applied sciences

MDPI

*Article*

# Efficient Key Exchange Using Identity-Based Encryption in Multipath TCP Environment

**Ali Saeed Almuflih [1,*], Khushi Popat [2,3], Viral V. Kapdia [3], Mohamed Rafik Noor Mohamed Qureshi [1], Naif Almakayeel [1] and Rabia Emhamed Al Mamlook [4]**

[1] Industrial Engineering Department, King Khalid University, Abha 62529, Saudi Arabia; mrnoor@kku.edu.sa (M.R.N.M.Q.); halmakaeel@kku.edu.sa (N.A.)
[2] Department of Computer Engineering, Devang Patel Institute of Advance Technology and Research (DEPSTAR), Faculty of Technology and Engineering (FTE), Charotar University of Science & Technology (CHARUSAT), CHARUSAT Campus, Changa, Anand 388421, India; khushipatel.ce@charusat.ac.in
[3] Computer Science and Engineering Department, The Maharaja Sayajirao University of Baroda, Vadodara 390002, India; khushi.popat-cse@msubaroda.ac.in (K.P.); viral.kapadia-cse@msubaroda.ac.in (V.V.K.)
[4] Department of Aeronautical Engineering, Al Zawiya University, Al-Zawiya City P.O. Box 16418, Libya; rabiaemhamedm.almamlook@wmich.edu
* Correspondence: asalmuflih@kku.edu.sa

**Abstract:** Across the globe, wireless devices with Internet facilities such as smartphones and tablets have become essential assets for communication and entertainment alike for everyday life for millions of people, which increases the network traffic and the demand for low-latency communication networks. The fourth-generation (4G)/long-term evolution (LTE)/ fifth-generation (5G) communication technology offers higher bandwidth and low latency services, but resource utilization and resiliency cannot be achieved, as transmission control protocol (TCP) is the most common choice for most of the state-of-art applications for the transport layer. An extension of TCP—multipath TCP (MPTCP)—offers higher bandwidth, resiliency, and stable connectivity by offering bandwidth aggregation and smooth handover among multiple paths. However, MPTCP uses multiple disjointed paths for communication to offer multiple benefits. A breach in the security of one of the paths may have a negative effect on the overall performance, fault-tolerance, robustness, and quality of service (QoS). In this paper, the research focuses on how MPTCP options such as MP_CAPABLE, ADD_ADDR, etc., can be used to exploit the vulnerabilities to launch various attacks such as session hijacking, traffic diversion, etc., to compromise the availability, confidentiality, and integrity of the data and network. The probable security solutions for securing MPTCP connections are analyzed, and the secure key exchange model for MPTCP (SKEXMTCP) based on identity-based encryption (IBE) is proposed and implemented. The parameters exchanged during the initial handshake are encrypted using IBE to prevent off-path attacks by removing the requirement for key exchange before communication establishment by allowing the use of arbitrary strings as a public key for encryption. The experiments were performed with IBE and an elliptic curve cryptosystem (ECC), which show that IBE performs better, as it does not need to generate keys while applying encryption. The experimental evaluation of SKEXMTCP in terms of security and performance is carried out and compared with existing solutions.

**Keywords:** multipath TCP (MPTCP); security; ADD_ADDR attack; off-path attacks; identity-based encryption (IBE); man-in-the-middle attack; session hijacking



**Citation:** Almuflih, A.S.; Popat, K.; Kapdia, V.V.; Qureshi, M.R.N.M.; Almakayeel, N.; Mamlook, R.E.A. Efficient Key Exchange Using Identity-Based Encryption in Multipath TCP Environment. *Appl. Sci.* **2022**, *12*, 7575. https://doi.org/10.3390/app12157575

Academic Editor: Juan Francisco De Paz Santana

Received: 9 June 2022
Accepted: 23 July 2022
Published: 27 July 2022



## 1. Introduction

With the evolution of communication technologies such as 4G, 5G, and now 6G, most state-of-the-art-devices are equipped with multiple network connections (Wi-Fi, ethernet,

4G/LTE, etc.), yet to have reliable communication, the applications use transmission control protocol (TCP) [1] as a transport layer protocol, which restricts the utilization of network resources by binding the connection over a single path as shown in Figure 1 [2]. The increment in usage of internet-enabled mobile devices in daily life increases the demand for higher bandwidth, fine-grained access control, and privacy of data for the real-time Internet-of-Things (IoT) based applications in the domain of smart healthcare, smart city, smart grid, etc. [3–5].

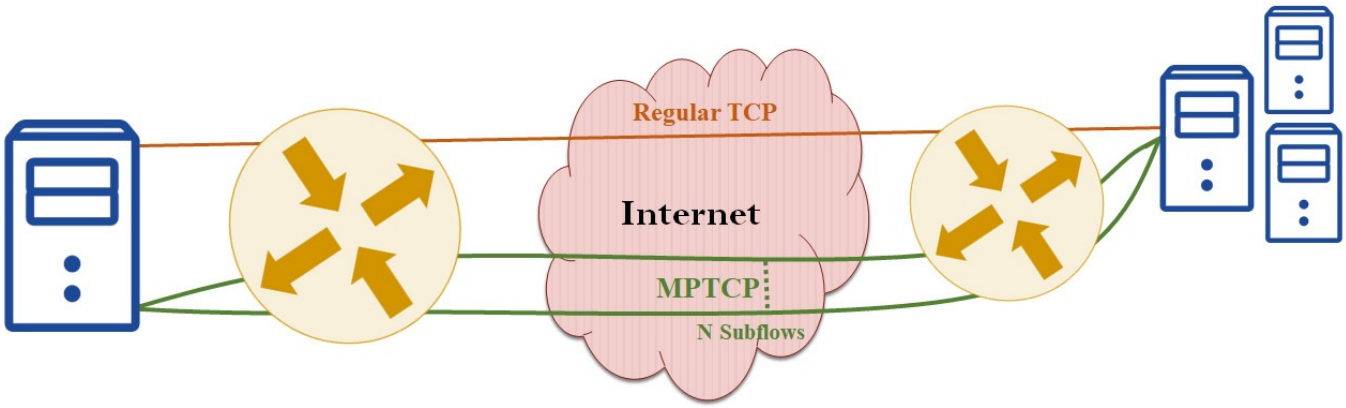


**Figure 1.** MPTCP and TCP connection.

Multipath TCP (MPTCP), the most promising extension of TCP, is an emerging transport layer protocol that overcomes the limitations of TCP and offers higher bandwidth and low-latency connections through bandwidth aggregation. It also offers resiliency by using soft handovers among paths at the time of failure. The integration of 5G and MPTCP improves the overall network performance by allowing for the use of Wi-Fi and mobile data simultaneously in mobile devices, which attracts many industries and academic fields to fulfill the incredible demand for high-speed communication in various fields.

The third-generation partnership project (3GPP), the organization that has led mobile communication standards, has demonstrated the prototype for the integration of MPTCP with 5G networks to support the usage of Wi-Fi and 5G simultaneously [6]. Many companies such as Apple, Tessars, 3GPP, Samsung, Huawei, etc. are adopting MPTCP for performance improvement in terms of bandwidth and fault tolerance by combining the usage of multiple network interfaces in mobile devices as well as in servers, datacenters, and end machines [7–11]. In order to facilitate the people of rural areas by providing access to various facilities using high-speed internet, Tessares has deployed MPTCP proxies to support hybrid access networks in different networks [12]. Smartphone companies such as Samsung, Huawei, LG, etc., have adopted MPTCP in their android devices to allow for the usage of multiple network interfaces such as Wi-Fi and long-term evolution (LTE) simultaneously to offer high-speed internet connectivity. By looking toward, the benefits of MPTCP in terms of higher throughput and fault tolerance at the time of Wi-Fi/Mobile network handover, Apple has adopted MPTCP for Apple Maps and Apple Music along with Siri since 2013 [8]. MPTCP also solves the issues related to reliability with high speed and robustness in vehicular IoT systems [13]. One of the important use cases of MPTCP is load management in data centers through multiple disjoint paths. Many architectures have been proposed for data centers, which differ from traditional systems in terms of link organization, but in all the architectures, servers are connected through multiple disjoint paths [14]. MPTCP can also be integrated with software-defined networks (SDNs) [15] to support optimized communication among governmental and commercial satellite communication systems and mobile unmanned aerial vehicle (UAV) heterogeneous networks [16].

MPTCP [17], an initiative of the Internet Engineering Task Force (IETF), is developed to offer efficient utilization of network resources, fault tolerance, and backward compatibility, which attracts various industries and academia to offer solutions in areas such as

mobile communication, vehicular networks, datacenter networks, robotics communication, software-defined networks, etc. MPTCP supports the multi-homing and multi-addressed nature of hosts for data transmission, which opens the door for security threats while availing the higher bandwidth and fault-tolerance that encourage many researchers to study the security issues of MPTCP and its solutions.

MPTCP uses the TCP header to incur a positive impact on the traditional TCP aware applications to achieve the goal of backward compatibility by using various options in TCP headers such as MP_CAPABLE, MP_JOIN, ADD_ADDR, etc., but they are vulnerable to attacks other than TCP. The ADD_ADDR option can be used to initiate a session hijacking attack by a man-in-the-middle attack, MP_JOIN can be exploited to initiate SYN flooding attacks and denial of service (DoS) attacks. Moreover, the key exchange in plaintext during the initial handshake in MPTCP invites other security threats because these keys are used for subflow authentication in the future. The focus of this article is on the security of MPTCP against ADD_ADDR vulnerability and security of keys exchanged during the initial handshake, which can be used to initiate various attacks.

Many solutions such as Transport Layer Security (TLS) [18], tcpcrypt [19], hash chained [20] and sum hash chained-based encryption [21], key exchange using SDN [22], secure and light-weight solutions [23] have been proposed and implemented by researchers to provide security against various attacks such as session hijacking, DoS attack, SYN flooding, etc., but some solutions increase overhead, which decreases overall performance of MPTCP, and in some solutions, the doors are still open for the attackers, which are available during initial key exchange. The attacker present during the initial handshake can obtain the key to initiate various attacks by bypassing the authentication at the time of adding a new subflow, advertisement of a new address, changing the priority of subflow, etc. Many researchers are working with MPTCP to resolve the issues related to congestion control, scheduling, and applicability of MPTCP in various areas such as data center networks, vehicular network, deep learning, etc., but few researchers are working with the security of MPTCP. Security is an important feature for the successful deployment of any of the protocols, which became the motivation for working in this area. Table 1 shows the available cryptographic security solutions for MPTCP with their limitations.

**Table 1.** Available cryptographic security solutions for MPTCP.

| Cryptographic Technique | References | Working with MPTCP | Limitations |
|---|---|---|---|
| Elliptic curve cryptography | [24] | The points required to plot the Elliptic curve are shared during initial four-way handshake in clear format. | One extra packet is required to share all four points to generate the keys. Vulnerable to time-shift attack. |
| Hash Chain-based Encryption | [20] | During the initial handshake, the random value will be exchanges, which will be used to generate the chain of hash by applying hash function for the authentication during addition of subflows and advertisement of new Internet Protocol (IP) Address (network interface). | Vulnerable to session hijacking using ADD_ADDR Attack. |
| Sum Chain-based encryption | [21] | Uses mathematical equation to create a chain instead of normal hash function. | Vulnerable to eavesdroppers in initial handshake attack. |
| Asymmetric Key cryptography | [18,19] | Public key cryptography can be used to avoid the key exchange during the initial handshake. Tcpcrypt and TLS uses asymmetric key cryptography. | Computational cost increases the overhead of MPTCP. Moreover, the TCP header option size is also limited. |
| Authentication using Hash based Message Authentication Code (HMAC) | [17] | The keys are exchanged during the initial handshake of MPTCP. The truncated HMAC calculated from the keys will be used to authenticate the user during ADD_ADDR and MP_JOIN. | Vulnerable to eavesdroppers in initial handshake. |

Currently, most wireless communication devices (smartphones, tablets, etc.) are equipped with mobile data and Wi-Fi interfaces, but usage is restricted to a single interface. Many times, due to connectivity issues, usage of a single interface is not sufficient in

terms of bandwidth and fault tolerance, but multiple network interfaces may resolve the issues related to speed. MPTCP allows for data transmission among end hosts over multiple TCP subflows over a single connection, which improves the transmission efficiency by sufficiently utilizing the bandwidth resources to provide TCP fairness to other TCP connections. The objective of the research is to offer secure and efficient utilization of network resources by securing the key exchanged during the initial handshake of MPTCP, which will be in favor of many industries such as data centers, wireless networks, software-defined networks, satellite communications, etc.

In order to enhance the security of MPTCP, SKEXMTCP is proposed in this paper. SKEXMTCP uses IBE [25] to encrypt the keys exchanged during the three-way handshake. The IBE, an asymmetric encryption algorithm, uses a random character sequence as a public key that lowers the key exchange overhead before communication. The same concept can be used here to exchange the session keys during the initial handshake. The public parameters can be used to encode the session keys, and the encrypted keys can be shared with another entity during the initial handshake, which can be decoded by using the private key recovered from the IBE-PKG (Private Key Generator).

Here, two modules have been proposed with SKEXMTCP: (i) Private Key Generation (SKG_SKEXMTCP); (ii) Use of Key Pair to exchange session keys during the initial handshake (MPC_SKEXMTCP). In the Key Generation (SKG_SKEXMPTCP) module, considering host Alice and host Bob are communicating with each other with IP addresses $IP_{Alice}$ and $IP_{Bob}$ using MPTCP, here $IP_{Alice}$ and $IP_{Bob}$ will be used as public parameters by PKG to generate private keys for Alice and Bob. IBE uses PKG, a third-party authority, which provides the private keys to the communicating hosts based on their identity (i.e., email id, IP address, etc.). Here, the IP address and port of the communicating host will be used as a public parameter to generate the private keys for the sender and receiver. These private keys will be used in the MPC_SKEXMTCP module to exchange the keys during the initial handshake by encrypting them using IBE. The communicating host will be authenticated by using the IP Address and port combination digitally signed by the host at the PKG. By using the IBE with MPTCP, the issue related to key distribution can be resolved, and the session key exchanged during the initial handshake can be secured by using SKEXMTCP.

The main contributions of this paper are as follows:

- MPTCP security threats are examined, and how threats can be used to launch session hijacking attacks is demonstrated.
- The existing solutions to the various security threats of MPTCP are analyzed.
- IBE is compared with the elliptic curve cryptosystem (ECC) in terms of performance and security.
- SKEXMTCP using IBE is proposed and evaluated in terms of security.

The structure of the paper is as follows: In Section 2, the background theory required to understand MPTCP and its vulnerabilities is discussed along with IBE. Section 3 covers the proposed SKEXMTCP. Section 4 analyzes the security of the proposed work and compares it with other solutions.

## 2. Background Study

### 2.1. Multipath TCP (MPTCP) and Its Security Threats

MPTCP [17,26] is the most promising transport layer protocol in the era of Industry 4.0, which offers higher resiliency and efficient congestion control by allowing for data transmission through multiple TCP subflows by selecting the least congested subflows and diverting the traffic to another flow in case of subflow failure or high congestion on a particular subflow. Improvement in bandwidth, TCP friendliness, balanced congestion control, and fault tolerance are the key characteristics of MPTCP, which make it the best option to offer traffic aggregation at the TCP level [27]. As shown in Figure 2, the MPTCP is built on top of the TCP, in which the individual subflow of MPTCP can be considered as a separate TCP connection [27–29]. Each subflow has its congestion window and round-trip

time. To distribute the data among multiple subflows efficiently, various packet scheduler options are available, such as default, round-robin, redundant, etc. but none of them are yet standardized. Many congestion control schemes are available to improve the overall performance, such as the linked increases algorithm (LIA) [30], opportunistic linked increases congestion control algorithm (OLIA) [31], weighted Vegas (wVegas) [32], etc., but LIA is the standard congestion control scheme of MPTCP [13]. MPTCP uses the TCP header option "Kind" to include the MPTCP-related data to make it compatible with TCP-aware middle-boxes and applications [17]. MPTCP uses various options in the TCP header such as MP_CAPABLE, MP_JOIN, MP_PRIO, ADD_ADDR, REMOVE_ADDR, etc., to perform various operations related to subflow [27,33]. MPTCP uses two levels of data sequences: subflow level and connection level. Subflow-level sequence numbers are similar to traditional TCP sequence numbers, which depict the number of packets on particular subflows while connection-level sequence numbers depict the sequence number of packets distributed among multiple subflows.

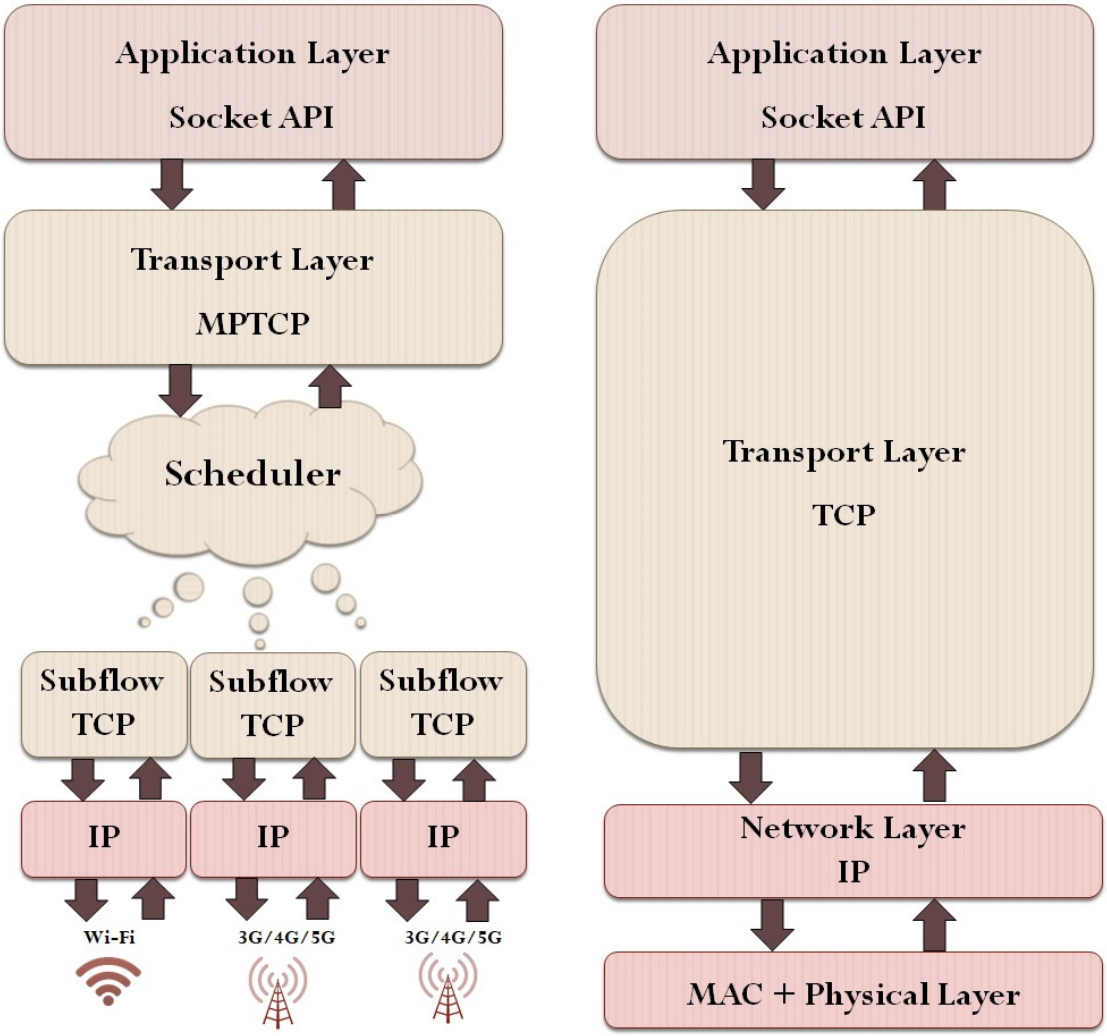


**Figure 2.** MPTCP vs. TCP network stack.

MP_CAPABLE [26] is used during the initial handshake process for connection establishment to indicate that the host supports the MPTCP connection. Similar to TCP, MPTCP hosts also exchange SYN, SYN+ACK, and ACK packets for initial connection establishment by exchanging the MP_CAPABLE option and flags along with security keys, which will be used in the future for authentication of end users to initiate subflows over the same

connection. During the initial three-way handshake, if one of the hosts does not support MPTCP, the connection is returned to the TCP.

The MP_JOIN [26] option is available to add another subflow with the pre-established connection. This process follows the four-way handshake for adding the subflow by exchanging SYN, SYN+ACK, ACK, and ACK. During the SYN+ACK and first ACK, the HMAC of the keys exchanged during the initial handshake will be used for authentication.

ADD_ADDR [26] is another option available with MPTCP use, which one can use to communicate with other hosts regarding the availability of a new interface. The host can also communicate the unavailability of any of the network interfaces during the lifecycle of the connection by using REMOVE_ADDR.

MPTCP Linux kernel Version (v1) [17] supports the same options, but packet sequences are changed in some cases. In MPTCP Version (v1), the ADD_ADDR option carries a truncated HMAC for authentication.

Figures 3–6 show the packet exchange scenario for connection establishment [26], adding a new subflow [26] and advertisement of new IP address [17,26].

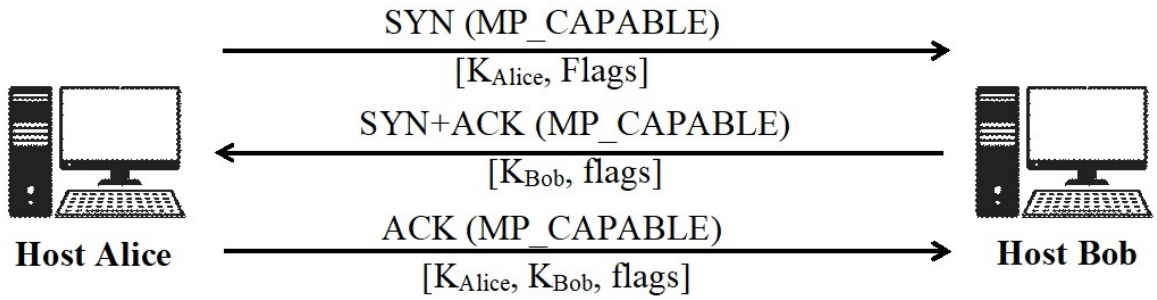


**Figure 3.** MPTCP Options in Version (v0) Connection Establishment with MP_CAPABLE.

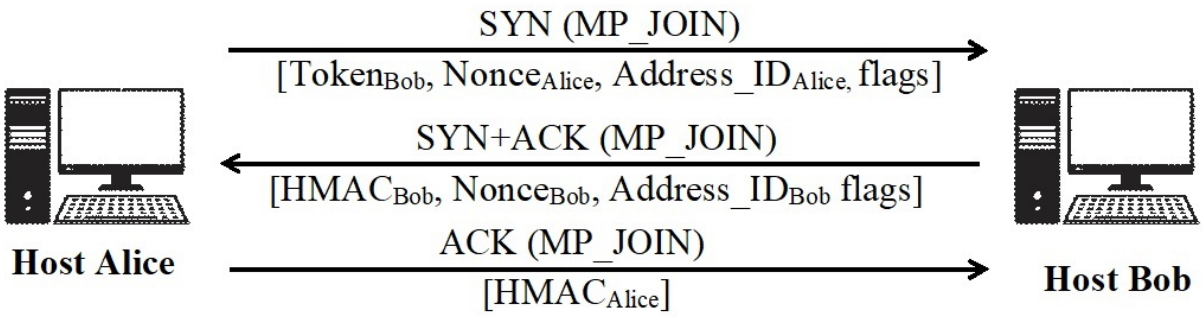


**Figure 4.** MPTCP Options in Version (v0) adding new subflow using MP_JOIN.

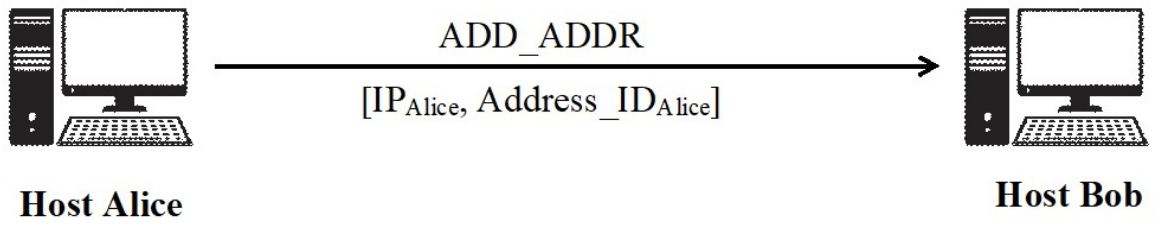


**Figure 5.** MPTCP Options in Version (v0) advertising new addresses with ADDR.

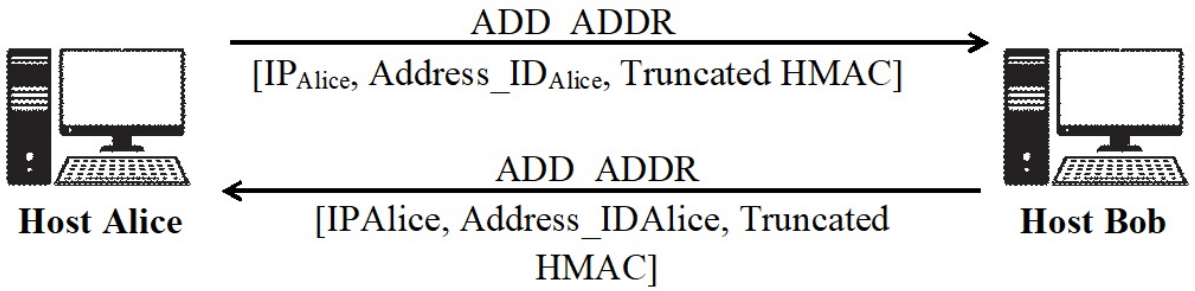


**Figure 6.** Advertisement of the new IP address with ADD_ADDR in MPTCP Version (v1).

These options of MPTCP make it vulnerable to various threats [34] by allowing an attacker to gain access to the MPTCP session. The session can be hijacked either by forging the keys communicated during the three-way handshake or by adding the forged address using ADD_ADDR packets or by using MP_JOIN packets on the communicating host. During the session hijacking, Bob will assume that the new subflow will be established with a legitimate user Alice only, and Alice will think that the request is coming from Bob, but in the backend, the subflow will be established with an attacker from both ends, and the attacker will be successful in implementing a man-in-the-middle attack. By using the compromised subflow, the attacker can monitor, manipulate or gain access to the entire connection by terminating the legitimate subflow. Moreover, the key exchange in plaintext during the initial handshake in MPTCP welcomes many security threats because these keys are used for subflow authentication in the future, which leads to SYN flood attack, MP_JOIN attack, SYN/MP_JOIN attack, session hijacking, traffic diversion attack, etc. [34].

On the basis of the location of the attacker, the attacks can be categorized as on-path (O), off-path (F) or partial-time-on-path (P) attackers, while based on the impact, attacks can be categorized as active or passive attacks [34]. On-path attackers stay on any one of the paths between the communicating hosts during their life span. Unlike on-path attackers, off-path attackers never rely on any of the paths of MPTCP during the connection life span. Partial-time-on-path attackers may stay on any one of the paths between the communicating hosts for at least some time. The significant threats to MPTCP are as shown in Table 2.

**Table 2.** Significant threats to MPTCP.

| Attack | Category * | Active/ Passive | References | Security Goals Impacted # | Remarks |
|---|---|---|---|---|---|
| Eavesdropper in initial handshake | P | Active | [34] | C | During the three-way handshake, the session keys are exchanged in clear format, which can be used in the future to initiate a SYN+MP_JOIN DoS attack or an ADD_ADDR attack. |
| ADD_ADDR attack | F | Active | [34,35] | C, I, A | By packet forging, an attacker can send the spoofed packet to the legitimate user and add the attacker's address as a legitimate address to add subflow between the authenticated host and the attacker over a legitimate connection. |
| ADD_ADDR2 attack | F | Active | [2] | C, I | The eavesdropper in the initial handshake can gather the keys exchanged between communicating hosts and use those keys to perform this attack by using the keys to find out the HMAC. |

**Table 2.** *Cont.*

| Attack | Category * | Active/ Passive | References | Security Goals Impacted # | Remarks |
|---|---|---|---|---|---|
| DoS attack on MP_JOIN | F | Active | [34] | A | The legitimate users will not be able to create new subflows by sending fake SYN+MP_JOIN requests, which will make the server busy; thus, the server will not be able to handle the requests of legitimate users. |
| SYN Flooding attack | F | Active | [34] | A | By using the SYN packet, the server will be exhausted; thus, the client will not be served. |
| Traffic diversion attack | F | Active | [36] | C, A | By cross-path inference, an attacker can monitor one of the subflows, and by using a forged MP_PRIO packet, all the traffic can be redirected to the compromised subflow. |
| Cross path inferences attack | F | Active | [37] | C, A | Attackers can infer the properties and sensitive information of an unmonitored path through side channels to create a negative impact on the design goals of MPTCP. |
| SYN/JOIN attack | P | Active | [34] | C, I, A | If the attacker is on the path during the initial SYN/JOIN message exchange, the attacker will be able to add any of the addresses to establish a new subflow over the connection. |
| Data Sequence signal manipulation | F | Active | [38] | A | The connection level ACK is manipulated on the top of the TCP optimistic ACKing, which will lead to a powerful attack scenario such as DoS, flood, etc. |

Keys: * Category: O, on-path; F, off-path; P, partial-time-on-path. # Security Goals: C, confidentiality; I, integrity; A, availability.

In this article, the eavesdropper in the initial handshake and ADD_ADDR vulnerabilities are focused. The steps to exploit the ADD_ADDR vulnerability to initiate the attack to hijack the connection are demonstrated in Figure 7. The attack covered in [2,34,35] can be initiated by forging the ADD_ADDR packet to add the IP address of the attacker as an additional IP address by impersonating the identity of the legitimate user. The same address can be used to establish the subflow over a legitimate connection to hijack the session or to redirect the traffic flow on the compromised path. In order to advertise the additional IP address, the host needs to send the ADD_ADDR packet with an IP address to be added as an additional IP and address identifier as shown in Figure 5. The attacker can easily forge this packet by identifying the source IP–port pair and destination IP–port pair. The service offered by the server can be used to identify the port of destination, as for in most cases, port 80 is used for http. Various packet sniffing tools, such as scapy, wirshark, etc., can be used to sniff and forge the packet to initiate the various attacks. The prerequisite information such as packet sequence number, IP address, port, etc., to initiate the attack can be captured through these sniffing tools. After obtaining the IP–port pair and sequence number, one can initiate the ADD_ADDR attack by using the steps shown in Figure 7 [2].

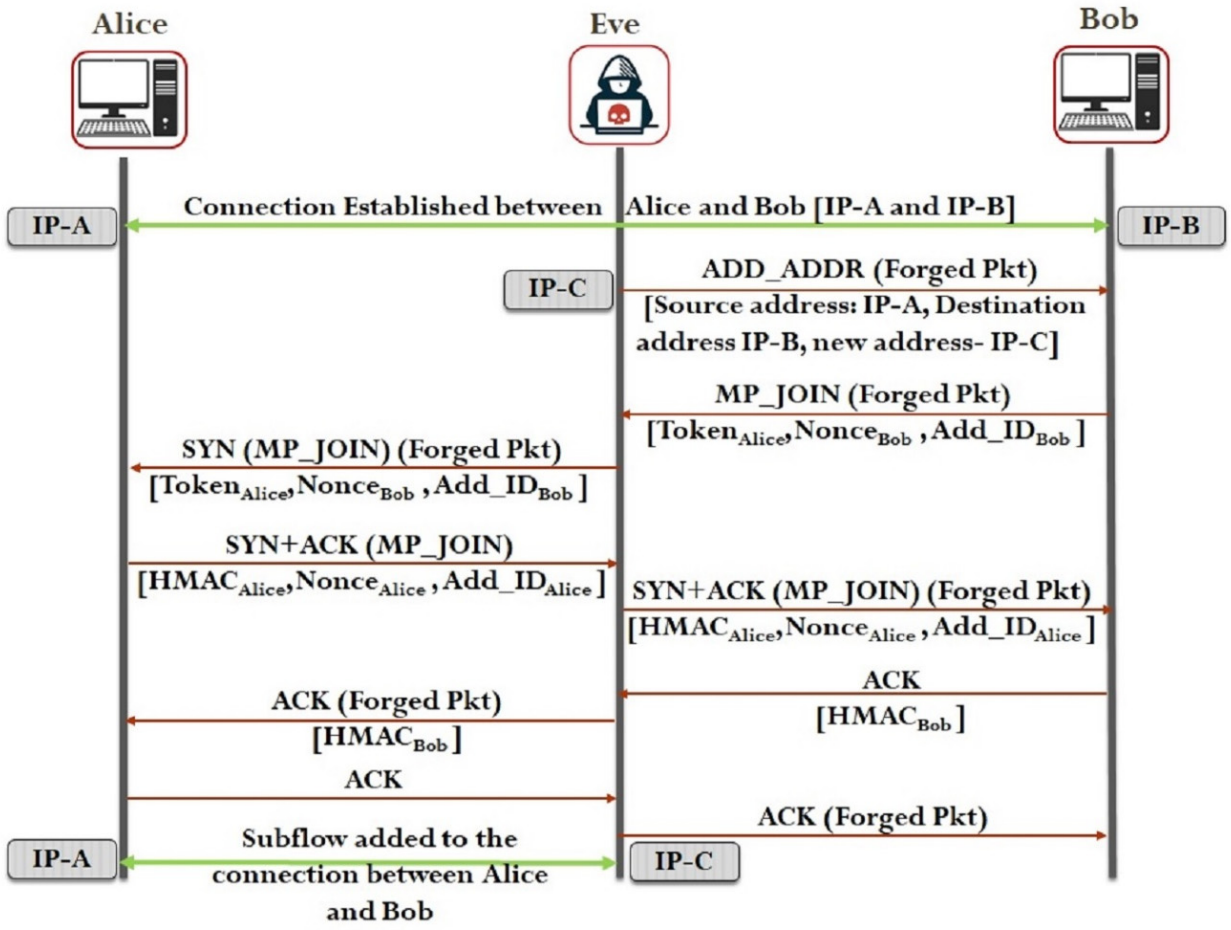


**Figure 7.** Use of ADD_ADDR vulnerability to initiate the attack to compromise the connection.

Here, Alice and Bob are the legitimate users who are communicating with each other through the connection established on IP-A and IP-B. Eve, an attacker, tries to add his address IP-C by impersonating the identity of Alice using the ADD_ADDR packet. Now, Bob will have the illusion that IP-C is the IP address, which is advertised by Alice; thus, he sends a request using MP_JOIN to add another subflow on the connection. Eve sends the forged packet to Alice by changing the source IP, and Alice has the illusion that the request is coming from Bob; thus, she replies with her HMAC, which will be used by Eve to authenticate herself as Bob, and again this packet is forged by Eve and sent to Bob and so on. After the four-way handshake, the new subflow will be established between Alice and Eve. Now, the actual situation and illusionary scenario is represented in Figure 8. Eve can change the priority of the subflow by sending the MP_PRIO packet to hijack the whole session.

### *2.2. Available Solutions to Enhance the Security of MPTCP*

Many solutions are available to enhance the security of MPTCP by preventing various attacks such as session hijacking, traffic diversion, DoS attacks, etc. In this section, the various solutions are covered and analyzed to identify the open paths for researchers in the area of MPTCP security. In order to fulfill the basic security goals (confidentiality, integrity, and availability), the keys shared during the initial handshake must be secured from eavesdroppers. The eavesdropper can use these keys to initiate other attacks as well. The encryption, hashing, and public key infrastructure are the various areas, which can be used to solve the issue, but MPTCP uses the TCP header to communicate various information.

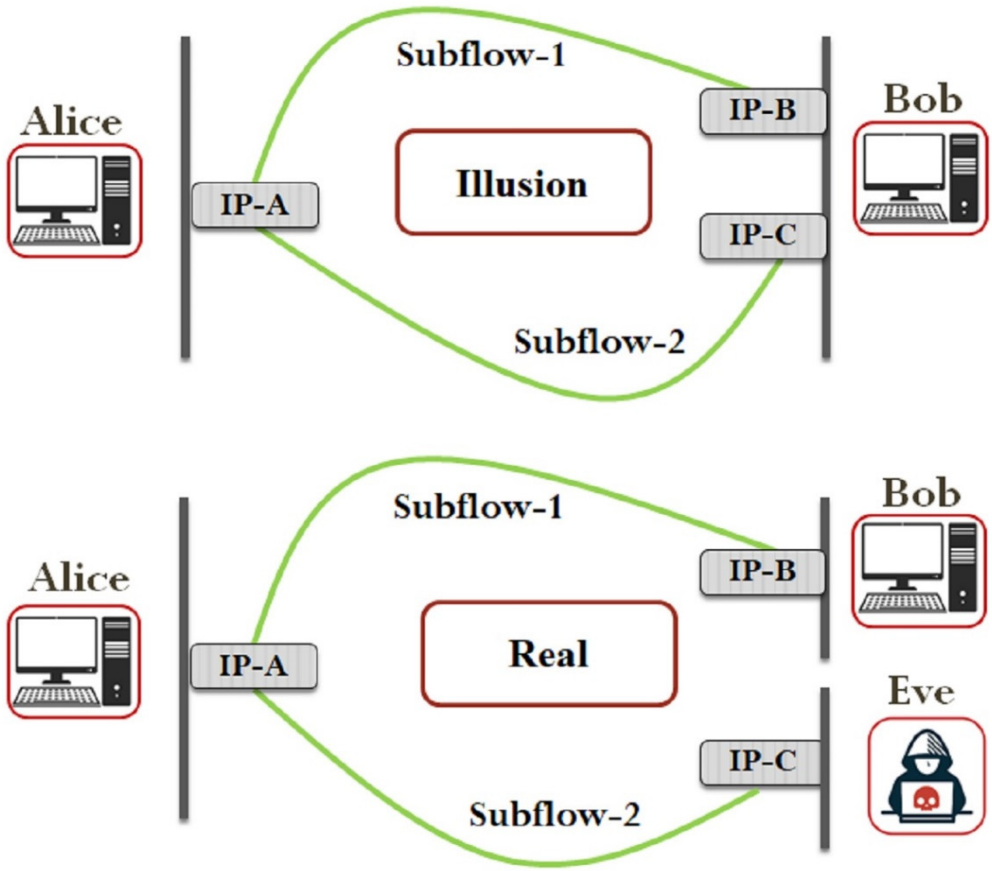


**Figure 8.** Real vs. illusion for Alice and Bob during the session hijacking attack.

The hash chain-based solution proposed by [20] uses the hashing algorithm recursively to avoid the usage of the same key for future authentication, but the initial random values are shared during the three-way handshake through which an eavesdropper can gain access to the initial values and hijack the upcoming session. The sum chained hash-based solution [21] is an extension to the hash chain-based algorithm, which is vulnerable to integrity time-shifted attacks. Both the solutions use hashing techniques to enhance the security of MPTCP, but none of them can prevent the attacks initiated by the eavesdropper in the initial handshake.

Tcpcrypt falls under the category of opportunistic security solutions, which use public key encryptions to offer cryptographic protection to enhance the security by using the session ID for individual TCP subflow. TLS is much more efficient then tcpcrypt, but TLS focuses on the security at the application level, which is again not solving the security issues related to MPTCP and TCP. Moreover, the use of an asymmetric key cryptosystem for subflow authentication increases the overall performance of MPTCP. TLS offers the facility to return back to TCP by detecting the attack, which slows down the whole communication. The use of long security keys increases the requirement for computation power.

In [39], the authors proposed that MPTCPsec offers authentication and encryption for the MPTCP. MPTCPsec prevents DoS attacks by authenticating every packet option. Moreover, it offers security against packet injection attacks by preventing the use of unsecure subflows using the MP_PRIO option to change the priority of the infected subflow. In [24], the authors have proposed a model that uses ECC by exchanging the points during the initial handshake by using a four-way handshake mechanism. This scheme decreases the overall computation overhead of the network, as it uses the ECC at the time of addition of a new subflow. The proposed model by the authors is vulnerable to an attacker that is present during the three-way handshake and can use the points to obtain the session key, which can be used in initial various types of attacks.

The advanced version of ADD_ADDR [17] has been integrated with the Linux kernel implementation of the MPTCP current version (v1) to offer security against ADD_ADDR vulnerability, but still, the attacker available during the three-way handshake can initiate the session hijacking by calculating the HMAC for the authentication using the keys exchanged during the initial handshake.

Table 3 [2] compares various solutions available to enhance the security of MPTCP and its limitations, which offers paths to researchers to think in the area of MPTCP security.

**Table 3.** Existing security solutions for MPTCP.

| References | Year | Solution | Remarks |
|---|---|---|---|
| [20] | 2011 | Hash chain-based solution | It does not offer security against on-path active attackers. |
| [19] | 2014 | Tcpcrypt | It does not authenticate the public key and is vulnerable to man-in-the middle attacks. |
| [18] | 2016 | Multipath Transport Layer Security (MPTLS) | Computation overhead during initial handshake. Need to modify the packet sequence. |
| [40] | 2015 | | |
| [24] | 2016 | Modified initial handshake | During initial handshake, the values of the points are communicated in a clear format, which can be used in the future to initiate time-shifted attack. |
| [21] | 2017 | Sum chain-based solution | Vulnerable to time-shifted attack. |
| [41] | 2017 | Data Scrambling technique for privacy | The proposed model only focuses on the eavesdropper on untrusted paths and does not work in a strict sense. Moreover, integrity of the data is not guaranteed. |
| [17] | 2018 | ADD_ADDR2 | Vulnerable to time-shifted attack. |
| [22] | 2019 | Key exchange through SDN | Single point of failure. |
| [33] | 2020 | Secure connection Multipath TCP (SCMTCP) | For each new connection request, it generates the unique key for each option, which increases the computational overhead. |
| [23] | 2019 | Secure and lightweight connection establishment scheme | Increases the packet overhead every time, confirming the new address and does not offer security against an eavesdropper in the initial handshake. |

## 3. Related Work: Identity-Based Encryption (IBE) and Elliptic Curve Cryptosystem (ECC)

The Bonah and Franklin [25] referred the IBE algorithm proposed by Shamir to drop the use of certificate authority (CA) for email application. Assume that the sender Alice sends an email to "Bob@mail.com" by encrypting the message using the public key "Bob@mail.com" and sending it to Bob. After receiving the encrypted mail from Alice, Bob requests a private key generator (PKG) for the private key through authentication and decrypts the mail. The private key will be delivered to Bob in his mailbox. In this way, one can share encrypted mail without setting up a public key infrastructure. The whole scenario [42] is as shown in Figure 9.

By using similar concepts, the session keys that are being exchanged during the initial handshake in MPTCP can be encrypted by using IBE. Here, the Internet Protocol (IP) address of the host can be used as a public parameter, and the corresponding private key can be retrieved from the PKG. Instead of delivering the private key to the mailbox, the private key will be handed over to the receiver by authenticating the receiver by using its IP address and port combination, which offers the service/receiving service. The communication between the hosts and PKG is secured using the ECC. The security level offered by ECC is similar to the use of a smaller key size as compared to RSA [31,43].

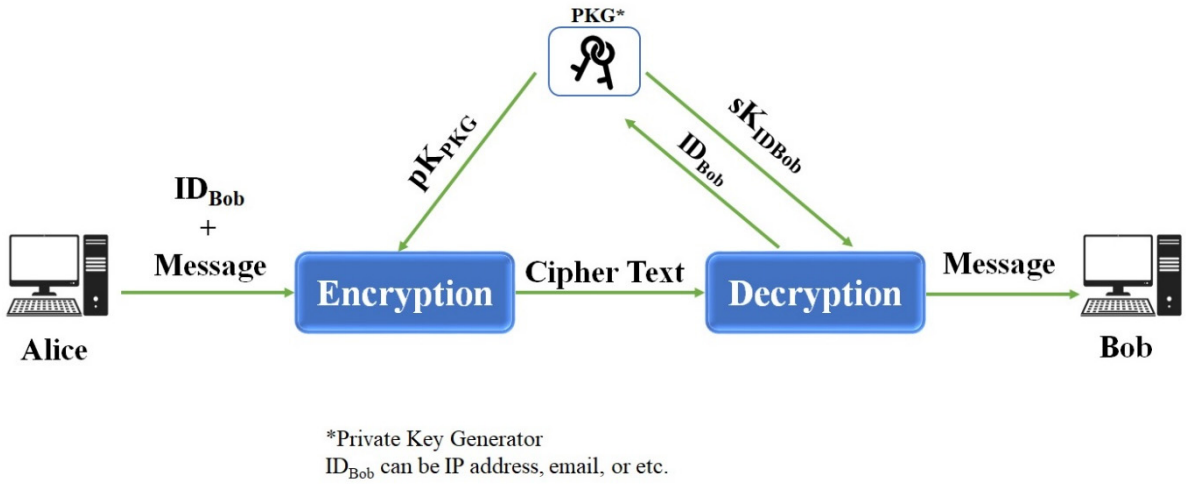


**Figure 9.** Identity-based encryption.

*Secure Communication between PKG and Communicating Hosts*

In order to secure the communication between hosts and PKG for the exchange of public parameters and private keys for IBE, the ECC is used. The ECC [44,45] is based upon the elliptic curve $E_p(a, b)$ over finite field $GF(2^m)$, which satisfies the below cubic equation.

$$y^2 = x^3 + ax + b,$$

where $4a^3 + 27b^2 \neq 0,\ a, b,\ x,\ y\ \in GF(2^m)$.

Here, $x$ and $y$ are the variables, and $a$ *and* $b$ are the coefficients. The equation represents a non-singular elliptic curve, where $x^3 + ax + b = 0$ has three distinct roots. On the basis of the selection of $a$ and $b$, the shape of the curve may vary.

The ECC can be defined using the Discrete Logarithmic Problem (ECDLP) and the Diffie Hellman Problem (ECDHP). ECDHP is a key agreement protocol, which works upon the concept of Diffie Hellman over elliptic curves. ECDLP is an extended version of DLP over a finite field. Here, ECDLP and ECDHP are used to secure the connection between the communicating hosts and PKG.

ECDLP can be defined by considering equation $Q = kP$, the discrete logarithm of $Q$ to the base $P$, for the given points $P$ and $Q$ in group $E_p\ (a, b)$ and $k < p$ It will be exceptionally challenging for the adversary to figure out the value of $k$. ECDHP uses elliptic curves for key exchange. First, select the large integer number $q$. Here, $q$ must be selected in such a way that it is either the prime number $p$ for the $Z_p$ or is the integer of the form $2^m$. In addition, select the coefficients $a$ and $b$ for the elliptic curve $E_q(a, b)$. Now, select point $G = (x_1, y_1)$ on the elliptic curve of the order of a large number $n$. Here, $E_q(a, b)$ and $G$ are the global parameters, which will be used for the key generation and will be known to all the participants. The key exchange process for Alice and PKG and Bob and PKG is as follow:

1. Alice selects the private key $n_{Alice}$, which is less than $n$. The public key $P_{Alice}$ will be point $(X_{Alice}, Y_{Alice})$ in elliptic curve $E_q(a, b)$, which can be generated by using the private key $n_{Alice}$ and global parameter $G$ using the below equation.

$$Public\ Key\ P_{Alice} = n_{Alice} \times G$$

2. By using the same equation, PKG can select the private key $n_{PKG}$ and generate the public key $P_{PKG.}$
3. Now, Alice can generate the secret key $K = n_{Alice} \times P_{PKG}$, and PKG can generate the secret key $K = n_{PKG} \times P_{Alice}$. Here, the secret key generated by Alice and PKG will be the same, which will be used for communication.

The session key for communication between Alice and PKG is as below:

$$S_k = HMAC\ (K,\ (Y_{Alice} \oplus Y_{PKG}))$$

The same process can be used to generate the secret key for the communication between Bob and PKG for secure communication as shown in Figure 10.

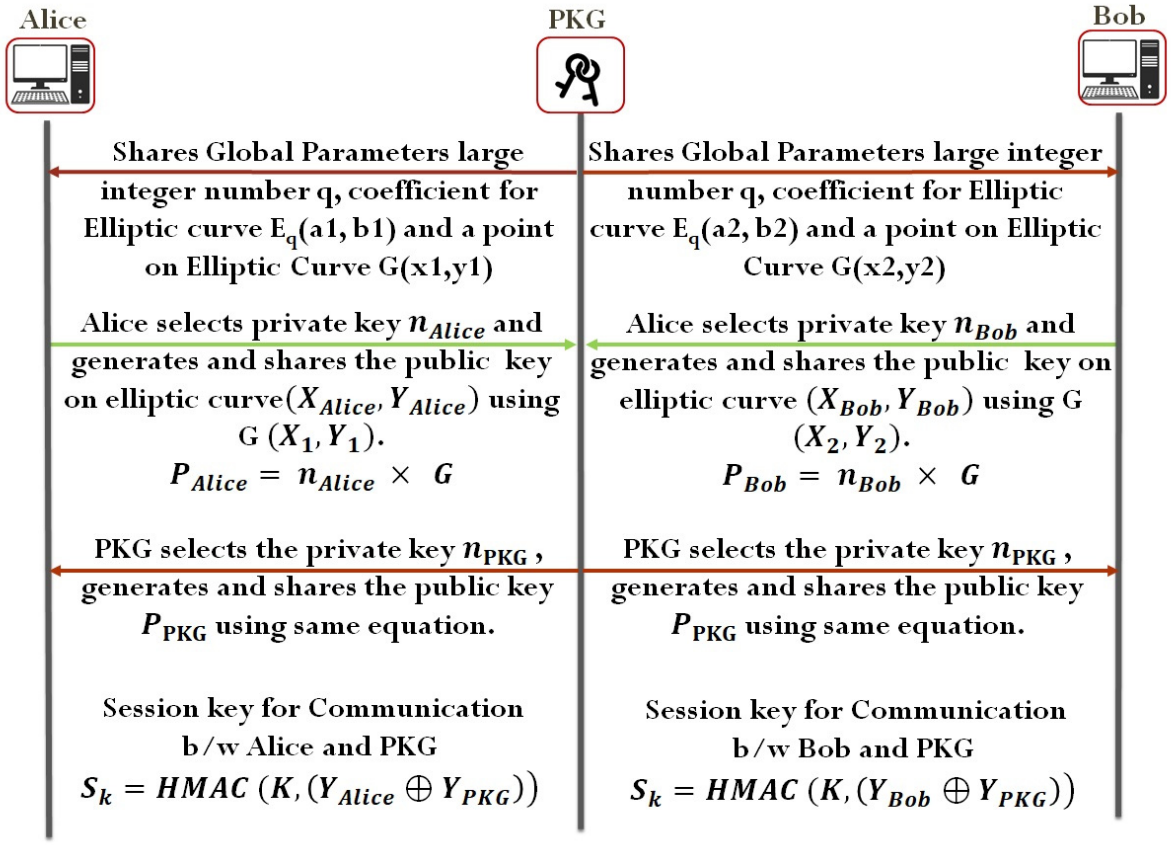


**Figure 10.** Key exchange scenario between Alice–PKG and Bob–PKG.

The IBE used for SKEXMTCP uses the Weil pairing on elliptic curves proposed by Bonah and Franklin [25], in which it can be defined using any of the bilinear maps $\hat{e} : G1 \times G2 \rightarrow G2$ where $G1$ and $G2$ are two cyclic groups of order $q$ for some large prime $q$. $G1$ can be viewed as an additive group, as it is the group of points of an elliptic curve over $F_p$ and $G2$, which can be viewed as a multiplicative group, as it is a subgroup of $F^*_{P2}$.

The IBE uses four algorithms: SetUp, KeyGeneration, Encryption, and Decryption for the implementation of the entire scenario [25].

- SetUp: By using this algorithm, the PKG provides the system parameters and master-key share, which will be used to generate the private key. This algorithm takes security parameter k $\in Z^+$, $G1$ and $G2$ to build a bilinear map $\hat{e} : G1 \times G2 \rightarrow G2$, 2 cryptographic hash functions $h1$ and $h2$ and returns system parameters $Params = \langle q,\ G_1,\ G_2,\ \hat{e},\ n,\ P,\ P_{pub},\ H1,\ H2 \rangle$ and master key share. Here, $q$ is the prime number, and $P$ is the random number generator, which generates the random number $s$ to obtain the value of $P_{pub} = sP$, where s is the master key share.
- KeyGeneration: By using this algorithm, the private key of the particular entity will be generated on request by using system parameters, master-key share, and public parameters. This algorithm uses the master key share and $ID$ to generate the private key $d_{ID} = sQ_{ID}$, where $s$ is a master key and $Q_{ID} = H1(ID) \in G^*_1$.

- Encryption: During the encryption, the $ID$ of the receiver and system parameters will be used to encrypt a message, and ciphertext will be generated. This algorithm computes the Cipher Text $C$ using $\langle rP,\ \sigma \oplus H2\ (g_{ID}{}^{r})\rangle$, and $g_{ID} = \hat{e}\left(Q_{ID},\ P_{pub}\right) \in G_2$, $\sigma$ is a random number and $r = H3(\sigma,\ M)$.
- Decryption: The private key generated by the KeyGeneration algorithm, system parameters generated by the SetUP algorithm, and the $ID$ of the receiver will be used to decrypt the ciphertext. The Plaintext message $M$ can be computed as below: if $C = \langle U,\ V,\ W\rangle$ is a cipher text encrypted using public key $ID$, compute $V \oplus H2\ (\hat{e}\ (d_{ID}\ ,\ U)) = \sigma$ and $W \oplus H4\ (\sigma) = M$ and $r = H3(\sigma,\ M)$. Test that $U = rP$. If not, reject the cipher text. Here, $M$ is the decryption of $C$.

## 4. Proposed Work: Secure Key Exchange Model for MPTCP (SKEXMTCP) Using Identity-Based Encryption

The current version of MPTCP suffers from security weaknesses such as ADD_ADDR vulnerability, SYN MP_JOIN vulnerability, eavesdropper in the initial handshake, etc., which lead to dangerous security attacks, such as man-in-the-middle attacks, DoS attacks, and session hijacking attack, which threaten the confidentiality, integrity, and availability of data over the connection. To prevent the communication over MPTCP from ADDR_ADDR attack and eavesdroppers in the initial handshake, the SKEXMTCP is proposed here. The SKEXMTCP uses the identity-based encryption scheme to exchange secret keys, which will be used to exchange the security parameters for MPTCP during the initial handshake. This will provide security against an eavesdropper in the initial handshake, which leads to the prevention of an ADD_ADDR attack as well. IBE uses a Private Key Generator (PKG), a third-party authority, which provides the private keys to the communicating hosts based on their identity (i.e., email id, IP address, etc.). Here, the IP address and port of the communicating host will be used as a public parameter to generate the private keys for the sender and receiver. The proposed solution contains two modules: (i) Private Key Generation (SKG_SKEXMTCP); (ii) use of key pairs to exchange session keys during the initial handshake (MPC_SKEXMTCP).

In the Key Generation (SKG_SKEXMPTCP) module, considering that host Alice and host Bob are communicating with each other with IP addresses $IP_{Alice}$ and $IP_{Bob}$ using MPTCP, here, $IP_{Alice}$ and $IP_{Bob}$ will be used as public parameters by PKG to generate private keys for Alice and Bob. These private keys will be used in the MPC_SKEXMTCP module to exchange the keys during the initial handshake by encrypting them using IBE. The communicating host will be authenticated by using the IP address and port combination digitally signed by the host at the PKG. Table 4 shows the terms used for the algorithm.

**Table 4.** Terms used in SKEXMTCP.

| Term | Significance/ Meaning | Generation |
|---|---|---|
| $PU_{Alice}$ | Public key of Alice | IP address of the Alice will be used as a Public Key. |
| $PR_{Alice\_Master}$ | Shared key used to generate the private key of Alice | Generated by PKG and shared with Alice. |
| $PR_{Alice}$ | Private Key of Alice | It can be generated by using $PR_{Alice\_Master}$ and $PU_{Alice}$. |
| $PU_{Bob}$ | Public key of Bob | IP address of the Bob will be used as a Public Key. |
| $PR_{Bob\_master}$ | Shared key used to generate the private key of Bob | The key will be generated by PKG and shared with Alice. |
| $PR_{Bob}$ | Private Key of Bob | It can be generated by using $PR_{Bob\_Master}$ and $PU_{Bob}$. |
| $ID_{Alice}$ | ID used as a Public key of Alice | $IP_{Alice}$ + $Port_{Alice}$ Combination. |
| $ID_{Bob}$ | ID used as a Public key of Bob | $IP_{Bob}$ + $Port_{Bob}$ Combination. |

**Module 1. Key Generation (KG_SKEXMTCP) using Identity-Based Encryption (IBE) Scheme:**

**Step 1. Host Alice Key Generation**

(a) Here, public key of Alice $PU_{Alice}$ = $ID_{Alice}$.
It can be used by the sender to encrypt the messages for Alice.

(b) Host Alice sends request to PKG with $IP_{Alice}$ and $Port_{Alice}$ as a parameter by authenticating itself using a digitally signed IP address and port combination.
(c) PKG calculates the share of Alice $PR_{Alice\_master}$, and it will be sent back to Alice.
(d) Alice can calculate the private key $PR_{Alice}$ from the $PR_{Alice\_master}$. $PR_{Alice}$ = Generate ($PR_{Alice\_master}$, $ID_{Alice}$). The messages encrypted by $PU_{Alice}$ can be decrypted using $PR_{Alice}$.

**Step 2. Host B Key Generation**

(a) Here, public key of Bob $PU_{Bob} = ID_{Bob}$.
It can be used by sender to encrypt the messages for Bob.
(b) Host Bob sends request to PKG with $IP_{Bob}$ as a parameter by authenticating itself using digitally signed IP address and port combination.
(c) PKG calculates the share of Bob $PR_{Bob\_master}$, and it will be sent is back to Bob.
(d) Bob can calculate private key $PR_{Bob}$ from the $PR_{Bob\_master}$. $PR_{Bob}$ = Generate ($PR_{Bob\_master}$, $ID_{Bob}$). The messages encrypted by $PU_{Bob}$ can be decrypted using $PR_{Bob}$.

**Module 2. Initial Handshake using MP_CAPABLE with SKEXMTCP (MPC_SKEXMTCP)**

**Step 1. SYN [MP_CAPABLE]**

(a) Encryption of Alice's key $K_{Alice}$. Alice encrypts the session key $K_{Alice}$ with public key of Bob ($PU_{Bob} = IP_{Bob}$) using IBE.
(b) Key Transmission of Alice. Alice sends the encrypted key $EK_{Alice} = En(PU_{Bob}, K_{Alice})$ to Bob with MP_CAPABLE.

**Step 2. SYN+ACK [MP_CAPABLE]**

(a) Encryption of Bob's key $K_{Bob}$. Bob encrypts the session key $K_{Bob}$ with public key of Alice $PU_{Alice}$ using IBE.
(b) Key transmission of Bob. Bob sends the encrypted key $EK_{Bob} = En\ (PU_{Alice}, K_{Bob})$ to Alice with MP_CAPABLE.

**Step 3. ACK [MP_CAPABLE]**

(a) Key Echoing

Alice sends the $EK_{Alice}$ and $EK_{Bob}$ again to complete the connection establishment.

Figure 11 shows the packet sequence and the parameters passed with each packet of the initial three-way handshake according to the proposed scheme.

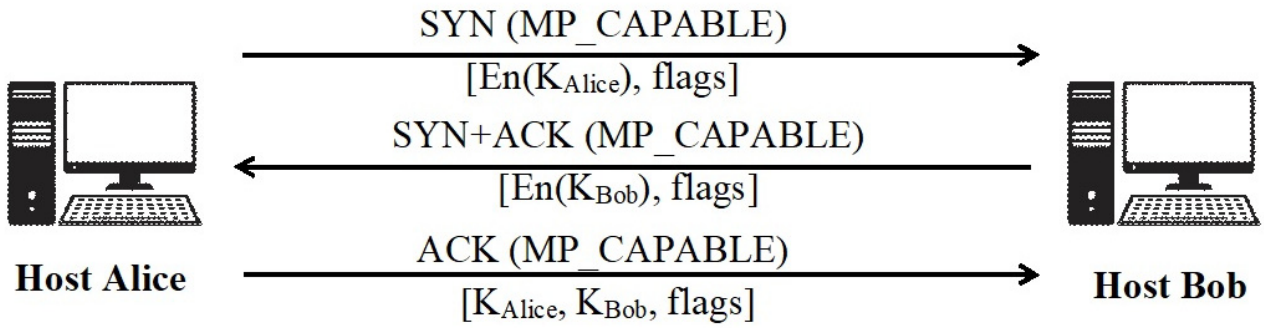


**Figure 11.** Three-way handshake with proposed scheme.

The whole scenario of module 1, which is about the key generation using IBE, and module 2, which is about the three-way handshake process of MPTCP, is represented in Figure 12. Bob needs to authenticate himself by proving his identity to PKG at the time of requesting the private key to decrypt the message received from Alice. Here, Bob proves his identity through the combination of his IP address and port number. The request for the private key will be encrypted by the session key generated using ECC and HMAC by PKG and digitally signed by the private key of Bob ($PR_{Bob}$). In order to forge the request packet,

the attacker needs to obtain the private key of Bob, which is extremely difficult. If the size of the private key is $n$ bits, the attacker needs to take at least $2^n$ trials to break the security.

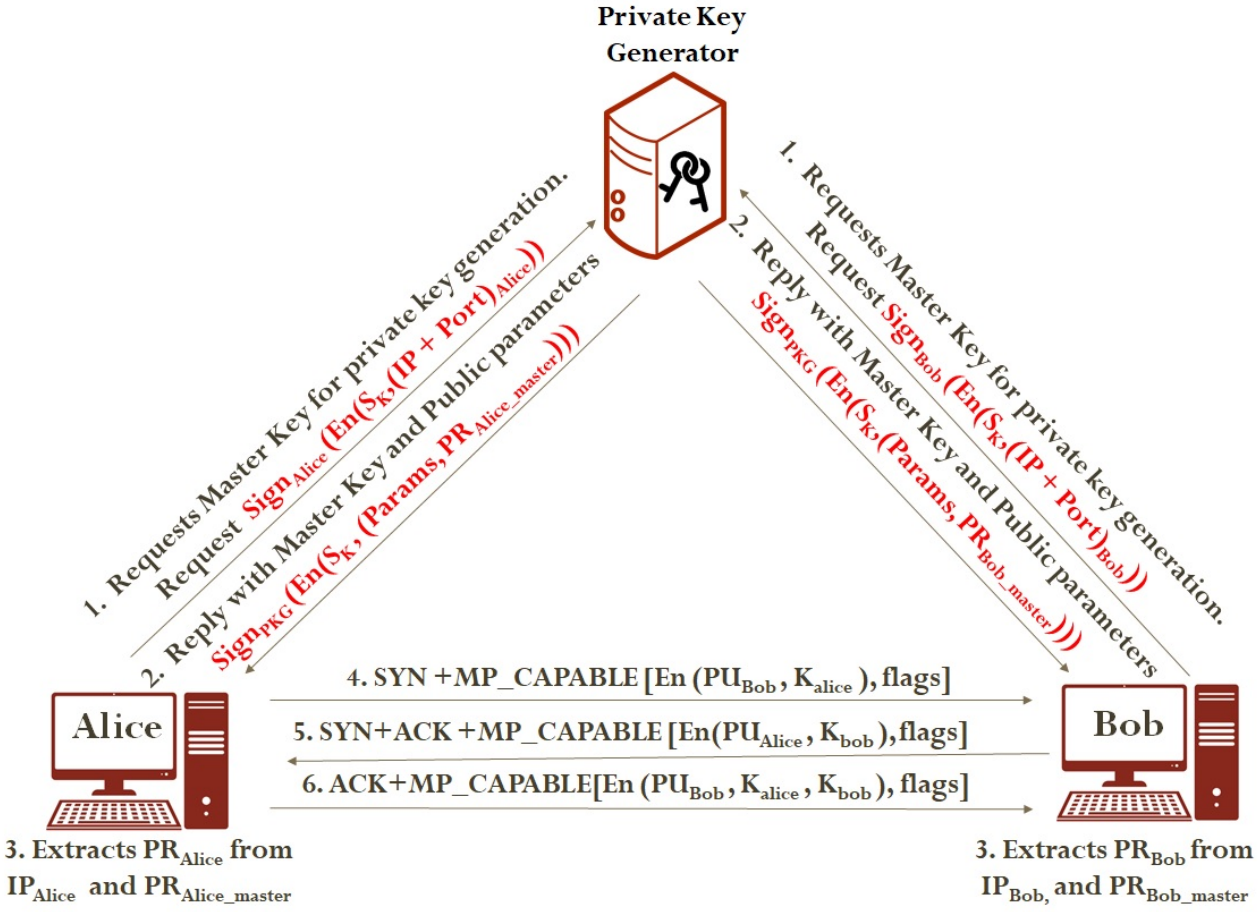


**Figure 12.** Secure key exchange during a three-way handshake using IBE.

## 5. Experimental Evaluation of Proposed Model

The proposed scheme is tested with MPTCP using the Linux kernel implementation of MPTCP. The Oracle VirtualBoxes are used to set up the scenario of the proposed scheme by creating two virtual machines (VMs), client and server, as shown in Figure 13. The client VM and server VM are configured with the Linux kernel implementation of MPTCP. Here, the PKG is configured on the host machine. In order to connect PKG with any of the hosts with MPTCP, the tap interfaces are used.

The IBE requires PKG for generating system parameters and distributing private keys on the basis of the ID of the host. The key role of PKG is to configure the system parameters and master share, which can be used during the encryption and decryption of the keys shared during the initial handshake of the MPTCP. The proposed scheme uses IBE for encrypting the data without communicating keys and with a communicating host. Here, PKG plays a significant role in authenticating the users and sharing the master private key to generate the private key using identity. Our proposed model uses ECC for the generation of session keys, and each communication will take place by digital signature for authentication.

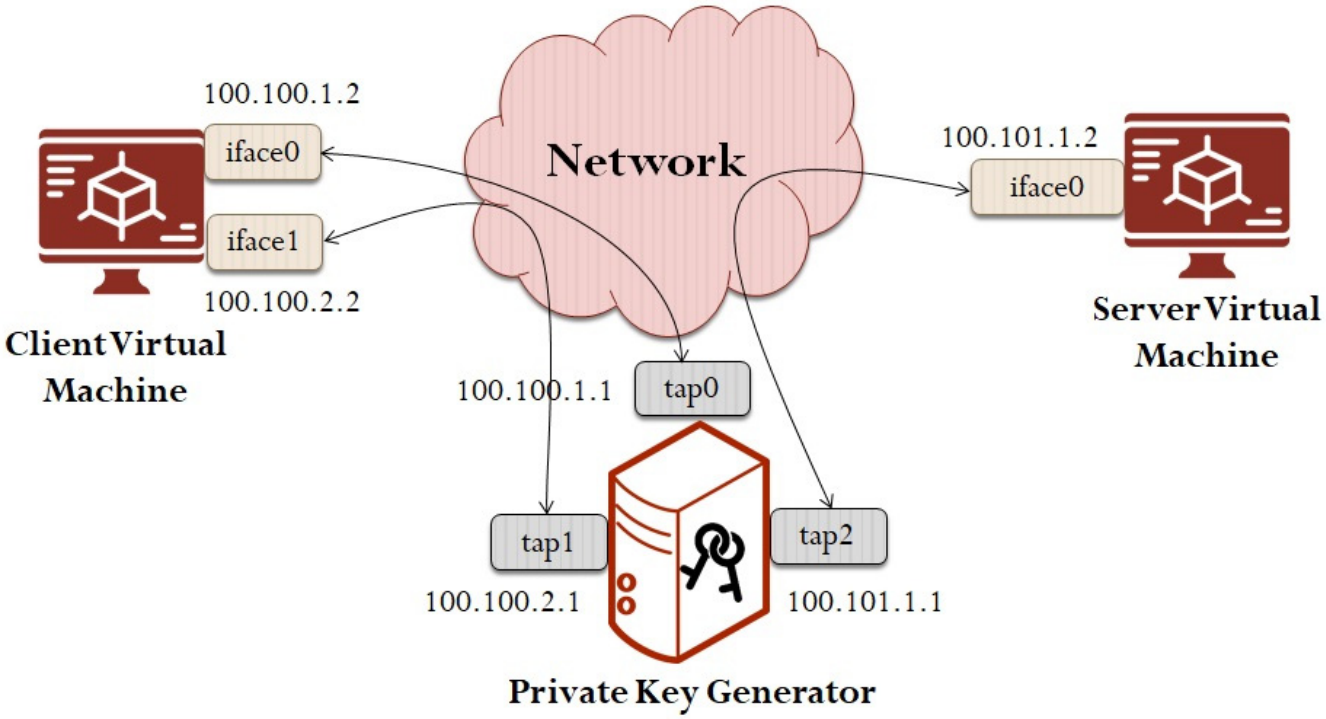


**Figure 13.** Experimental setup for testing the proposed work.

### 5.1. Security Evaluation

The proposed model, SKEXMTCP, uses the IBE technique to encrypt the session keys being exchanged during the initial handshake. In order to encrypt the session keys using IBE, the client uses the IP address as a public key, and the server obtains the corresponding private key from the PKG to decrypt the session keys. Here, the server authenticates itself by using the IP address and port number, which will be encrypted by the public key of PKG and digitally signed by the server. If an attacker tries to find out the private key of PKG to decrypt the packet and tries to change the digital signature of the server to forge the packet, it is required to break the encryption algorithm and hashing algorithm. Thus, the security complexity of the model relies upon the complexity of IBE and the encryption algorithm used for encrypting the private key request packet. Table 5 shows the comparison of whether the various solutions offer security against various attacks or not.

**Table 5.** Comparative evaluation of existing security solutions against attack vector.

| Attack | Type | Proposed Solution | SCMTCP [33] | Secure and Lightweight Subflow Scheme [23] | Secure MPTCP (SMPTCP) [46] | MPTLS [47] | Hash Chain [20] | MPTCP [17] |
|---|---|---|---|---|---|---|---|---|
| **Session hijacking using ADD_ADDR Vulnerability** | **Off Path Active attack/Partial Time on Path Active attack** | Y | Y | Y | Y | Y | N | N |
| **Eavesdropper in the initial handshake** | **On Path Attack** | Y | Y | N | Y | Y | N | N |

Keys: Y: Yes-Offers Security, N: No-Doesn't offer Security.

### 5.2. Security Complexity of IBE

The chosen-cipher text attack (CCA) is considered a standard acceptable attack model, in which attackers can gain access to plaintexts corresponding to the chosen cipher text for asymmetric encryption schemes. Hence, IBE needs to be proven as secure under the

chosen-cipher text attack (CCA) model, which can be further considered adaptive (IDA-ID-CCA) or non-adaptive (IDA-CCA). The basic indent (IBE) proposed by BONAH and FRANKLIN [25] was not a secured chosen-cipher text, but the FullIndent proposed by FUSAKI-OKAMOTO is a secured chosen-cipher text IBE, as the security of this IBE scheme is dependent on the bilinear Diffie-Hellman assumption (BDH). In the proposed model SKEXMTCP, the FullIndent version of secure IBE is used for encryption, which is secure in random oracle against the chosen-cipher text attack by assuming that BDH is hard in groups [25].

### 5.3. Security Complexity of ECC

In order to offer confidentiality of communication between PKG and the communicating node, the messages are required to be encrypted using a public key encryption scheme such as RSA, Elgamal, ECC, etc. ECC offers the same level of security as compared to RSA even after using the small size of the key. The security of the public key cryptosystem is higher if the attacker requires an exponential amount of time with respect to key size to initiate an attack.

The well-known methods to solve ECC are naïve exhaustive search, Baby Step Giant Step (BSGS), the square root, and Silver–Pohling–Hellman (SPH) [48]. The computation time required to solve ECC using the naïve exhaustive search method is $n$ = order of $P$, as it adds point $P$ to itself until it obtains $Q = kP$, which is infeasible if the number of steps is larger. The BSGS is an extension of the naïve exhaustive search, but its space complexity and time complexity is O($n$), which is too high, as it requires volatile memory for $n$ *points* and additional $n$ steps. Hence, the time required to solve ECC using the square root method varies exponentially in terms of key length, and public key cryptosystems are secure against the attack, which require an exponentially proportional time to key size, and the ECC can be considered secured against the square root method. Moreover, the SPH is only useful when the order of the curve is defined by selecting the product of small prime numbers. As for products of large prime numbers, the computation time varies exponentially for SPH, which makes ECC secure against the SPH method. One of the other general-purpose methods to solve ECC is the Pollard-p algorithm, whose time complexity is similar to BSGS, but the space complexity is O(1). The computational complexity of the Pollard method is too high, as the expenses to attack ECC-163 within 1 year are approximately USD 200 million. Moreover, the current status of the security breach of ECC [48] states that "a 112-bit key for the prime field and a 109-bit key for the binary field are the extreme level security breach till date". Thus, ECC is the best-suited scheme for encrypting the packets between nodes and PKG. The HMAC can be used to produce the session key, which uses the hash function to offer the higher level of security. The security complexity of HMAC is much higher than others.

### 5.4. Performance Evaluation

In the proposed model of SKEXMTCP, the session keys, which are used for the authentication of entities during the establishment of new subflows and advertisement of new addresses, are encrypted by using IBE. In order to retrieve the public parameters of IBE and private key from the PKG, extra packets are required to be exchanged between communicating nodes and the PKG, but it does not add any overhead on communication through MPTCP. Moreover, in order to encrypt the packet using IBE, the host does not require a public key, as any arbitrary string can be used to encrypt the messages using IBE, whereas if any of the public key cryptosystems such as ECC are used, the hosts need to generate the session keys before encryption. Figure 14 shows the comparison between the time required for the key generation and encryption using ECC and IBE. The graph shows that ECC requires more time, as it needs to generate session keys for encryption, while IBE can use any arbitrary string for the encryption. In [31], the authors used ECC for the session key generation, and the keys will be used for authentication, which decreases the overall performance.

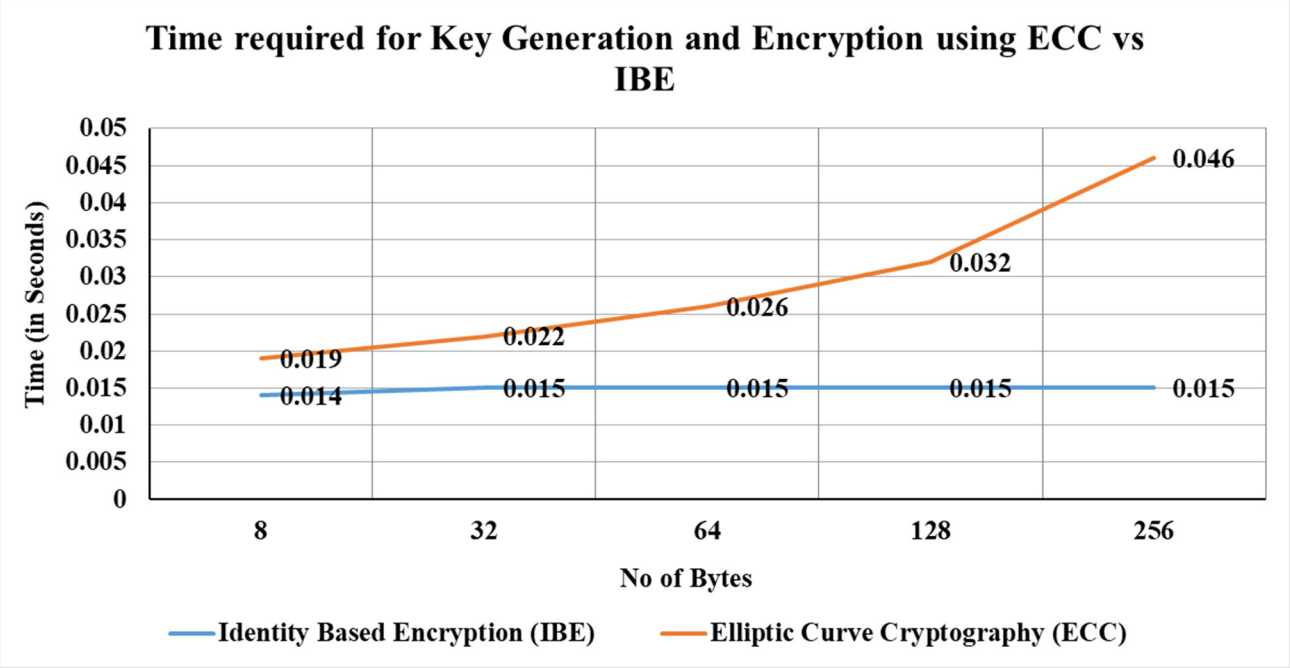


**Figure 14.** Comparison between times required for the key generation and encryption using ECC and IBE.

The cost of the proposed model in terms of implementation can be calculated by considering: (i) the cost of key generation, (ii) the cost of communication between hosts and PKG, and (iii) the cost of the three-way handshake.

- Let us assume that the cost of key generation is $n$.
- To obtain the cost of communication between the hosts and PKG, one needs to consider the cost of a request for a private key from a host to PKG and the cost of a reply from PKG to a host with a private key.
- Assume that the cost of a request for a private key from a host to PKG is $n1$ and the cost of a reply from PKG to a host with a private key is $n2$.
- Thus, the cost of communication between Alice and PKG to deliver a private key to Alice is $n1 + n2$, and the cost of communication between Bob and PKG to deliver a private key to Bob is also $n1 + n2$.
- Thus, the total cost for communication between PKG and hosts is $2(n1 + n2)$.
- Now, let us calculate the cost of a three-way handshake SYN, SYN+ACK, and ACK is $n3$, $n4$, and $n5$ respectively.
- Thus, the overall cost is

$$N1 = 2\ (n1 + n2) + n3 + n4 + n5$$

- If we consider that the overall cost of the model is $\mathrm{O}(N) = \mathrm{O}(N1)$, then

$$N1 \ll N1 \times N1$$

Table 6 shows the comparative analysis in terms of the number of bytes required for key exchange and delay for packet exchanges [23,24] of various solutions to enhance the security of MPTCP. It also shows the comparison of various proposed solutions to enhance the security of MPTCP in terms of bytes required for key exchange and no delays [22]. Here, the delay shows the number of extra packets required, which is a one-way delay. The graphical representation in Figure 15 shows that the proposed solution behaves the same in terms of required bytes in the key exchange and delay as MPTCP.

**Table 6.** Comparative evaluation of existing security solutions.

| | Proposed Solution | SCMTCP [33] | Secure and Lightweight Subflow Scheme [23] | SMPTCP [46] | MPTLS [47] | Hash Chain [20] | MPTCP [17] |
|---|---|---|---|---|---|---|---|
| **MP_CAPABLE** | | | | | | | |
| – Key exchange (bytes) | 32 | 32 | 32 | 124 | 7468 | 52 | 32 |
| – No of delay | 3 | 3 | 3 | 4 | 7 | 3 | 3 |
| **ADD_ADDR** | | | | | | | |
| – Key exchange (bytes) | 10 | 10 | 30 | 18 | 18 | 18 | 10 |
| – No of delay | 1 | 1 | 3 | 1 | 1 | 1 | 1 |
| **MP_JOIN** | | | | | | | |
| – Key exchange (bytes) | 12 | 12 | 12 | 12 | 12 | 24 | 12 |
| – No of delay | 40 | 40 | 40 | 40 | 40 | 28 | 40 |
| **MP_REMOVE** | | | | | | | |
| – Key exchange (bytes) | 10 | 10 | 30 | 18 | 18 | 18 | 10 |
| – No of delay | 1 | 1 | 3 | 1 | 1 | 1 | 1 |

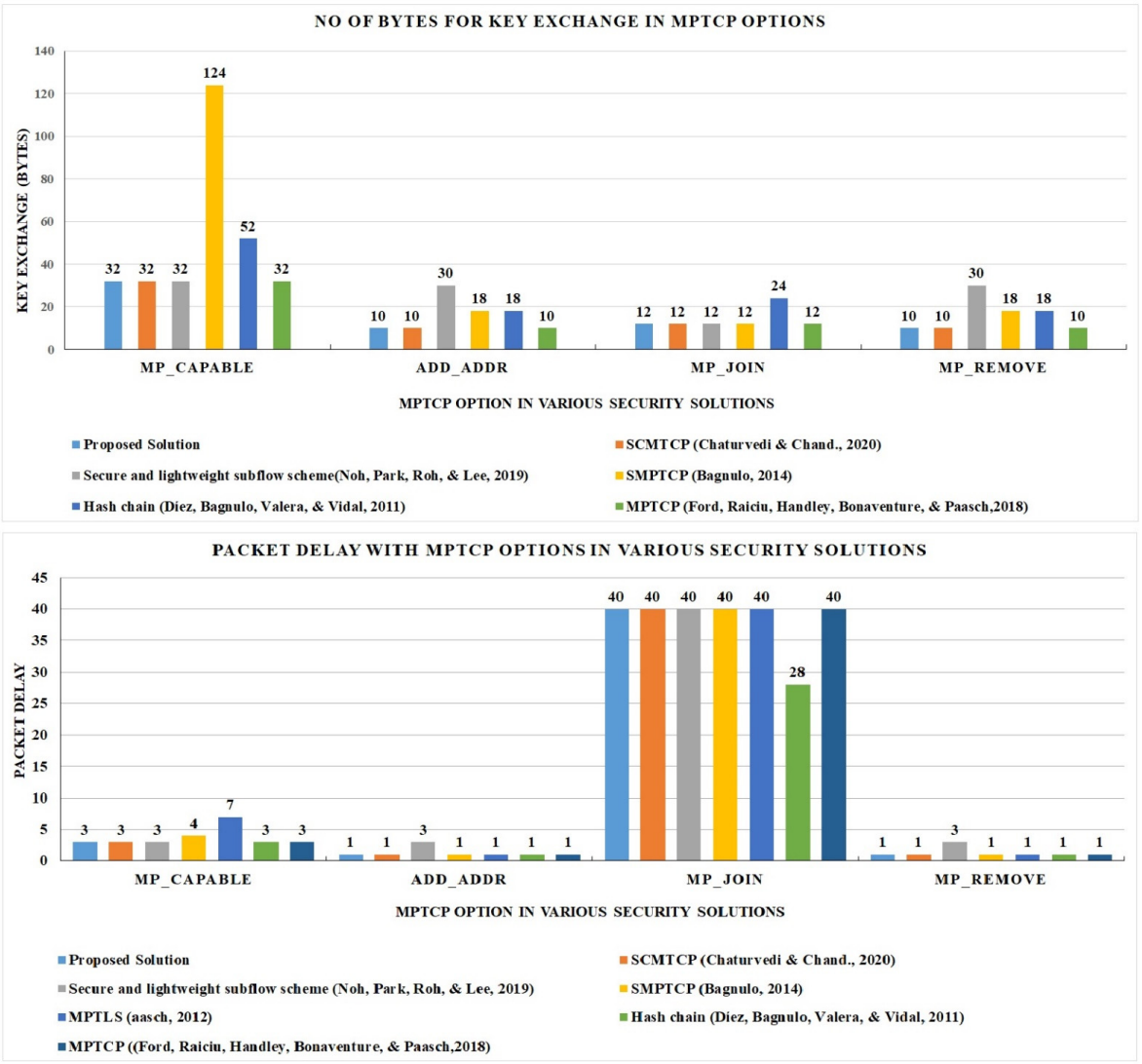


**Figure 15.** Comparative study of bytes required in key exchange with various MPTCP options in security solutions [17,20,23,24,33,46,47].

## 6. Conclusions

In this article, the security threats with respect to their position of attack and to the MPTCP by exploiting the vulnerabilities of MPTCP options are explored, and their impact on security goals have been analyzed. The step-by-step procedure to initiate an ADD_ADDR attack to hijack the connection using the ADD_ADDR option vulnerability is demonstrated in the paper. The probable security solutions for the various vulnerabilities of MPTCP have been analyzed and compared in terms of attack vectors. In order to prevent the off-path active attacks and enhance security of MPTCP from ADD_ADDR vulnerability and eavesdroppers in the initial handshake, the SKEXMTCP using IBE is proposed and tested using the Linux kernel implementation of MPTCP. Using IBE, the session keys exchanged during the initial handshake and used in the future for authentication can be encrypted by using the IP address and port (used as an ID in IBE) as a public key, and the corresponding private keys will be provided by the PKG. The security complexity and overhead of the proposed model on MPTCP is analyzed. The analysis shows that the proposed solution enhancing the security of MPTCP and does not create any overhead on the existing protocol. The main limitation of MPTCP is the available option size for MPTCP, and in TCP, the header is 64 bits, which can be increased to improve the security of MPTCP in future. Moreover, the security of the IBE scheme can be extended by increasing the complexity of BDH and by generating the unique ID for the encryption. In the future, the solution can be extended to prove the security against other categories of attacks. Moreover, the unique ID generation based on IP address and port can be proposed to generate the public key. The lightweight IBE solution can be proposed for resource-constrained devices. The integration of machine learning to identify threats and prevent attacks by data flow management in MPTCP could be a promising area of research in MPTCP security.

**Author Contributions:** Conceptualization, K.P., V.V.K., M.R.N.M.Q. and A.S.A.; methodology, K.P., V.V.K., M.R.N.M.Q. and A.S.A. software, N.A., A.S.A. and R.E.A.M.; validation, N.A., A.S.A., R.E.A.M. and V.V.K.; formal analysis, K.P., R.E.A.M. and A.S.A.; investigation, A.S.A., R.E.A.M. and N.A.; resources, R.E.A.M.; writing—original draft preparation, K.P., V.V.K., M.R.N.M.Q. and A.S.A. writing—review and editing, M.R.N.M.Q. and N.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Deanship of Scientific Research, King Khalid University, Kingdom of Saudi Arabia, and the grant number is R.G.P.2/178/43.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We would like to express our gratitude to the Deanship of Scientific Research, King Khalid University, Kingdom of Saudi Arabia for funding this work, as well as family, friends, and colleagues for their constant inspiration and encouragement.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Postel, J. Transmission Control Protocol, RFC 793. September 1981. Available online: https://datatracker.ietf.org/doc/html/rfc793 (accessed on 1 January 2022).
2. Popat, K.; Kapadia, V.V. Multipath TCP Security Issues, Challenges and Solutions. In *Information, Communication and Computing Technology*; Communications in Computer and Information Science; Springer: Cham, Switzerland, 2021.
3. Patil, S.; Raut, R.; Jhaveri, R.; Ahanger, T.; Dhade, P.; Kathole, A.; Vhatkar, K. Robust Authentication System with Privacy preservation of Biometrics. *Secur. Commun. Netw.* **2022**, *2022*, 7857975. [CrossRef]
4. Patel, C.; Joshi, D.N.; Doshi, V.A.; Jhaveri, R. An enhanced approach for three factor remote user authentication in multi-server environment. *J. Intell. Fuzzy Syst.* **2020**, *39*, 8609–8620. [CrossRef]
5. Durga, R.; Poovammal, E.; Ramana, K.; Jhaveri, R.H.; Singh, S.; Yoon, B. CES Blocks—A Novel Chaotic Encryption Schemes-Based Blockchain System for an IoT Environment. *IEEE Access* **2022**, *10*, 11354–11371. [CrossRef]

6. Dharmadhikari, O. 5G Link Aggregation with Multipath TCP (MPTCP); CableLabs. 2019. Available online: https://www.cablelabs.com/blog/5g-link-aggregation-mptcp (accessed on 1 January 2022).
7. 5G & Wi-Fi: From Coexistence to Convergence. Tessares. Available online: https://www.tessares.net/solutions/5g-atsss-solution/ (accessed on 1 January 2022).
8. Bonaventure, O. The First Multipath TCP Enabled Smartphones. Multipath-TCP. 2018. Available online: http://blog.multipath-tcp.org/blog/html/2018/12/10/the_first_multipath_tcp_enabled_smartphones.html (accessed on 1 January 2022).
9. Bonaventure, O. Important Milestone for Multipath TCP. Tessares. 10 April 2020. Available online: https://www.tessares.net/important-milestone-for-multipath-tcp/ (accessed on 1 January 2022).
10. Opening the Way to 4G/5G Wi-Fi Convergence (NEW). Tessares. 2020. Available online: https://www.tessares.net/new-white-paper-opening-the-way-to-5g-convergence-september-2020/ (accessed on 1 January 2022).
11. Use Multipath TCP to Create Backup Connections for IOS. Apple. 2017. Available online: https://support.apple.com/en-in/HT201373 (accessed on 1 January 2022).
12. Detal, G.; Barré, S.; Peirens, B.; Bonaventure, O. *Leveraging Multipath TCP to Create Hybrid Access Networks*; ACM SIGCOMM: Los Angeles, CA, USA, 2017. Available online: https://conferences.sigcomm.org/sigcomm/2017/files/program-industrial-demos/sigcomm17industrialdemos-paper4.pdf (accessed on 1 January 2022).
13. Chao, L.; Wu, C.; Yoshinaga, T.; Bao, W.; Ji, Y. A Brief Review of Multipath TCP for Vehicular Networks. *Sensors* **2021**, *21*, 2793. [CrossRef]
14. Bonaventure, O. Multipath TCP in the Datacenter. 2018. Available online: http://blog.multipath-tcp.org/blog/html/2018/12/11/multipath_tcp_in_the_datacenter.html (accessed on 1 January 2022).
15. Zhao, Q.; Du, P.; Mena, J.; Gerla, M. A Multi-path TCP Solution for Software-Defined Military Heterogeneous Network. In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), Barcelona, Spain, 15–18 April 2018.
16. Zhao, Q.; Du, P.; Mena, J.; Gerla, M. Software Defined Multi-Path TCP Solution for Mobile Wireless Tactical Networks. In Proceedings of the 2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018.
17. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O.; Paasch, C. TCP Extensions for Multipath Operation with Multiple Addresses RFC6824(if approved); draft-ietf-mptcp-rfc6824bis-12. 2018. Available online: https://datatracker.ietf.org/doc/pdf/draft-ietf-mptcp-rfc6824bis-12 (accessed on 1 January 2022).
18. Hamza, A.; Lali, M.I.; Javid, F.; Din, M.U. Study of MPTCP with Transport Layer Security. In Proceedings of the 3rd International Conference on Engineering & Emerging Technologies (ICEET), Superior University, Lahore, Pakistan, 7–8 April 2016.
19. Bittau, A.; Boneh, D.; Hamburg, M.; Handley, M.; Mazieres, D.; Slack, Q. Cryptographic Protection of TCP Streams (Tcpcrypt); Internet-Draft draft-ietf-tcpinc-tcpcrypt-03. 2014. Available online: https://datatracker.ietf.org/doc/html/rfc8548 (accessed on 1 January 2022).
20. Díez, J.; Bagnulo, M.; Valera, F.; Vidal, I. Security for multipath TCP: A constructive approach. *Int. J. Internet Protoc. Technol.* **2011**, *6*, 146–155. [CrossRef]
21. Krishnan, A.; Amritha, P.P.; Sethumadhavan, M. Sum Chain Based Approach against Session Hijacking in MPTCP. In Proceedings of the 7th International Conference on Advances in Computing & Communications, ICACC-2017, Cochin, India, 19–22 September 2017.
22. Melki, R.; Hussein, A.; Chehab, A. Enhancing Multipath TCP Security Through Software Defined Networking. In Proceedings of the 2019 Sixth International Conference on Software Defined Systems (SDS), Rome, Italy, 10–13 June 2019.
23. Noh, G.; Park, H.; Roh, H.; Lee, W. Secure and Lightweight Subflow Establishment of Multipath-TCP. *IEEE Access* **2019**, *7*, 177438–177448. [CrossRef]
24. Kim, D.-Y.; Choi, H.-K. Efficient design for secure multipath TCP against eavesdropper in initial handshake. In Proceedings of the International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 19–21 October 2016.
25. Boneh, D.; Franklin, M. Identity-based encryption from the Weil pairing. *SIAM J. Comput.* **2003**, *32*, 586–615. [CrossRef]
26. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O. Extensions for Multipath Operation with Multiple Addresses; RFC 6824 TCP. 2013. Available online: https://datatracker.ietf.org/doc/html/rfc6824 (accessed on 1 January 2022).
27. Ford, A.; Raiciu, C.; Handley, M.; Bonaventure, O. *Architectural Guidelines for Multipath TCP Development (RFC6182)*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2011.
28. Popat, K.J.; Raval, J.; Johnson, S.; Patel, B. Experimental Evaluation of Multipath TCP with MPI. In Proceedings of the Third International Symposium on Women in Computing and Informatics, Kochi, India, 10–13 August 2015.
29. Popat, K.; Kapadia, D.V. Recent Trends in Security Threats in Multi-Homing Transport Layer Solutions. *Int. J. Adv. Sci. Technol.* **2020**, *29*, 5641–5648.
30. Raiciu, C.; Handley, M.; Wischik, D. *Coupled Congestion Control for Multipath Transport Protocols (RFC 6356)*; Internet Engineering Task Force (IETF): Fremont, CA, USA, 2011.
31. Khalili, R.; Gast, N.; Popovic, M. Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP. 2013. Available online: https://datatracker.ietf.org/doc/html/draft-khalili-mptcp-congestion-control-05 (accessed on 1 January 2022).
32. Cao, Y.; Xu, M.; Fu, X. Delay-based congestion control for multipath TCP. In Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, TX, USA, 30 October–2 November 2012.
33. Chaturvedi, R.K.; Chand, S. Multipath TCP security over different attacks. *Trans. Emerg. Telecommun. Technol.* **2020**, *31*, 4081. [CrossRef]

34. Bagnulo, M.; Paasch, C.; Gont, F.; Bonaventure, O.; Raiciu, C. Analysis of Residual Threats and Possible Fixes for Multipath TCP (Mptcp); (No. RFC 7430). 2015. Available online: https://www.rfc-editor.org/rfc/pdfrfc/rfc7430.txt.pdf (accessed on 1 January 2022).
35. Demaria, F. Security Evaluation of Multipath TCP, Analyzing and Fixing Multipath TCP Vulnerabilities, Contributing to the Linux Kernel Implementation of the New Version of the Protocol. Ph.D. Thesis, Master of Science in Engineering-Information and Communication Technology. KTH Royal Institute of Technology, Stockholm, Sweden, March 2016.
36. Munir, A.; Qian, Z.; Shafiq, Z.; Liu, A.; Le, F. Multipath TCP traffic diversion attacks and countermeasures. In Proceedings of the IEEE 25th International Conference on Network Protocols (ICNP), Toronto, ON, Canada, 10–13 October 2017.
37. Shafiq, M.Z.; Le, F.; Srivatsa, M.; Liu, A.X. Cross-path inference attacks on multipath tcp. In Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks, College Park, MA, USA, 21–22 November 2013; Association for Computing Machinery: New York, NY, USA, 2013.
38. Kumar, V.A.; Das, D.; Senior Member IEEE. Data sequence signal manipulation in multipath TCP (MPTCP): The vulnerability, attack and its detection. *Comput. Secur.* **2021**, *103*, 102180. [CrossRef]
39. Jadin, M.; Tihon, G.; Pereira, O.; Bonaventure, O. Securing multipath TCP: Design & implementation. In Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017.
40. Bonaventure, O. MPTLS: Making TLS and Multipath TCP Stronger Together. 2015. Available online: https://datatracker.ietf.org/doc/html/draft-bonaventure-mptcp-tls-00 (accessed on 1 January 2022).
41. Kato, T.; Cheng, S.; Yamamoto, R.; Ohzahata, S.; Suzuki, N. Protecting Eavesdropping over Multipath TCP Communication Based on Not-Every-Not-Any Protection. In Proceedings of the SECURWARE 2017: The Eleventh International Conference on Emerging Security Information, Systems and Technologies, Rome, Italy, 10–14 September 2017.
42. Pandya, V.; Saiyed, A.; Patel, K. Recent Advancement in Fine-Grained Access Control and Secure Data Sharing Scheme for Distributed Environment. In *Emerging Technologies for Computing, Communication and Smart Cities*; Springer: Singapore, 2022.
43. Mallouli, F.; Hellal, A.; Saeed, N.S.; Alzahrani, F.A. A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms. In Proceedings of the 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), Paris, France, 21–23 June 2019.
44. Stallings, W. *Cryptography and Network Security, 4/E*; Pearson Education Inc.: Upper Saddle River, NJ, USA, 2006.
45. Forouzan, B.; Mukhopadhyay, A.D. *Cryptography and Network Security*; Mc Graw Hill Education (India) Private Limited: New York, NY, USA, 2015.
46. Bagnulo, M. Secure MPTCP, Draft-Bagnulo-Mptcp-Secure-00; ietf-bagnulo-mptcp-secure-00. 2014. Available online: https://datatracker.ietf.org/doc/html/draft-bagnulo-mptcp-secure-00 (accessed on 1 January 2022).
47. Aasch, C.A.O.B. Securing the MultiPath TCP Handshake with External Keys; Work in Progress; draft-paasch-mptcp-ssl-00. 2012. Available online: https://tools.ietf.org/id/draft-paasch-mptcp-ssl-00.html (accessed on 1 January 2022).
48. Elliptic-Curve Cryptography. Available online: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography (accessed on 1 April 2022).