoS attacks and eliminate overburdened connections before they impact and invade 5G networks.

While the researchers in [15] presented a mechanism for implementing a quantum hybrid protocol with the classic protocol to achieve security for the openflow channel by encrypting messages between the controller and network devices in the software-defined networks.

III. Proposed Methodology for Securing OpenFlow Channel

The use of authentication and encryption is the most important security measures to protect communications. So, in the proposed methodology, the hybrid key [14] is used in the TLS protocol to add new way of authentication based on physical properties of quantum as well as improving encryption process depending on the hybrid key produced by two systems, the first one depends on the computational complexity and the other depends on the quantum properties of the QKD.

The main goal is to secure the communication between the controller and network devices in SDN and NS, thus secure openflow messages. So, there are many messages between controller and SDN switches before exchanging encrypted openflow messages as shown in Figure 3.

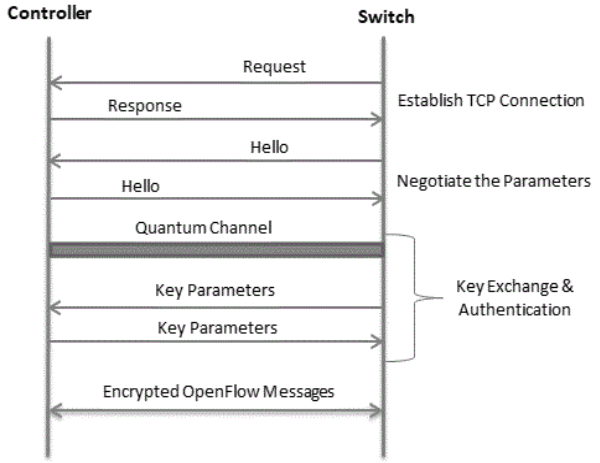

Fig. 3. Controller-switch channel.

In the first step, TCP connection was established, the network device in data plane initiates a request to connect the controller based on the IP and port of controller, and then it responds to this request.

In the second step, session parameters were negotiated to open a secure channel between controller and SDN switch (in data plane) over TLS, Hello message contains the main security parameters.

While in the third step, the key exchange begins between the controller and SDN switch, which is used in the encryption. This step begins by opening a quantum channel and achieving the authentication based on the quantum properties. Then the quantum and classical parameters are exchanged to establish a hybrid key. These steps are known as TLS handshake protocol.

Finally, the hybrid key is passed to the TLS Record protocol to encrypt the communication channel between the controller and SDN switch by using the AES-256 encryption algorithm, thus confidentiality for openflow messages is achieved.

In Figure 4 we summarized the operation of the proposed methodology where a switch initiates a connection to the controller. When the controller receives the connection request, it checks whether the switch supports QKD protocol. If not, the connection will be established by use standard TLS protocol. Otherwise, controller will open the quantum channel and exchange the parameters of the hybrid key. Then, the controller checks if quantum bit error rate (QBER) is less than the threshold (threshold limit of 0.5 has been used) then pass otherwise the key exchange phase is repeated. After this the hybrid key is generated and the quantum TLS handshake protocol is completed, and communication based on openflow between the controller and the switch begins via the QTLS channel.

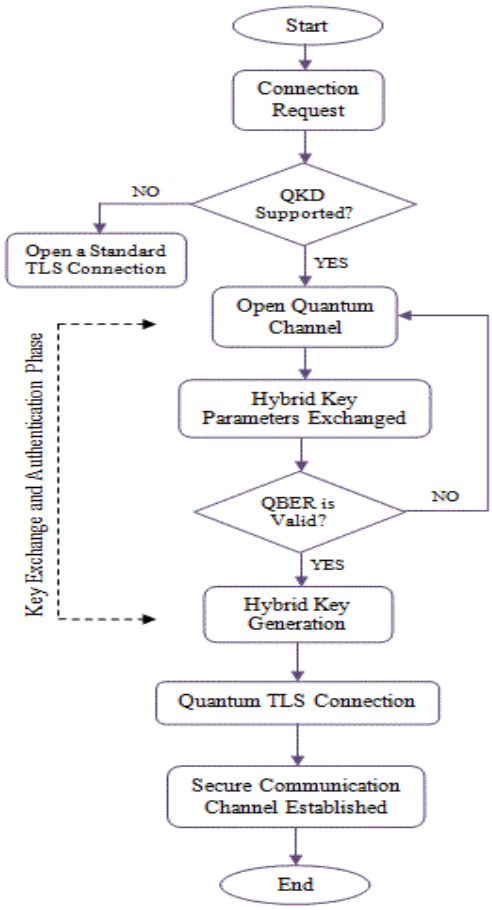

Fig. 4. Operation of the proposed methodology

IV. Implementation Results and Evaluation

A. SDN-Testbed

To do our work, the network environment is initialized depended on the SDN testbed as explain in Figure 5.

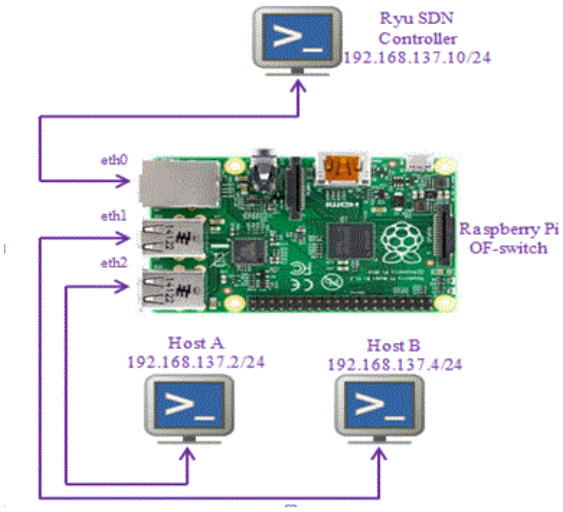

Fig. 5. Environment of SDN-testbed.

In our work, raspberry pi board was converted to openflow switch (SDN switch) to use in our SDN-testbed. We can install and configuring the open vswitch to convert Raspberry Pi as an openflow switch based on the following steps:

- The "ovs-vsctl" was used to create a bridge and add three interfaces (eth0, eth1, eth2) to it through using the add-port command.
- Interfaces were configured and they enable the links connection with three Laptops via Ethernet interface through by using USB-to-Ethernet adapters. One Laptop was used as a ryu controller to manage the open vswitch remotely and two laptops as hosts for test.
- The openflow protocol 1.3 was enabled on the bridge and used the -O option to enable the openflow version in ovs-ofctl.
- Then, the bridge was connected to a remote controller by using IP address and port number of SDN controller.
- Next, the fallback mode of OVS was set to secure mode.

Based on the previous steps, the raspberry pi was converted to openflow switch in low cost [25]. While ryu controller was used depending on Ubuntu 18.04 to configuration of the openflow protocol to allow the controller to program the openflow switch (raspberry pi).

B. Network Slicing Implementation

The scenario for implementing network slicing based on SDN consists of several controllers and a single owner. SDN proxy (Network Hypervisor) plays a key role in dividing the network infrastructure into many virtual networks, and the infrastructure owner is usually the person who controls the SDN proxy [26].

Multiple virtual tenants can use this scenario to deploy their SDN controllers on the network slices management infrastructure and maintain isolation between them. Figure 6 shows the structure of the SDN proxy in the slicing environment.

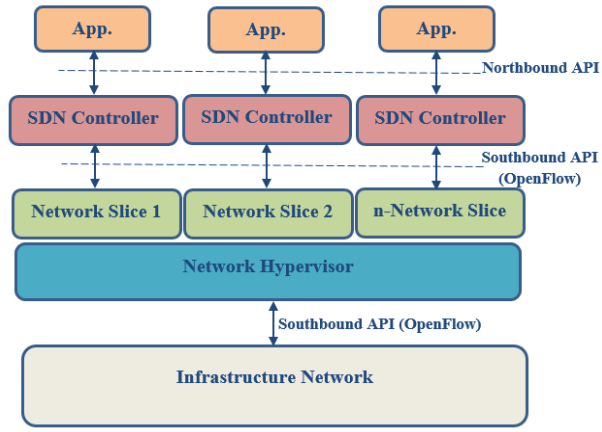

Fig. 6. Architecture of network slicing based on SDN.

As indicated in Figure 7, one of the most important hypervisors used to achieve this scenario is FlowVisor, which works as an SDN proxy intercepting messages between the data layer and the control layer [27].

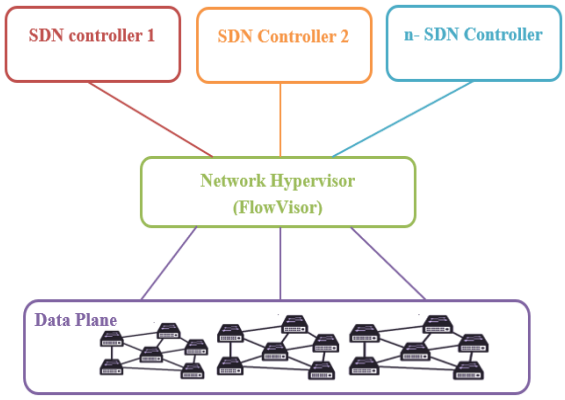

Fig. 7. Network slicing with SDN proxy.

FlowVisor is an infrastructure resource virtualization layer that enables the creation of multiple network slices, and each slice includes a dedicated SDN controller. With FlowVisor, network slices are conceptually separated from each other, and communication between the infrastructure and FlowVisor takes place through the OpenFlow protocol, as well as between FlowVisor and SDN controllers.

Therefore, securing the openflow channel is important in NS, as it represents the communication channel between the infrastructure layer and the virtualization layer, as well as between the virtualization layer and the control layer.

C. Experimental Results

To explain the proposed hybrid solution, we have implemented the hybrid keys to secure openflow channel, through incorporating the hybrid security into TLS protocol.

Figure 8 shows the initial messages to exchange hybrid key parameters, also TLS version that is used to openflow channel.

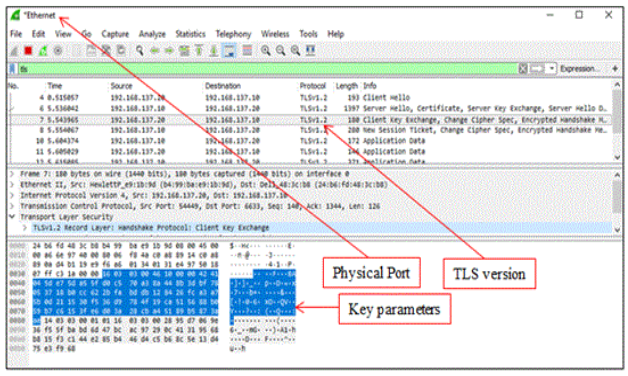

Fig. 8. Initial messages for key agreement.

Our experimental results indicated that the time required to implement the proposed QTLS protocol is acceptable to establish a secure connection compared to the standard TLS protocol. Figure 9 shows the time difference of implementation the standard TLS protocol and proposed QTLS.

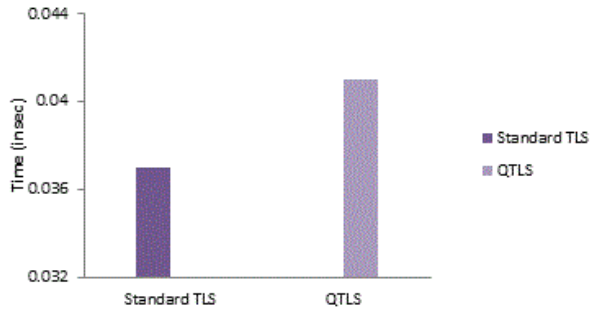

Fig. 9. Required time for implementing the standard TLS and QTLS protocols.

The reason for the time difference is due to the increased complexity within handshake in QTLS protocol, but this is at the expense of increasing security and authentication for key exchange between the two parties.

On the other hand, the randomness of the obtained hybrid key was measured by depended on six randomized NIST tests. The p-value was observed ≥ 0.01, and thus the binary sequence of hybrid key is more randomness, as explained in Table 1.

TABLE I
NIST (RANDOMNESS TESTS) RESULTS

	Key1	Key 2	Key 3	Key 4	Key 5
Frequency Test	0.725	0.508	0.536	0.965	0.595
Block Frequency (n = 128)	0.764	0.952	0.684	0.986	0.724
Runs	0.382	0.165	0.257	0.809	0.732
Longest Block Run	0.253	0.265	0.745	0.498	0.530
Approximate Entropy	1	1	1	1	1
Cumulative Sums	0.895	0.698	0.778	0.972	0.856

While we calculated the key space of the hybrid key and the results showed that the length of the key is large enough to make it impossible for the brute-force attacks to search for all possible keys using classic and quantitative computing. It has been shown [7] that a brute-force key search on a quantum computing cannot be faster than about $2^{n/2}$ when compared with about 2^{n} in the classical computing. Therefore, the hybrid key can be considered safe against quantum brute force attack, as shown in Table 2 the quantum and classical security levels for hybrid keys.

TABLE II
QUANTUM AND CLASSICAL SECURITY LEVELS FOR HYBRID KEYS

Keys	Key Length	Key Space	Security Level (in bits)	
			Classical Computing	Quantum Computing
Key 1	512	2^{512}	512	256
Key 2	1024	2^{1024}	1024	512
Key 3	2048	2^{2048}	12048	1024

D. Performance Evaluation and Comparison

As introduced in section IV.A, SDN-testbed was relied upon to implement our work. So in this section we review the

evaluation of network performance depending on throughput using iperf tests.

Analytically, throughput can be defined as the rate of maximum receiver bandwidth (Max BW) to round trip time (RTT) between hosts, where host A sends the number of packets to host B using the iperf tool.

Figure 10 show throughput according to different sizes of the segment size (64, 128, 512, 1024, and 1400) bytes.

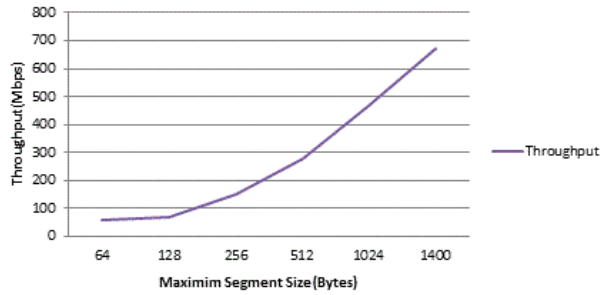

Fig. 10. Throughput of data transfer in the secure SDN-testbed.

Additionally, the results of SDN-testbed was compared with net-FPGA results after being converted to openflow switch [28], as explained in Figure 11. The result concluded that the result of our SDN-testbed is approximately similar to the performance of openflow switch based on net-FPGA hardware.

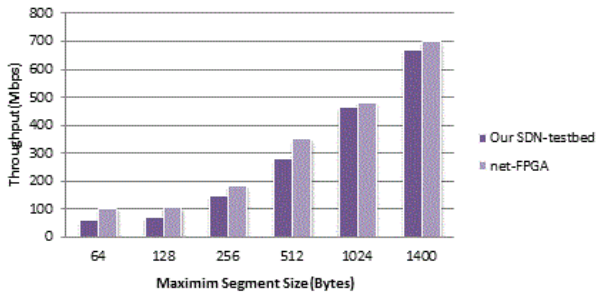

Fig. 11. Throughput in our SDN-testbed and net-FPGA.

Most of the previous research stresses the increase in the security of the openflow communication channel through the use of the standard TLS protocol, but there are security gaps in phase of exchanging the keys within TLS handshake protocol. Also, the emergence of quantum computers and the use of grover's and shor's algorithms have made it easy to break most classic cryptography protocols [29], and thus classic solutions have become inadequate for the purpose of securing communications.

In our work, the focus was on using the quantum keys distribution protocol with classical protocol to add a new layer of security based on quantum laws to increase authentication and security for openflow communication channel between the controller and network devices. Our experimental results showed that the hybrid key enhances the authentication and security of key exchange between the two parties in the QTLS protocol compared to the classic methods used for key exchange in the standard TLS protocol.

Our results were based on an analysis of the effect of classical and quantum computers on hybrid keys. The results showed that the hybrid key has the physical properties of the quantum in addition to the mathematical complexities, which make hybrid key difficult to break using the quantum or classical computer.

V. SECURITY ANALYSIS

In our work, the hybrid key was used to achieve authentication between the two parties at two levels in addition to using it to encrypt the channel. First Level: through the classical methods using exchange of certificates between the two parties while the second level: physical authentication through the quantum channel of QKD protocol and exploiting of the physical properties of quantum [12]. Therefore, It can be said that this study has achieved more secure authentication between the control plane and the data plane in SDN as well as between the virtualization layer and control layer in NS addition to secure the communication channel between the data layer and the virtualization layer. Therefore, it can be said that our proposal helps to avoid MitM attacks and achieve authentication mechanism on the openflow messages flowing through the channel.

On the other hand, in our work we provided a strong and reliable key for encrypting the communication channel, thus preventing data modification and providing a high level of confidentiality to openflow messages. In this way, the openflow communication channel was protected. As well, TLS protocol has been used with the hybrid key for securing communication and, as known, the TLS handshake protocol uses the nonce value and timestamp to prevent replay attacks.

VI. CONCLUSION

In recent times, SDN has been developed significantly as a result of high flexibility and programmability and SDN technology had seen as one of the most promising enablers of network development, which will play an essential role in the design of 5G networks through network slicing technology. Although it is promising in terms of cost reduction, it contains some security vulnerabilities that need solutions to address them.

In our work, we relied on enhancing the security of TLS protocol by using a hybrid key based on the mathematical complexities and the physical properties of the quantum. We have achieved sufficient security of openflow communication channel which is the basis of communication between layers of SDN networks as well as in NS. This security came about by fending off classical and quantum computer attacks by adding a new quantum security layer to TLS, as well as enhancing authentication between layers based on quantum properties.

The current work is effective to reduce the risk of attacks that threaten the security of the openflow communication channel.

REFERENCES

[1] Masoudi, R., & Ghaffari, A., 2016. Software defined networks: A survey. Journal of Network and computer Applications, 67, pp. 1–25. DOI: 10.1016/j.jnca.2016.03.016

[2] Alotaibi, D., 2021. Survey on network slice isolation in 5G networks: fundamental challenges. Procedia Computer Science, 182, pp. 38–45. DOI: 10.1016/j.procs.2021.02.006

[3] Zhang, S., 2019. An overview of network slicing for 5G. IEEE Wireless Communications, 26(3), pp. 111–117. DOI: 10.1109/MWC.2019.1800234

[4] Lee, S., Kim, J., Woo, S., Yoon, C., Scott-Hayward, S., Yegneswaran, V., Porras, P. and Shin, S., 2020. A comprehensive security assessment framework for software-defined networks. Computers & Security, 91, p. 101720. DOI: 10.1016/j.cose.2020.101720

[5] Nisar, K., Welch, I., Hassan, R., Sodhro, A.H. and Pirbhulal, S., 2020. A survey on the architecture, application, and security of software defined networking. Internet of Things, p. 100289. DOI: 10.1016/j.iot.2020.100289

[6] Yurekten, O. and Demirci, M., 2021. Citadel: Cyber threat intelligence assisted defense system for software-defined networks. Computer Networks, 191, p. 108013. DOI: 10.1016/j.comnet.2021.108013

[7] Kumar, M., 2021. Quantum Computing and Post Quantum Cryptography. International Journal of Innovative Research in Physics, 2(4), pp.37-51. DOI: 10.15864/ijiip.2405

[8] Sadkhan, S. B., Abbas, M. S., Mahdi, S. S., & Hussein, S. A., 2022, March. Software-Defined Network Security-Status, Challenges, and Future trends. In 2022 Muthanna International Conference on Engineering Science and Technology (MICEST), pp. 10–15. IEEE. DOI: 10.1109/MICEST54286.2022.9790219

[9] Portmann, C. and Renner, R., 2021. Security in quantum cryptography. arXiv preprint arXiv:2102.00021. DOI: 10.1103/RevModPhys.94.025008

[10] Bennett, C.H. and Brassard, G., 2020. Quantum cryptography: Public key distribution and coin tossing. arXiv preprint arXiv:2003.06557. DOI: 10.1016/j.tcs.2014.05.025

[11] Sun, S., & Huang, A., 2022. A review of security evaluation of practical quantum key distribution system. Entropy, 24(2), 260. DOI: 10.3390/e24020260

[12] Chen, Y., Gong, M., Xue, P., Yuan, H. and Zhang, C., 2021. Quantum deleting and cloning in a pseudo-unitary system. arXiv preprint arXiv:2103.15353. DOI: 10.1007/s11467-021-1063-z

[13] Czermann, M., Trócsányi, P., Kis, Z., Kovács, B., & Bacsárdi, L., 2021. Demonstrating BB84 quantum key distribution in the physical layer of an optical fiber based system. Infocommunications Journal, 13(3), pp. 45–55. DOI: 10.36244/ICJ.2021.3.5

[14] Abdullah, A.A. and Mahdi, S.S., 2019. Hybrid Quantum-Classical Key Distribution. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8(12), pp. 4786–4791. DOI: 10.35940/ijitee.L3682.1081219

[15] Mahdi, S. S., & Abdullah, A. A., 2022, March. Improved Security of SDN based on Hybrid Quantum Key Distribution Protocol. In 2022 International Conference on Computer Science and Software Engineering (CSASE), pp. 36–40. IEEE. DOI: 10.1109/CSASE51777.2022.9759635

[16] Peng, Y., Wu, C., Zhao, B., Yu, W., Liu, B. and Qiao, S., 2016, November. QKDFlow: QKD based secure communication towards the openflow interface in SDN. In International Conference on Geo-Informatics in Resource Management and Sustainable Ecosystem (pp. 410–415). Springer, Singapore. DOI: 10.1007/978-981-10-3969-0_45

[17] Lam, J., Lee, S.G., Lee, H.J. and Oktian, Y.E., 2016. Securing SDN southbound and data plane communication with IBC. Mobile Information Systems, 2016. DOI: 10.1155/2016/1708970

[18] Agborubere, B. and Sanchez-Velazquez, E., 2017, June. Openflow communications and tls security in software-defined networks. In 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) (pp. 560–566). IEEE. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.88

[19] Hugues-Salas, E., Ntavou, F., Ou, Y., Kennard, J.E., White, C., Gkounis, D., Nikolovgenis, K., Kanellos, G., Erven, C., Lord, A. and Nejabati, R., 2018, March. Experimental demonstration of DDoS mitigation over a quantum key distribution (QKD) network using software defined networking (SDN). In 2018 Optical Fiber Communications Conference and Exposition (OFC) (pp. 1–3). IEEE. DOI: 10.48550/arXiv.1802.05679

[20] Ni, J., Lin, X., & Shen, X. S., 2018. Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT. IEEE Journal on Selected Areas in Communications, 36(3), pp. 644–657. DOI: 10.1109/JSAC.2018.2815418

[21] Liu, J., Zhang, L., Sun, R., Du, X., & Guizani, M., 2018. Mutual heterogeneous signcryption schemes for 5G network slicings. IEEE Access, 6, p.p.7854-7863. DOI: 10.1109/ACCESS.2018.2797102

[22] Yigit, B., Gur, G., Tellenbach, B. and Alagoz, F., 2019. Secured communication channels in software-defined networks. IEEE Communications Magazine, 57(10), pp. 63–69. DOI: 10.1109/MCOM.001.1900060

[23] Porambage, P., Miche, Y., Kalliola, A., Liyanage, M., & Ylianttila, M., 2019. Secure keying scheme for network slicing in 5G architecture. In 2019 IEEE Conference on Standards for Communications and Networking (CSCN) (pp. 1-6). IEEE. DOI: 10.1109/CSCN.2019.8931330

[24] Thantharate, A., Paropkari, R., Walunj, V., Beard, C., & Kankariya, P., 2020. Secure5G: A deep learning framework towards a secure network slicing in 5G and beyond. In 2020 10th annual computing and communication workshop and conference (CCWC) (pp. 0852-0857). IEEE. DOI: 10.1109/CCWC47524.2020.9031158

[25] Gupta, V., Kaur, K. and Kaur, S., 2018. Developing small size low-cost software-defined networking switch using raspberry pi. In Next-generation networks (pp. 147–152). Springer, Singapore. DOI: 10.1007/978-981-10-6005-2_16

[26] Blenk, A., Basta, A., Reisslein, M., & Kellerer, W., 2015. Survey on network virtualization hypervisors for software defined networking. IEEE Communications Surveys & Tutorials, 18(1), pp. 655–685. DOI: 10.1109/COMST.2015.2489183

[27] Kurniawan, M. T., Moszardo, I., & Almaarif, A., 2022, June. Network Slicing On Software Defined Network Using Flowvisor and POX Controller To Flowspace Isolation Enforcement. In 2022 10th International Conference on Smart Grid (icSmartGrid), pp. 29–34. IEEE. DOI: 10.1109/icSmartGrid55722.2022.9848585

[28] Naous, J., Erickson, D., Covington, G. A., Appenzeller, G., & McKeown, N., 2008, November. Implementing an OpenFlow switch on the NetFPGA platform. In Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, pp. 1–9. DOI: 10.1145/1477942.1477944

[29] Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Kelsey, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R. and Perlner, R., 2020. Status report on the second round of the NIST post-quantum cryptography standardization process. US Department of Commerce, NIST. DOI: 10.6028/NIST.IR.8309

Suadad S. Mahdi received her B.S. degree in 2016 from the College of Information Technology (IT), University of Babylon, Iraq, in 2020, her MS degree in Information Networks from the College of Information Technology (IT). Her main interests include Software-Defined Networks, Cryptography, Steganography and Quantum Cryptography.

Alharith A. Abdullah received his BS degree in Electrical Engineering from Military of Engineering College, Iraq, in 2000, his MS degree in Computer Engineering from University of Technology, Iraq, in 2005, and his PhD degree in Computer Engineering from Eastern Mediterranean University, Turkey, in 2015. His research interests include Security, Network Security, Cryptography, Quantum Computation and Quantum Cryptography.