


Article

Design and Implementation of Lightweight Certificateless Secure Communication Scheme on Industrial NFV-Based IPv6 Virtual Networks

Zeeshan Ashraf ^{1,*} , Adnan Sohail ² and Muddesar Iqbal ^{3,*}¹ Department of Computer Science, The University of Chenab, Gujrat 50700, Pakistan² Department of Computing & Technology, IQRA University, Islamabad Campus, Islamabad 44310, Pakistan; adnan.sohail@iqraisb.edu.pk³ Renewable Energy Laboratory, College of Engineering, Prince Sultan University, Riyadh 11586, Saudi Arabia

* Correspondence: zeeshan@cs.uchenab.edu.pk (Z.A.); m.iqbal@lsbu.ac.uk (M.I.);

Abstract: With the fast growth of the Industrial Internet of Everything (IIoE), computing and telecommunication industries all over the world are moving rapidly towards the IPv6 address architecture, which supports virtualization architectures such as Network Function Virtualization (NFV). NFV provides networking services like routing, security, storage, etc., through software-based virtual machines. As a result, NFV reduces equipment costs. Due to the increase in applications on Industrial Internet of Things (IoT)-based networks, security threats have also increased. The communication links between people and people or from one machine to another machine are insecure. Usually, critical data are exchanged over the IoE, so authentication and confidentiality are significant concerns. Asymmetric key cryptosystems increase computation and communication overheads. This paper proposes a lightweight and certificateless end-to-end secure communication scheme to provide security services against replay attacks, man-in-the-middle (MITM) attacks, and impersonation attacks with low computation and communication overheads. The system is implemented on Linux-based Ubuntu 20.04 virtual machines using Java programming connected to NFV-based large-scale hybrid IPv4-IPv6 virtual networks. Finally, we compare the performance of our proposed security scheme with existing schemes based on the computation and communication costs. In addition, we measure and analyze the performance of our proposed secure communication scheme over NFV-based virtualized networks with regard to several parameters like end-to-end delay and packet loss. The results of our comparison with existing security schemes show that our proposed security scheme reduces the computation cost by 38.87% and the communication cost by 26.08%.

Keywords: authentication; cryptography; key exchange; network function virtualization; Industrial IoT; virtualization



Citation: Ashraf, Z.; Sohail, A.; Iqbal, M. Design and Implementation of Lightweight Certificateless Secure Communication Scheme on Industrial NFV-Based IPv6 Virtual Networks. *Electronics* **2024**, *13*, 2649. <https://doi.org/10.3390/electronics13132649>

Academic Editor: Andrea Bonci

Received: 20 May 2024

Revised: 20 June 2024

Accepted: 24 June 2024

Published: 5 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

All over the world, the internet is moving rapidly towards Internet Protocol version 6 (IPv6) address architectures, with the support of virtualization architectures such as Software-Defined Networks (SDNs) and Network Function Virtualization (NFV) [1]. The IPv6 address architecture provides a large IP address range, efficient segment routing, built-in security, and mobility features [2]. The industrial revolution and technological enhancement in mobile communication have changed the traditional way of networking and have introduced the term “Industrial Internet of Everything (IIoE)”. The Internet of Everything (IoE) is a network connection of people, processes, data, and things [3,4]. Smart homes, smart cities, smart industry, and smart education are some projects of the IoE [5,6]. In the IoE, billions of smart devices and people are connected and exchange data with the help of several applications. Due to the increase in applications on different IoE-based smart networks, security threats have also increased [7]. The communication links

between people or from one machine to another machine are insecure, so authentication and confidentiality are significant concerns [8].

A Next-Generation Network (NGN) transports information and services at a very high speed through software-based devices instead of dedicated physical devices [9]. Data communication services are provided through NFV and SDNs in the NGN. NFV is a new concept and an emerging technology that was introduced in 2012 [10]. The main objectives of NFV are to eliminate hardware equipment and to deliver networking services like routing, security, storage, etc., through software-based virtual machines (VMs). NFV gives many advantages, such as decreasing equipment expenses, the platform's openness, improved overall performance, efficient operations, scalability, flexibility, and short production cycles [11].

The IPv6 address architecture provides security services, including extension headers [12]. But some attacks, such as man-in-the-middle (MITM) attacks, impersonation attacks, and replay attacks affect the IPv6 architecture [13]. In an MITM attack, an intruder is secretly involved between two communicating parties [14]. The sender or receiver does not verify the legitimacy of the source and replies to the attacker. Authentication and data confidentiality are compromised due to the MITM attack [15,16]. Authentication is a technique for the legitimate identification of someone or a device [17]. Confidentiality means that the personal data that might be dispatched with the aid of the sender is not made available or disclosed to unauthorized persons during conversations [18].

1.1. Motivation and Contribution

Although many asymmetric key-based or symmetric key-based secure systems are available to provide security services, the available secure systems follow computationally complex procedures. Therefore, computation and communication overheads are increased. The objective of this research is to propose a lightweight and certificateless client-server secure communication system that provides authentication, data confidentiality, and data integrity services with low/reduced computation and communication overheads.

The major contributions of the paper are described as follows:

1. We introduce a lightweight symmetric-session-key exchange algorithm for the client-server network model.
2. We propose an authentication scheme by using the Hash-based Message Authentication Code (HMAC) algorithm with a symmetric session key to identify the receiver and the sender.
3. We use an Advanced Encryption Standard (AES) algorithm with an initially symmetric session key of 128 bits in length to provide data confidentiality and the Secure Hash Algorithm 2 (SHA-2) with a 256-bit digest (SHA-2-256) for data integrity.
4. We exhibit the security analysis of our proposed client-server secure communication scheme.
5. We use Graphical Network Simulator-3 (GNS3) and Oracle VirtualBox manager 6.1 to design NFV-based IPv4-IPv6 virtual networks and implement our proposed security scheme on Linux-based Ubuntu 20.04 Long-Term Support (LTS) virtual machines through Java programming.
6. We compare our proposed security scheme with other security schemes with regard to computation and communication costs.
7. Finally, we measure and analyze the performance of our proposed security scheme over the NFV-based IPv4 and IPv6 virtual networks based on several parameters, such as throughput, delays, and packet loss.

1.2. Organization of the Paper

The remaining portion of the paper is structured as follows: Section 2 presents related works and highlights the weaknesses of existing studies. Section 3 describes our proposed end-to-end secure communication scheme. Section 4 presents formal and informal security

analysis. Section 5 shows the implementation, comparisons of the results, and performance evaluation. Eventually, Section 6 concludes the paper.

2. Related Work

Authentication and data confidentiality services with asymmetric keys are vulnerable in the single form [19]. Asymmetric key-based security systems provide data confidentiality and authentication services together. So they increase the computation cost and communication time. Therefore, they are considered too heavyweight for smart devices. Rivest–Shamir–Adleman (RSA) and Elliptic Curve Cryptography (ECC) are the two most famous asymmetric key generation algorithms. Researchers proposed mutual authentication schemes using the RSA algorithm in [20–23]. The RSA algorithm generates key pairs slowly as compared to ECC [24]. So RSA-based security systems are not suitable for lightweight devices.

In [25,26], the researchers introduced mutual authentication schemes using ECC. The schemes adopted complex procedures for authentication. So this also increases computation and communication costs. ECC is threatened by numerous attacks, including twist-security attacks, easy-timing attacks, and facet-channel attacks [27]. The major drawback of ECC is that it is not easily implemented. So these security models are not feasible for smart devices.

In [28], the researchers proposed an end-to-end secure communication framework for smart devices. The proposed framework provides authentication and encryption services by using a pre-shared key. This framework is not suitable for smart devices because the framework adopted computationally complex procedures for authentication. The proposed framework did not exhibit formal or informal proof against known attacks such as MITM attacks, impersonation attacks, or replay attacks.

In [29], the researchers added a comfortable remote consumer authentication scheme for the smart home environment. The stated authentication scheme uses multiple hashing and XOR functions during the whole process. The said scheme provides security against replay attacks, eavesdropping attacks, smartphone device loss attacks, impersonation attacks, session key guessing attacks, and MITM attacks. The authors proved the security of the proposed scheme through Burrows–Abadi–Needham (BAN) logic and the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool. The said authentication scheme increases computation and communication overhead for smart devices. Therefore, the said authentication scheme is not recommended for smart home environments.

In [30], the researchers proposed a hash-based authentication and key exchange protocol for the Industrial Internet of Things (IIoT). The proposed scheme achieves mutual authentication and key establishment whilst ensuring the anonymity of identities. The said scheme is stated as lightweight and effective due to limited computational overhead and resistance against many significant attacks. Although the scheme is declared robust through the AVISPA tool, the said scheme does not ensure privacy. However, the stated scheme does not employ any ciphering version. The said scheme adopted a very complicated key agreement and mutual authentication method that increased the computation time as well as the communication time. In this scheme, the large size of messages exchanged during conversations overburdens the overall scheme. Thus, the stated scheme is not considered to be lightweight.

In [31], the researchers introduced a lightweight key exchange and physically secure mutual authentication protocol for Industrial Wireless Sensor Networks (IWSNs). The proposed protocol uses multiple iterations of one-way hash functions, physically unclonable functions (PUFs), and bitwise exclusive operations during the key exchange and authentication process. The researchers evaluated their proposed security protocol on the real-or-random (ROR) model. The proposed security model provides security services against several known attacks, but the security model adopted a complicated procedure

for authentication. It consumed a large amount of time. So it is not recommended for wireless devices.

In [32], the researchers proposed a symmetric key agreement protocol and lightweight XOR-based authentication scheme named A2P for vehicle-to-smart-grid networks. The proposed scheme reduced the load on the grid. The proposed scheme provides security services against DoS attacks, replay attacks, and MITM attacks. The researchers proved their proposed authentication scheme through informal and formal security analyses through the AVISPA tool. This scheme also adopted complex mathematical techniques during the key agreement and pseudonym update phase. Therefore, this scheme increased the computational time of the key agreement phase.

Most of the existing security systems consist of computationally complex algorithms, and they require extra time to process, while some systems do not offer the claimed robust protection against safety assaults. A comparative analysis based on computation and communication costs between existing security models is shown in Table 1. After assessing the existing research, we recommend and enforce a strong, lightweight (in terms of computation and conversation costs), and end-to-end secure communication model. Our proposed secure communication scheme provides mutual authentication and data confidentiality services by using a pre-shared key. IPsec also provides these security services [33] on the network layer, with high costs in terms of computation and communication.

Table 1. Comparative analysis of existing models.

Reference	Computation Time (ms)	Communication Cost (bits)
[25]	10.524	2016
[26]	7.666	2624
[29]	0.424	1664
[30]	1.658	2400
[31]	2.614	3584
[32]	0.414	4064

3. Proposed Secure Communication Model

The primary goal of our proposed secure communication scheme is to provide authentication and data confidentiality services in the client–server network model so that the MITM’s interception will fail. After the client–server connection is established, our proposed security scheme performs. The process state diagram of our proposed client–server security model is shown in Figure 1. Our proposed security scheme consists of several phases. All of the phases are discussed in detail in [34]. Table 2 describes the notations and their descriptions.

Table 2. Notations used in our scheme.

Notation	Description
C	User as a Client
S	Server
ID_C	Unique identity of the client
PSW_C	Password of the client
N_1, N_2	Two large random numbers
N_C, N_S	Secret numbers generated by the client and server
R_C, R_S	Final results of the client and the server
K_S	Symmetric key
\oplus	XOR operation
\parallel	Concatenation operation
Hash (.)	One-way hash function
$HMAC_C$	Hash code generated by the client
$HMAC_S$	Hash code generated by the server
$E_K(.)$	Encryption with symmetric key

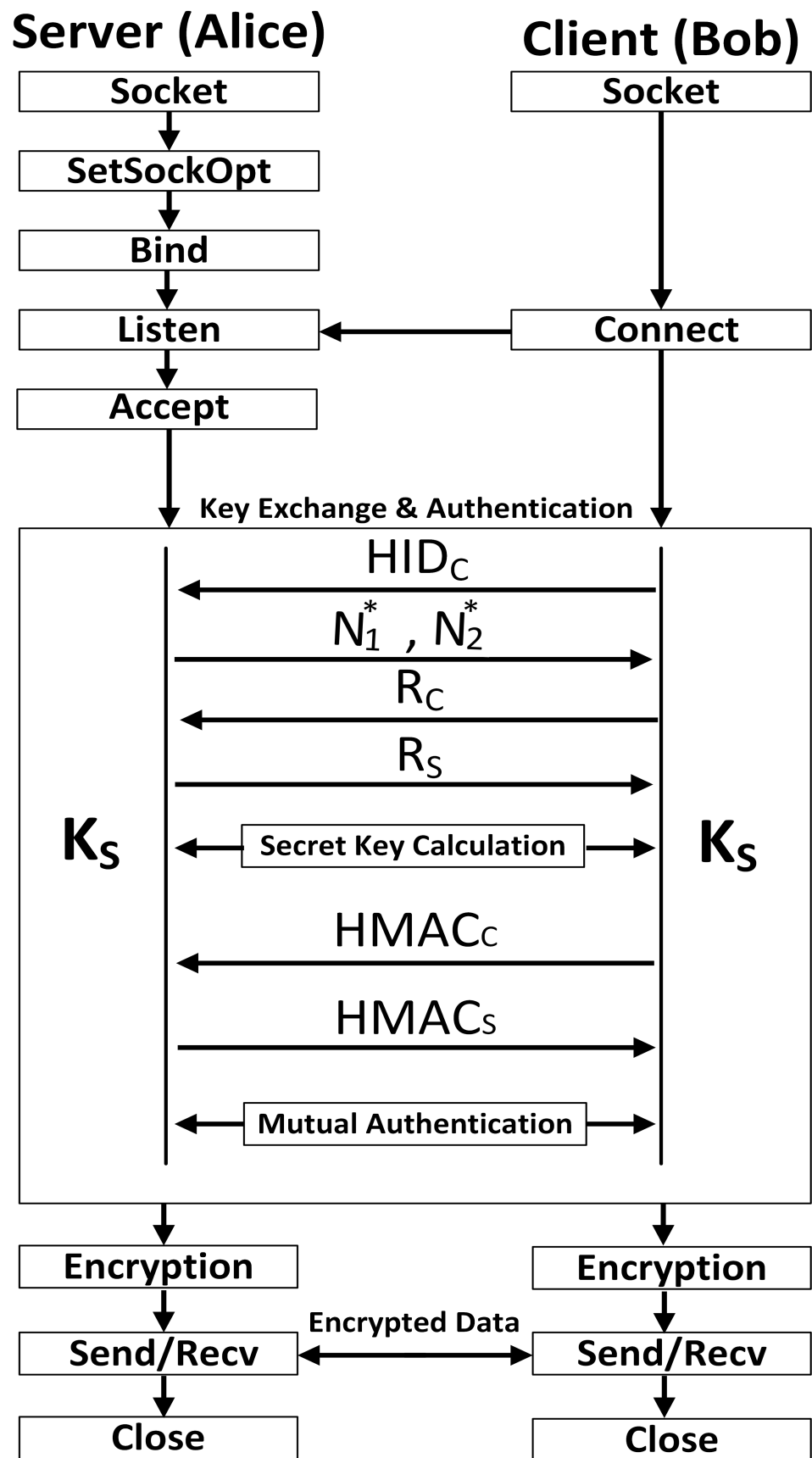


Figure 1. Proposed secure communication scheme.

3.1. User Registration Phase

During the user registration phase, the user's credentials, such as hash-based identity, hash-based password, name, address, phone number, and email address, are saved on the server's database. The user as a client picks an identity as ID_C and a password as PSW_C . The client calculates the hash-based identity HID_C and hash-based password $HPSW_C$ as described in Equation (1) and Equation (2), respectively.

$$HID_C = \text{hash}(ID_C) \quad (1)$$

$$HPSW_C = \text{hash}(ID_C \parallel PSW_C) \quad (2)$$

3.2. Login Phase

The user as a client C sends his hash-based identity HID_C to the server S for verification. The server locates the hash-based password from the database against a given identity. If the given identity is not located, then the server terminates the connection. If the server locates the hash-based password against the given identity, then the server generates two large random numbers N_1 and N_2 . The server calculates $N_1^* = (N_1 \oplus HPSW_C)$, $N_2^* = (N_2 \oplus HPSW_C)$ and sends N_1^*, N_2^* to the user.

3.3. Session Key Exchange Phase

The next step is to exchange the secret key between the server and the client. A bigger key length reduces the threat of brute-force attacks [35]. We proposed a robust and lightweight symmetric key exchange algorithm [36]. Initially, the key size is 128 bits. However, the proposed key exchange algorithm supports larger key sizes such as 1024 bits or more without modification. A bigger key size also reduces post-quantum resistance [37].

3.4. Authentication Phase

The next step is to authenticate the sender and the receiver. We use the HMAC algorithm, which uses a hashing function along with a symmetric session key K_S that is exchanged between the sender and receiver for authentication [38]. The HMAC size depends on the hashing algorithm [39]. We use the (SHA-2-256) algorithm with a symmetric session key, concatenation with a random number generated by the device, and the IP address of the device for mutual authentication as described in Equation (3) and Equation (4), respectively. Figure 2 shows the key exchange and authentication processes.

$$HMAC_C = \text{SHA} - 256 (IP\ Address_C \parallel N_C, K_S) \quad (3)$$

$$HMAC_S = \text{SHA} - 256 (IP\ Address_S \parallel N_S, K_S) \quad (4)$$

3.5. Secret Data Transmission

After the mutual authentication process has been done, the next process is to send or receive data secretly. For encryption and decryption, we use the AES algorithm with the symmetric session key as described in Equation (5) and Equation (6), respectively.

$$M_{Enc} = \text{AES} (Plain\ Text\ Message, K_S) \quad (5)$$

$$M_{Dec} = \text{AES} (M_{Enc}, K_S) \quad (6)$$

The working of AES is fast [40]. AES provides services in different block cipher modes (at least 10) of operation [41]. The comparative results between the five different modes indicate that the electronic code block (ECB) consumes less execution time in encryption and decryption processes based on the software and hardware specifications that were used during experiments [42].

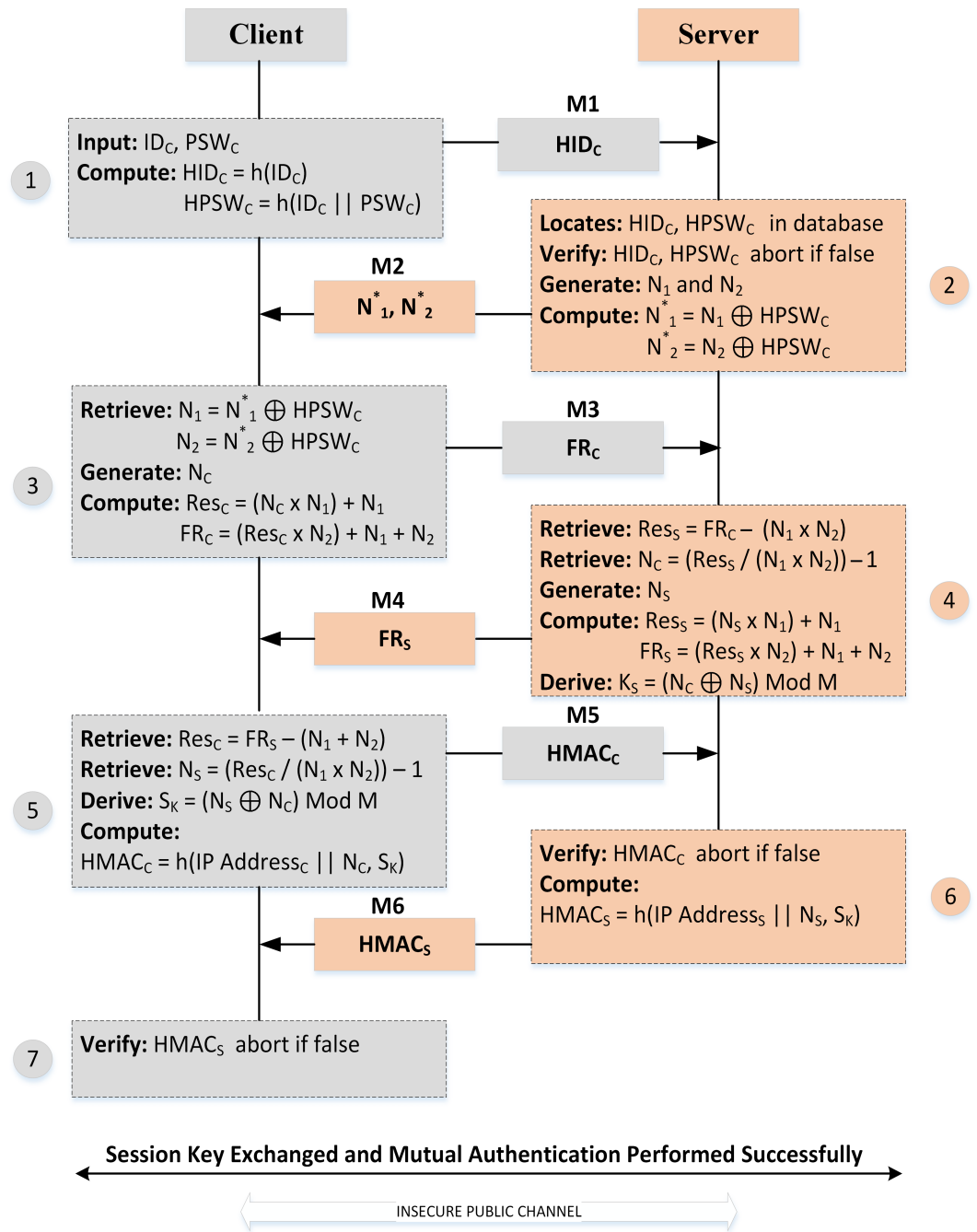


Figure 2. Key exchange and authentication process.

3.6. Data Integrity

To ensure data integrity, remote users and the server send or receive hash values along with the data. The hash value is calculated by using any hashing algorithm, such as (SHA-2-256), along with the key, as shown in Equation (7). Hashing is a one-way process [43]. On the receiving end, the receiver calculates the hash value and compares it to what was received. If both hash values are the same, it means the data were not altered. If both hash values are not the same, then it means that the data were altered.

$$Hash = SHA - 256 (TextMessage, Key) \quad (7)$$

4. Security Analysis

We show the robustness of our proposed end-to-end secure client–server communication scheme against MITM attacks and impersonation attacks through informal and formal security analyses.

4.1. Security against Impersonation Attacks

In our proposed system, the client sends a hash-based identity as HID_C instead of the identity in clear text. On the server side, the server locates HID_C in the database. If the given identity is incorrect, then the server terminates the connection immediately. Multiple wrong attempts block the identity temporarily. The size of the hash-based identity is large. So brute-force and spoofing attacks fail.

4.2. Security against Replay Attacks

Using the random nonce concept, our proposed security scheme provides safety against replay attacks. The random nonce will be changed on every newly established connection between the client and the server.

4.3. Safety against MITM Attacks

The client and the server authenticated each other using the combination of a pre-shared key, the IP address of the party, and a random number generated by the party. The values of N_1 , N_2 , N_C , and N_S are hidden to adversaries. Similarly, the pre-shared symmetric key is also hidden from adversaries. The pre-shared key's value changes every time. The client and the server send only R_C and R_S to each other. The attacker only knows N_1^* , N_2^* , R_C , R_S , $HMAC_C$, and $HMAC_S$, as shown in Table 3. If the adversary sends a fake hash value, it is detected easily during the authentication process, as shown in Figure 3.

Table 3. MITM.

Client	Intruder	Server
N_1, N_2	N_1^*, N_2^*	N_1, N_2
N_C, N_S	-	N_C, N_S
R_C, R_S	R_C, R_S	R_C, R_S
K_S	-	K_S
$HMAC_C, HMAC_S$	$HMAC_C, HMAC_S$	$HMAC_C, HMAC_S$

4.4. Safety against Side-Channel Attacks

Side-channel attacks (SCAs) are timing attacks. In SCAs, the attacker gains information from the physical implementation of a computer system rather than exploiting software vulnerabilities [44]. The clock randomization technique is an old method of countermeasure against side-channel attacks [45]. Our proposed security scheme is safe against SCAs because our key exchange algorithm shows constant time behavior [36]. In addition, the SHA module is also safe against SCAs because we use SHA in HMAC along with a large symmetric session key.

4.5. Safety against Quantum Computers

Post-quantum cryptography (PQC) is another fundamental security aspect of current scientific phenomena [46]. Authentication, hashing, and cryptography algorithms provide security against post-quantum resistance by having large keys [34]. Our proposed security scheme is safe against post-quantum resistance because our key exchange algorithm can exchange a large key size such as 1024 bits or more without modification.

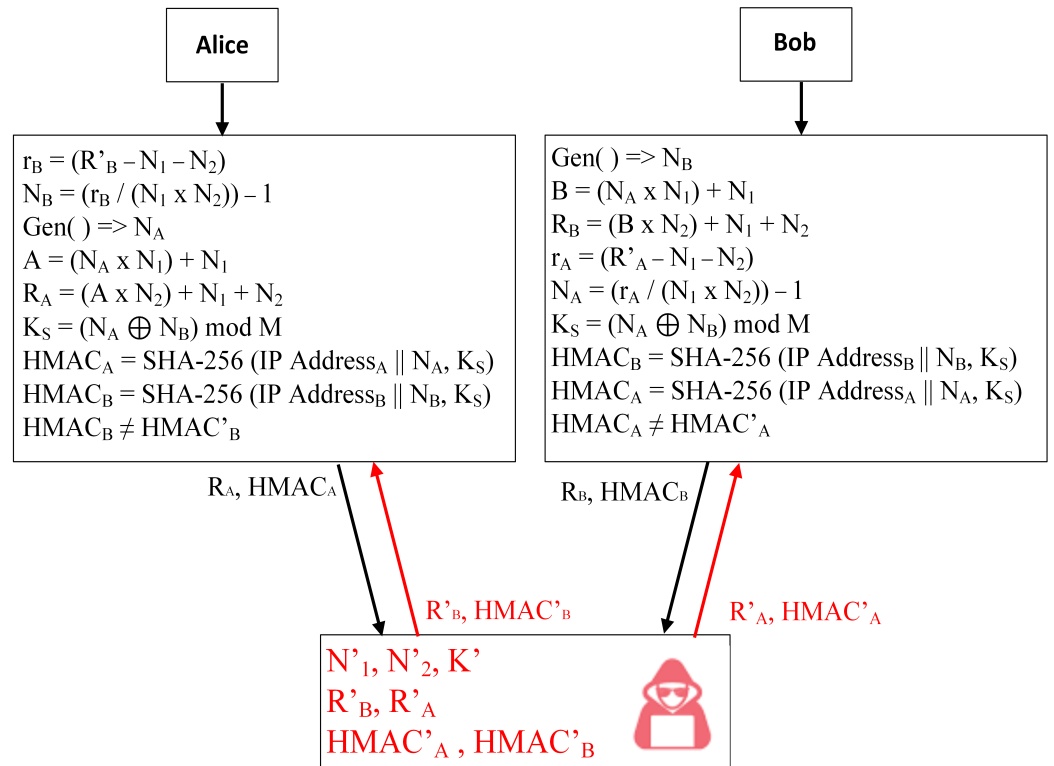


Figure 3. MITM attack detection.

4.6. Formal Security Analysis through AVISPA

We used the AVISPA tool to examine the robustness of our proposed security scheme against MITM attacks and impersonation attacks. AVISPA is a dependable and open-source tool. It is used to check the safety of several types of security protocols and schemes that contain messages exchanged between two or more nodes [47]. AVISPA adopts the Dolev–Yao (DY) adversary model in which the communication channels that are used between nodes are compromised and liable to all forms of attacks [48].

4.6.1. Server Role

The server issues the IDs and password and sends two randomly generated large numbers N_1 and N_2 . Symmetric key exchange and authentication processes are also performed by the server. The AVISPA code and other relevant configuration files are available on GitHub [49].

4.6.2. Client Role

The client sends a request to the server for login, key exchange, and mutual authentication.

4.6.3. Results through AVISPA

The robustness of our designed cozy communication version in opposition to a couple of intense assaults has been validated using the OFMC and CL-AtSe on the backend and became safe, as shown in Figure 4.

<pre> % OFMC SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS PROTOCOL /home/span/span/testsuite/results/myScheme.if GOAL as specified BACKEND OFMC COMMENTS STATISTICS parseTime: 0.00s searchTime: 0.53s visitedNodes: 466 nodes depth: 4 plies </pre>	<pre> %AtSe SUMMARY SAFE DETAILS BOUNDED_NUMBER_OF_SESSIONS TYPED_MODEL PROTOCOL /home/span/span/testsuite/results/myScheme.if GOAL As Specified BACKEND CL-AtSe STATISTICS Analysed : 48 states Reachable : 37 states Translation: 0.04 seconds Computation: 0.00 seconds </pre>
---	---

Figure 4. Results through OFMC and AtSe.

5. Implementation over NFV-Based Virtual Networks and Performance Analysis

We implement our proposed secure end-to-end communication scheme on Linux-based Lubuntu virtual machines by using socket programming in Java with built-in Java packages such as `java.io.*` and `java.net.*` integrated on the GNS3 simulator. The system specifications are: HP EliteBook 840 G3 series, Intel(R) Core i5-6300U 2.4 GHz processor, 16 GB DDR4 RAM, 3 MB cache memory, and 64-bit Windows 10 Professional operating system.

5.1. NFV

NFV is a rising technology that was introduced recently and is quickly becoming a part of the Internet due to its many benefits. NFV is changing the shape of networks. NFV is used to convert networking functions from committed hardware appliances to primarily software-based packages [50]. NFV provides network functionality and services through one or more VMs that may run on different high-volume servers and software. For example, a virtual firewall could be deployed on a high-volume server to protect the network without installing and mounting a physically dedicated security device. Similarly, many other examples of NFV include virtualized routers, intrusion detection/prevention devices, load balancers, WAN accelerators, and session border controllers [51]. It is used to design, deploy, and manage network services with lower cost and energy consumption by decoupling proprietary physical network equipment [10].

5.2. Experimental Setup

The experimental design is shown in Figure 5. Our virtual network consists of 6 virtual routers, 2 virtual layer-2 switches, and 3 virtual hosts. The virtual routers are supported by IPv4 and IPv6 and are connected with Gigabit Ethernet as a mesh topology. The outline of the devices that were used in the experiments is shown in Table 4. A Cisco IOS XRv router is installed on a virtual machine through VirtualBox and is integrated with the GNS3 simulator. The virtual machine consists of 4 GB RAM and 2 processors. It reduces the equipment cost. PC-1 acts as a server, and PC-3 acts as a client, while PC-2 acts as an intruder. The intruder controls the communication network fully. Both virtual machines run a Linux-based Lubuntu 20.04 LTS operating system installed as a VM through Oracle VM VirtualBox manager and integrated with the GNS3 simulator. Lubuntu is a lightweight operating system [52]. The Open Shortest Path First (OSPF) routing protocol is configured for both the IPv4 and IPv6 networks for dynamic routing.

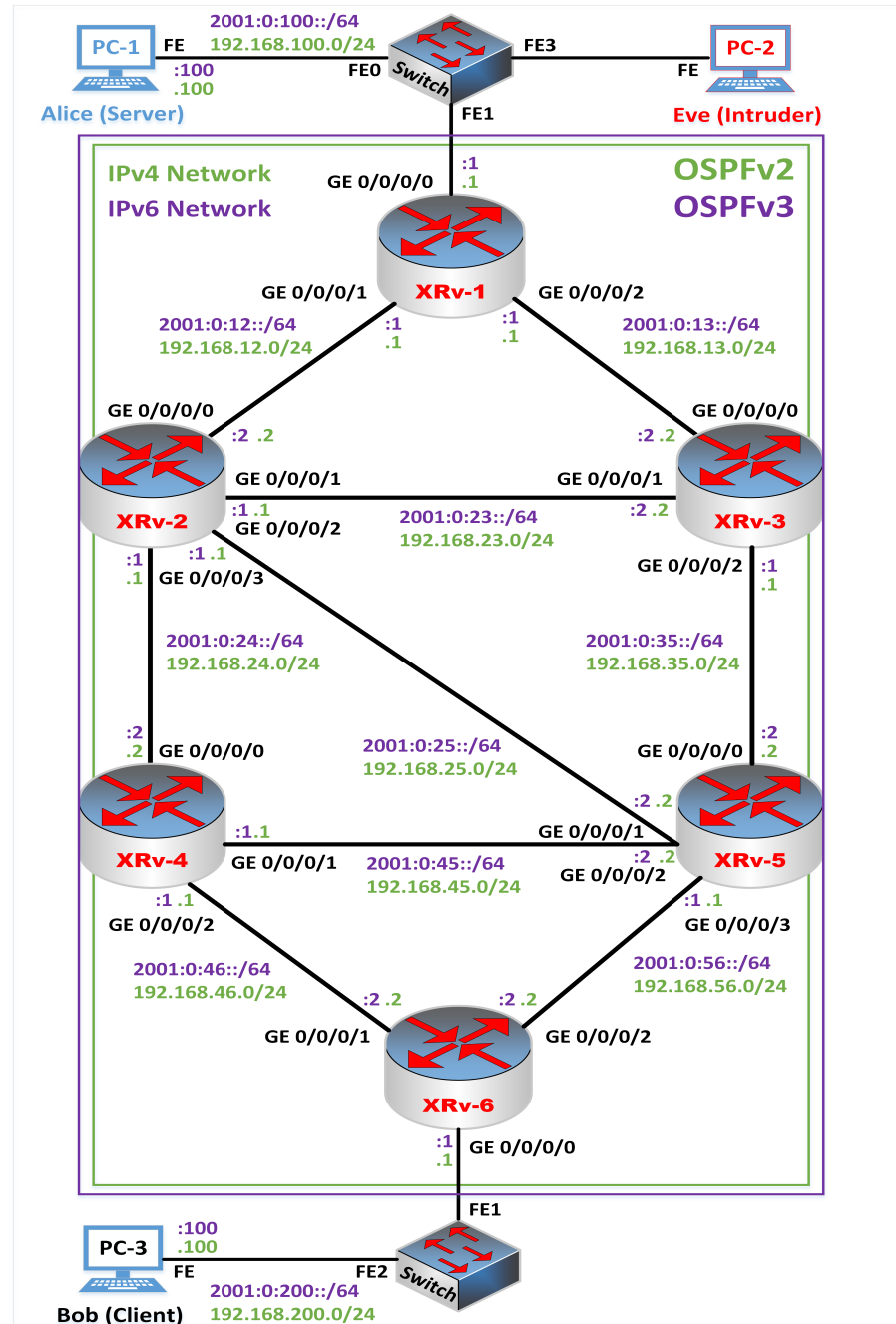


Figure 5. Experimental setup of NFV-based IPv4-IPv6 virtual networks.

Table 4. Devices and their descriptions.

Device	Description
vRouter	Cisco IOS XR virtual router, IOS iosxrv-k9-demo-6.0.0; total = 6
vSwitch	GNS3-based FastEthernet virtual switch with trunk ports; total = 2
vPC	Linux-based Lubuntu 20.04 LTS virtual machine; total = 2

Socket programming is a technique for connecting two or more nodes on a network to communicate with each other [53]. The virtual machine PC-1 runs the server.java program, while PC-3 runs the client.java program for client–server communication. Suppose the server sent $N_1 = 79$ and $N_2 = 95$ to the client. The client picked $N_C = 60$, and the server

picked $N_S = 35$. $M = 128$. The pre-shared session key $K_S = 31$. The binary value of the pre-shared key is "11111". An identical system is carried out for bigger symmetric keys. The key exchange, mutual authentication processes, and encryption or decryption between the client and the server on the IPv6 network are shown in Figure 6 and Figure 7, respectively.

```

Alice (Server)

Hi, I am Server
Client: Hello Server
Send me your ID
Client: Bob
Send me your password
Client: * * * * *
You are verified
Send me your Results
Client: 457979
270354
Client: Key has been Shared
Key has been Shared
Send me your HMAC
Client: afab7e15d203d80dcd50bb3110891916dd5ef91924c594787afc45aad4d009e1
03eb7063156b8b9356f9ef87da98a5230d236fa6d5686cedd94c8d19ce887b5e
You have recognized successfully
Client: You have recognized successfully
How are you?
Client: wsvnL0s9+hDhc1dpjd6VKA==
Client: (I am fine)
I am terminating Connection
Client: a2YuN9SWZQaEb9Szqx0hLw==
Client: (ok)
Bye

```

Figure 6. Server output.

```

Bob (Client)

Hi, I am Client
Hello Server
Server: Send me your ID
Bob
Server: Send me your password
* * * * *
Server: You are verified
Server: Send me your Results
457979
Server: 270354
Key has been Shared
Server: Key has been Shared
Server: Send me your HMAC
afab7e15d203d80dcd50bb3110891916dd5ef91924c594787afc45aad4d009e1
Server: 03eb7063156b8b9356f9ef87da98a5230d236fa6d5686cedd94c8d19ce887b5e
Server: You have recognized successfully
You have recognized successfully
Server: jFlhTUIcWSQEE/v21HyILw==
Server: (How are you?)
I am fine
Server: q/M99FjSXk+L8GuMR+fLSTDS2hGxzQiq56Zn6MvHz4s=
Server: (I am terminating Connection)
ok
Server: WwODa9zMI/A0u7K//CYg0Q==
Server: (Bye)
Bye

```

Figure 7. Client output.

5.3. Computation Cost Comparisons

We compare our designed secure communication scheme with existing schemes based on predicted computation cost. The computation results come after performing multiple mathematical or computing operations that are finished at some point in the key exchange and authentication process. Table 5 shows the unit times for numerous operations that were performed throughout the execution. We suppose ECC with a 32-byte key length and SHA-256 as a hash function. The predicted calculated computation value of our proposed scheme is $4T_H + 6T_{XOR} + 3T_{\parallel} + 22T_{Math}$ on both sides. Table 6 shows the comparison results for the computational costs for our designed scheme and most related schemes. The table shows that our designed secure scheme decreased the computation cost by up to 38.87% compared to the method in [29].

Table 5. Anticipated running times of various operations.

Notation	Description	Time (ms)
T_{Math}	Unit time for mathematical operation	0.006
T_{Mult}	Unit time for modular multiplication	0.038
T_H	Unit time for hash function	0.004
T_{Inv}	Unit time for modular inverse	0.456
T_{\parallel}	Unit time for concatenation operation	0.006
T_{EC}	Unit time for EC multiplication	1.266
T_{Add}	Unit time for EC addition	0.028
T_{XOR}	Unit time for XOR function	0.020
T_P	Unit time for physically unclonable function	0.430
T_{HMAC}	Unit time for HMAC	0.020
$T_{E/D}$	Unit time for symmetric encryption/decryption	0.147

Table 6. Computation cost comparison.

Reference	Computation Cost	Time (ms)
[25]	$19T_H + 14T_{XOR} + 57T_{\parallel} + 8T_{EC} + 3T_{Math}$	10.524
[26]	$14T_H + 5T_{XOR} + 5T_{\parallel} + 6T_{EC} + 4T_{Add}$	7.666
[29]	$12T_H + 8T_{XOR} + 36T_{\parallel}$	0.424
[30]	$32T_H + 20T_{XOR} + 87T_{\parallel} + 4T_{Mult} + 1T_{Inv}$	1.658
[31]	$22T_H + 8T_{XOR} + 36T_{\parallel} + 5T_P$	2.614
[32]	$14T_H + 2T_{XOR} + 53T_{\parallel}$	0.414
Our Proposed	$4T_H + 6T_{XOR} + 3T_{\parallel} + 22T_{Math}$	0.286

5.4. Communication Cost Comparisons

We compared our designed secure communication scheme with existing schemes based on transmitted messages on both sides. To make a reasonable comparison, we defined the lengths of different messages' variable timestamps, random numbers, IDs, hash values, EC points, and encryption/decryption as 32 bits, 128 bits, 128 bits, 256 bits, 320 bits, and 512 bits, respectively. The anticipated overall calculated conversation value of our designed scheme is 1280 bits (160 bytes) for six messages that cross both facets. Table 7 displays the comparative results for our proposed safety scheme and other current models. The table shows that our designed secure system decreased communication cost by up to 26.08% compared to the method in [29].

Table 7. Communication cost comparison.

Reference	Communication Cost (bits)	Messages Exchanged
[25]	$576 + 544 + 384 + 512 = 2016$	4
[26]	$1312 + 1312 = 2624$	2
[29]	$512 + 384 + 384 + 384 = 1664$	4
[30]	$800 + 1184 + 1184 = 2400$	3
[31]	$256 + 512 + 768 + 896 + 384 + 768 = 3584$	6
[32]	$1024 + 896 + 800 + 800 + 544 = 4064$	5
Our Proposed	$256 + 256 + 128 + 128 + 256 + 256 = 1280$	6

5.5. Comparison with IPsec

IPsec is a protocol suite that provides security services such as key exchange, authentication, data confidentiality, and data integrity on the network layer of the OSI model using a combination of different protocols. IPsec works in two modes. One is called “transport mode”, while the second is called “tunnel mode”. IPsec encapsulates the data with additional headers such as Encapsulating Security Payload (ESP), Authentication Header (AH), and Internet Key Exchange (IKE) [54]. It increases overhead for resource-constrained IoT-enabled smart devices. IPsec provides authentication with digital certificates by using the asymmetric key. This method consumes extra time and is not suitable for resource-constrained devices.

In IPsec, the DH key exchange algorithm is used for symmetric key exchange as a common practice. The DH key exchange algorithm is not self-authenticated and is vulnerable to MITM attacks [36]. The enhanced version of DH increases the key exchange time because the complexity behavior of the DH is polynomial. Our proposed secure system is most applicable in smart homes and smart healthcare systems where small IoT-enabled devices and sensors are used.

5.6. Experimental Results over IPv4-IPv6

Data were captured through the network commands during experiments. All experiments were done several times (5–7) at alternative times, then we picked the mean of the results.

5.6.1. Connectivity and Traffic Path

The “ping” command is used to check the connectivity of the devices on the network. The ping command sends an Internet Control Message Protocol (ICMP) echo request to the receiver, and the receiver sends the echo reply to the sender over the network [55]. If the sender receives a response from the receiver, then this ensures that connection has been established. We tested connectivity from PC-1 to PC-3 over the NFV-based IPv4-IPv6 networks by using the ping command. Similarly, the “trace” command is used to check the traffic path between the sender and the receiver over the network. The path shows the total number of hops from the source to the destination with the time in milliseconds for each node. Most of the time, multiple paths exist between the sender and receiver. The routing protocol selects the best path if various paths exist. In our experimental setup, multiple paths are available from PC-1 to PC-3. Using the trace command, we discovered the path from PC-1 to PC-3 over the NFV-based IPv4-IPv6 networks. The route XRv-1 → XRv-3 → XRv-5 → XRv-6 was followed. Figure 8 shows the connectivity and traffic path from PC-1 to PC-3.

```

PC-1> ping 192.168.200.100
64 bytes from 192.168.200.100 icmp_seq=1 ttl=60 time=19.731 ms
64 bytes from 192.168.200.100 icmp_seq=2 ttl=60 time=10.725 ms
64 bytes from 192.168.200.100 icmp_seq=3 ttl=60 time=9.784 ms
64 bytes from 192.168.200.100 icmp_seq=4 ttl=60 time=12.773 ms
64 bytes from 192.168.200.100 icmp_seq=5 ttl=60 time=9.743 ms

PC-1> trace 192.168.200.100
trace to 192.168.200.100, 8 hops max, press Ctrl+C to stop
 1  192.168.100.1  0.978 ms  1.951 ms  0.976 ms
 2  192.168.13.2  7.808 ms  2.937 ms  2.930 ms
 3  192.168.35.2  8.783 ms  4.878 ms  4.880 ms
 4  192.168.56.2  9.761 ms  7.809 ms  6.833 ms
 5  *192.168.200.100  9.782 ms  (ICMP type:3, code:3, Destination port unreachable)

PC-1> ping 2001:0:200::100

64 bytes from 2001:0:200::100 icmp6_seq=1 ttl=56 time=27.641 ms
64 bytes from 2001:0:200::100 icmp6_seq=2 ttl=56 time=14.690 ms
64 bytes from 2001:0:200::100 icmp6_seq=3 ttl=56 time=13.663 ms
64 bytes from 2001:0:200::100 icmp6_seq=4 ttl=56 time=16.688 ms
64 bytes from 2001:0:200::100 icmp6_seq=5 ttl=56 time=12.667 ms

PC-1> trace 2001:0:200::100

trace to 2001:0:200::100, 64 hops max
 1  2001:0:100::1  1.953 ms  1.953 ms  1.950 ms
 2  2001:0:13::2  11.713 ms  5.855 ms  5.857 ms
 3  2001:0:35::2  16.594 ms  9.760 ms  8.783 ms
 4  2001:0:56::2  20.494 ms  13.664 ms  10.737 ms
 5  2001:0:200::100  14.641 ms  14.638 ms  14.643 ms

```

Figure 8. Connectivity and traffic path.

5.6.2. Round Trip Time

We measured the average round trip times (RTTs) of different sizes of messages that passed over the hybrid IPv4-IPv6 NFV-based virtual networks during communication by using the simple ping command. The RTT is the full time it takes for a packet to be dispatched plus the amount of time it takes for an acknowledgment that the packet was received [56]. It is based on queuing delays in routers and processing at the end system. More put-off and heavy congestion will cause the packet to drop [57].

Table 8 shows the total time, minimum, maximum, average, standard deviation (SD), and coefficient of variation (CV) of the RTT from PC-1 to PC-3 with different sizes of packets passed over NFV-based IPv4 and IPv6 virtual networks. The experimental results show that the performance of IPv6 is better than the performance of IPv4 based on the CV.

Table 8. RTT over NFV-based IPv4 and IPv6 networks.

Packet Size	Protocol	Packets Sent	Time (ms)	Min (ms)	Max (ms)	Mean (ms)	SD (ms)	CV (ms)
64 Bytes	IPv4	15	14,026	9.2	19.3	10.3	2.54	24.19
	IPv6	15	14,021	12.7	27.6	15.0	3.60	23.86
32 KBytes	IPv4	15	14,015	38.1	58.3	41.9	5.66	13.42
	IPv6	15	14,014	132.8	173.0	143.6	10.42	7.25
64 KBytes	IPv4	15	14,018	61.8	135.4	85.0	20.21	24.23
	IPv6	15	14,144	244.8	329.9	261.4	21.28	7.84

5.6.3. Packet Loss

Packet loss is when any packet does not reach its destination for any reason. Packet loss takes place due to errors in transmission statistics, buffer overflow, or congestion [58]. Various sizes of the 100 packets of ICMP were sent to calculate the packet loss ratio.

Table 9 shows the packet loss ratios for different sizes of packets from PC-1 to PC-3 passed over both IPv4 and IPv6 NFV-based virtual networks. The results show that packet loss for smaller packets is zero, while for 64 KByte packets, the packet loss ratio in IPv6 is comparatively high as compared to IPv4. For fixed window sizes, when multiple heavy sizes of packets arrive, then due to queuing delays, processing delays, and propagation delays, packets are lost.

The successful execution, robustness verification, and comparative analyses proved the supremacy of the proposed secure communication scheme in contrast to the traditional approaches over large-scale NFV-based hybrid IPv4-IPv6 virtual networks.

Table 9. Packet loss ratios over NFV-based IPv4 and IPv6 networks.

Packet Size	Protocol	Packets Sent	Time (ms)	Packets Lost	Loss Ratio
64 Bytes	IPv4	100	99,127	0	0.0%
	IPv6	100	99,125	0	0.0%
32 KBytes	IPv4	100	99,149	0	0.0%
	IPv6	100	99,135	0	0.0%
64 KBytes	IPv4	100	99,261	6	6.0%
	IPv6	100	99,509	29	29.0%

6. Conclusions

The computing and telecommunication industries are moving towards the IPv6 address architecture rapidly all over the world; IPv6 supports virtualization architectures such as NFV. Some attacks, such as MITM attacks or impersonation attacks, affect the IPv6 address architecture. We proposed and implemented a lightweight end-to-end secure communication scheme by using Java programming on Ubuntu 20.04 LTS virtual machines connected to NFV-based IPv4 and IPv6 virtual networks to provide security against known attacks. The robustness of the proposed end-to-end secure communication scheme was proven. We compared our proposed security scheme with existing security models in terms of communication and computation costs. The comparative results showed that our designed security scheme is lightweight in terms of communication and computation costs. Our proposed security scheme reduced the computation cost by 38.87% and the communication cost by 26.08% related to existing models discussed in the literature review. In addition, we evaluated the performance of our proposed security scheme over large-scale NFV-based hybrid IPv4-IPv6 virtual networks with different sizes of packets based on several parameters such as end-to-end delay and packet loss. Performance evaluation over an NFV-based hybrid IPv4-IPv6 virtual network indicated that the overall performance of IPv6 is better than the performance of IPv4 with regard to most of the parameters. In

the future, we will implement our proposed security scheme integrated with blockchain to SDN with cloud computing and determine the performance.

Author Contributions: Methodology, test-bed, results, formal analysis, and writing—original draft preparation, Z.A.; validation, conceptualization, and writing—review and editing, A.S.; conceptualization, visualization, analysis, results verification, proofreading, and funding acquisition, M.I. All authors have read and agreed to the published version of the manuscript.

Funding: This article is derived from a research grant funded by the Research, Development, and Innovation Authority (RDIA), Saudi Arabia, with grant number (13354-psu-2023-PSNU-R-3-1-EI-).

Data Availability Statement: Data are contained within the article.

Acknowledgments: The authors would like to acknowledge the support of Prince Sultan University for paying the article processing charge (APC) of this publication. The authors would like to thank Prince Sultan University for their support. The authors also thank the anonymous reviewers and the editor for their valuable feedback on the paper, which helped the authors to improve its quality and presentation.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Rahman, A.; Islam, J.; Kundu, D.; Karim, R.; Rahman, Z.; Band, S.S.; Sookhak, M.; Tiwari, P.; Kumar, N. Impacts of blockchain in software-defined Internet of Things ecosystem with Network Function Virtualization for smart applications: Present perspectives and future directions. *Int. J. Commun. Syst.* **2023**, *36*, e5429.
- Deering, S.; Hinden, R. *Internet Protocol, Version 6 (IPv6) Specification*; RFC 8200; 2017. Available online: <https://datatracker.ietf.org/doc/html/rfc8200> (accessed on 23 June 2024).
- Rwibasira, M.; Suchithra, R. Blockchain-based security for internet of everything. In *Blockchain-Based Systems for the Modern Energy Grid*; Elsevier: Amsterdam, The Netherlands, 2023; pp. 101–114.
- Sharma, J.; Mehra, P.S. Secure communication in IOT-based UAV networks: A systematic survey. *Internet Things* **2023**, *23*, 100883.
- Arents, J.; Greitans, M. Smart industrial robot control trends, challenges and opportunities within manufacturing. *Appl. Sci.* **2022**, *12*, 937.
- Luo, H.; Wu, Y.; Sun, G.; Yu, H.; Guizani, M. ESCM: An efficient and secure communication mechanism for UAV networks. *IEEE Trans. Netw. Serv. Manag.* **2024**, *21*, 1–14.
- Rehman, A.; Haseeb, K.; Alruwaili, F.F.; Ara, A.; Saba, T. Autonomous and Intelligent Mobile Multimedia Cyber-Physical System with Secured Heterogeneous IoT Network. In *Mobile Networks and Applications*; Springer: Amsterdam, The Netherlands, 2024; pp. 1–10.
- Rao, P.M.; Deebak, B. Security and privacy issues in smart cities/industries: Technologies, applications, and challenges. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *14*, 10517–10553.
- Dawadi, B.R.; Rawat, D.B.; Joshi, S.R.; Manzoni, P.; Keitsch, M.M. Migration cost optimization for service provider legacy network migration to software-defined IPv6 network. *Int. J. Netw. Manag.* **2021**, *31*, e2145.
- Ray, P.P.; Kumar, N. SDN/NFV architectures for edge-cloud oriented IoT: A systematic review. *Comput. Commun.* **2021**, *169*, 129–153.
- Atzori, L.; Bellido, J.L.; Bolla, R.; Genovese, G.; Iera, A.; Jara, A.; Lombardo, C.; Morabito, G. SDN&NFV contribution to IoT objects virtualization. *Comput. Netw.* **2019**, *149*, 200–212.
- Gont, F.; Liu, W. *Recommendations on the Filtering of IPv6 Packets Containing IPv6 Extension Headers at Transit Routers*; RFC 9288; 2022. Available online: <https://datatracker.ietf.org/doc/rfc9288/> (accessed on 23 June 2024).
- Ashraf, Z.; Sohail, A.; Latif, S.; Hameed, A.; Yousaf, M. Challenges and Mitigation Strategies for Transition from IPv4 Network to Virtualized Next-Generation IPv6 Network. *Int. Arab J. Inform. Technol.* **2023**, *20*, 78–91.
- Shiranzaei, A.; Khan, R.Z. IPv6 security issues—A systematic review. *Next-Gener. Netw.* **2018**, *638*, pp. 41–49.
- Haseeb, K.; Saba, T.; Rehman, A.; Abbas, N.; Kim, P.W. AI-driven IoT-fog analytics interactive smart system with data protection. In *Expert Systems*; Elsevier: Amsterdam, The Netherlands, 2024; p. e13573.
- Feng, W.; Zhao, X.; Zhang, J.; Qin, Z.; Zhang, J.; He, Y. Image encryption algorithm based on plane-level image filtering and discrete logarithmic transform. *Mathematics* **2022**, *10*, 2751.
- Wang, X.; Yan, Z.; Zhang, R.; Zhang, P. Attacks and defenses in user authentication systems: A survey. *J. Netw. Comput. Appl.* **2021**, *188*, 103080.
- Zeadally, S.; Das, A.K.; Sklavos, N. Cryptographic technologies and protocol standards for Internet of Things. *Internet Things* **2021**, *14*, 100075.
- Forouzan, B.A.; Mukhopadhyay, D. *Cryptography and Network Security*; Mc Graw Hill Education (India) Private Limited: New York, NY, USA, 2015; Volume 12.

20. Arumugam, M.; Deepa, S.; Arun, G.; Sathishkumar, P.; Jeevanantham, K. Secure data sharing for mobile cloud computing using RSA. IOP Publishing: Bristol, UK, 2021; Volume 1055, p. 012108.
21. Raniyal, M.S.; Woungang, I.; Dhurandher, S.K. An RSA-based user authentication scheme for smart-homes using smart card. In Proceedings of the International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments, Vancouver, BC, Canada, 28–30 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 16–29.
22. Bagha, A.M.; Woungang, I.; Dhurandher, S.K.; Traore, I. A RSA-Biometric Based User Authentication Scheme for Smart Homes Using Smartphones. In Proceedings of the International Conference on Advanced Information Networking and Applications, Caserta, Italy 15–17 April 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 845–857.
23. Somsuk, K.; Thakong, M. Authentication system for e-certificate by using RSA's digital signature. *TELKOMNIKA (Telecommun. Comput. Electron. Control)* **2020**, *18*, 2948–2955.
24. Suárez-Albela, M.; Fernández-Caramés, T.M.; Fraga-Lamas, P.; Castedo, L. A practical performance comparison of ECC and RSA for resource-constrained IoT devices. In Proceedings of the 2018 Global Internet of Things Summit (GloTS), Bilbao, Spain, 4–7 June 2018; pp. 1–6.
25. Li, X.; Peng, J.; Niu, J.; Wu, F.; Liao, J.; Choo, K.K.R. A robust and energy efficient authentication protocol for industrial internet of things. *IEEE Internet Things J.* **2017**, *5*, 1606–1615.
26. Eftekhari, S.A.; Nikooghadam, M.; Rafighi, M. Robust session key generation protocol for social internet of vehicles with enhanced security provision. *J. Supercomput.* **2021**, *77*, 2511–2544.
27. Abarzúa, R.; Valencia, C.; López, J. Survey for performance & security problems of passive side-channel attacks countermeasures in ECC. *Cryptol. eprint Arch.* **2021**, *11*, 71–102.
28. Jan, M.A.; Zhang, W.; Usman, M.; Tan, Z.; Khan, F.; Luo, E. SmartEdge: An end-to-end encryption framework for an edge-enabled smart city application. *J. Netw. Comput. Appl.* **2019**, *137*, 1–10.
29. Fakroon, M.; Alshahrani, M.; Gebali, F.; Traore, I. Secure remote anonymous user authentication scheme for smart home environment. *Internet Things* **2020**, *9*, 100158.
30. Paliwal, S. Hash-based conditional privacy preserving authentication and key exchange protocol suitable for industrial internet of things. *IEEE Access* **2019**, *7*, 136073–136093.
31. Gope, P.; Das, A.K.; Kumar, N.; Cheng, Y. Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks. *IEEE Trans. Ind. Informatics* **2019**, *15*, 4957–4968.
32. Agilandeswari, L.; Paliwal, S.; Chandrakar, A.; Prabukumar, M. A new lightweight conditional privacy preserving authentication and key-agreement protocol in social internet of things for vehicle to smart grid networks. *Multimed. Tools Appl.* **2022**, *81*, 27683–27710.
33. Ullah, S.; Choi, J.; Oh, H. IPsec for high speed network links: Performance analysis and enhancements. *Future Gener. Comput. Syst.* **2020**, *107*, 112–125.
34. Ashraf, Z.; Sohail, A.; Yousaf, M. Lightweight and authentic symmetric session key cryptosystem for client-server mobile communication. *J. Supercomput.* **2023**, *79*, 16181–16205.
35. Verma, R.; Dhanda, N.; Nagar, V. Enhancing Security with In-Depth Analysis of Brute-Force Attack on Secure Hashing Algorithms. In Proceedings of the Trends in Electronics and Health Informatics, Tirunelveli, India 15–17 June 2020; Springer: Berlin/Heidelberg, Germany, 2022; pp. 513–522.
36. Ashraf, Z.; Sohail, A.; Yousaf, M. Robust and lightweight symmetric key exchange algorithm for next-generation IoE. *Internet Things* **2023**, *22*, 100703.
37. Chawla, D.; Mehra, P.S. A roadmap from classical cryptography to post-quantum resistant cryptography for 5G-enabled IoT: Challenges, opportunities and solutions. *Internet Things* **2023**, *24*, 100950.
38. Lawrence, T.; Li, F.; Ali, I.; Kpiebaareh, M.Y.; Haruna, C.R.; Christopher, T. An HMAC-based authentication scheme for network coding with support for error correction and rogue node identification. *J. Syst. Archit.* **2021**, *116*, 102051.
39. Kelly, S.; Frankel, S. Using *hmac-sha-256*, *hmac-sha-384*, and *hmac-sha-512 with Ipsec*; RFC 4864; 2007. Available online: <https://datatracker.ietf.org/doc/rfc4868/> (accessed on 23 June 2024).
40. Pandian, R.; Columbus, C. An Analytical approach for optimal secured data storage on cloud server for online education platform. *Geosci. Instrumentation, Methods Data Syst. Discuss.* **2022**, *2022*, 1–36.
41. Nannipieri, P.; Di Matteo, S.; Baldanzi, L.; Crocetti, L.; Zulferti, L.; Saponara, S.; Fanucci, L. VLSI design of Advanced-Features AES CryptoProcessor in the framework of the European Processor Initiative. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *30*, 177–186.
42. Almuhammad, S.; Al-Hejri, I. A comparative analysis of AES common modes of operation. In Proceedings of the 2017 IEEE 30th Canadian conference on electrical and computer engineering (CCECE), Windsor, ON, Canada, 30 April–3 May 2017; pp. 1–4.
43. Alkhonaini, M.A.; Alenizi, F.A.; Jazyah, Y.H.; Lee, S. A two-phase spatiotemporal chaos-based protocol for data integrity in IoT. *Sci. Rep.* **2024**, *14*, 8629.
44. Devi, M.; Majumder, A. Side-channel attack in Internet of Things: A survey. In *Applications of Internet of Things: Proceedings of ICCCIOT 2020*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 213–222.
45. Brisfors, M.; Moraitis, M.; Dubrova, E. Do not rely on clock randomization: A side-channel attack on a protected hardware implementation of AES. In *International Symposium on Foundations and Practice of Security*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 38–53.

46. Mavroeidis, V.; Vishi, K.; Zych, M.D.; Jøsang, A. The impact of quantum computing on present cryptography. *arXiv* **2018**, arXiv:1804.00200.
47. Vigano, L. Automated security protocol analysis with the AVISPA tool. *Electron. Notes Theor. Comput. Sci.* **2006**, *155*, 61–86.
48. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208.
49. AVISPA Code and Simulation Results. GitHub: 2023. Available online: <https://github.com/zashraf-sudo/researchpaper-6-code> (accessed on 15 May 2024).
50. Mostafavi, S.; Hakami, V.; Sanaei, M. Quality of service provisioning in network function virtualization: a survey. *Computing* **2021**, *103*, 917–991.
51. Xie, Y.; Wang, S.; Wang, B.; Xu, S.; Wang, X.; Ren, J. Online algorithm for migration aware Virtualized Network Function placing and routing in dynamic 5G networks. *Comput. Netw.* **2021**, *194*, 108115.
52. Whitt, P. Linux: The Free Alternative to Windows and macOS. In *Pro Freeware and Open Source Solutions for Business: Money-Saving Options for Small Enterprises*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 219–243.
53. Jayasekara, G. Network Security Programming (JAVA) Socket Programming With TCP & UDP: Case Study Analysis. In *Network Security Programming (JAVA) Socket Programming With TCP & UDP: Case Study Analysis (September 7, 2022)*; Elsevier: Amsterdam, The Netherlands, 2022.
54. Schwenk, J. IP Security (IPSec). In *Guide to Internet Cryptography: Security Protocols and Real-World Attack Implications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 135–190.
55. Gezer, A. The delay measurement and analysis of unreachable hosts of internet. *Int. Arab J. Inf. Technol.* **2022**, *19*, 63–71.
56. de Oliveira Filho, A.T.; Freitas, E.; do Carmo, P.R.; Sadok, D.H.; Kelner, J. An experimental investigation of round-trip time and virtualization. *Comput. Commun.* **2022**, *184*, 73–85.
57. Zhang, B.; Li, Y.; Liang, Y. *Impact of Packet Size on Performance of TCP Traffic with Small Router Buffers*; EDP Sciences: Les Ulis, France, 2017; Volume 128, p. 02023.
58. Oleiwi, S.S.; Mohammed, G.N.; Al_barazanchi, I. Mitigation of packet loss with end-to-end delay in wireless body area network applications. *Int. J. Electr. Comput. Eng.* **2022**, *12*, 460.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.