Security Enhancement of Multipath TCP by using off-path active attack prevention

A SYNOPSIS

Submitted in partial fulfillment of the
requirements for the award of the degree
of
DOCTOR OF PHILOSOPHY
in
COMPUTER SCIENCE & ENGINEERING

By
POPAT KHUSHI JAGDISHBHAI
Under Guidance of
DR. VIRAL V. KAPADIA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF TECHNOLOGY & ENGINEERING
THE MAHARAJA SAYAJIRAO UNIVERSITY OF BARODA
VADODARA-390002 (INDIA)
AUGUST 2022

ABSTRACT

With the advancement in communication technologies and Industry 4.0, the usage of computing devices has increased in day-to-day life in the field of communication, entertainment, education, healthcare, etc. for quality lifestyle which has increased the demand of low latency and higher resilient communication technology. Moreover, the most of the state-of-art devices such as laptops, wireless devices, sensors, etc. have feature to get connected with multiple networks through various network interfaces like Ethernet, Wi-Fi, etc. 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.

Multipath TCP (MPTCP) is a bidirectional byte stream transport layer protocol introduced by Internet Engineering Task force (IETF) which provides numerous benefits such as higher throughput, reliability, fault tolerance, backward compatibility and load balancing by supporting multi-homing that allows use of multiple paths for data transfer over single network connection. 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). 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. Many solutions to prevent the mentioned attacks have been proposed by researcher but some of them are vulnerable to other category of attack and others affects the overall performance of MPTCP.

The proposed research focuses 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. In order to exploit the ADD_ADDR vulnerability to launch the session hijacking attack has been configured over MPTCP network and demonstrated the rate of information lost due to the attack. The secure key exchange model for multipath TCP (SKEXMTCP) is proposed which uses identity based encryption (IBE) to encrypt the keys exchanged during the initial handshake to avoid the use of certificate authority to exchange the key pairs ahead of the communication. 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). The experimental

evaluation along with the security evaluation and performance evaluation of the proposed model has been carried out to compare the security of proposed model with existing security solutions.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

4G	Fourth Generation
5G	Fifth Generation
API	Application Programming Interface
CA	Certificate Authority
DoS	Denial of Service
ECC	Elliptic Curve Cryptosystem
GSM	Global System Mobile Communication
IBE	Identity Based Encryption
ID_{Alice}	ID used as a Public key of Alice
ID_{Bob}	ID used as a Public key of Bob
IETF	Internet Engineering Task Force
IP	Internet Protocol
LTE	Long-term Evolution
MitM	Man in the middle
MPTCP	Multipath Transmission Control Protocol
MPTCPsec	MPTCP Secure
MPTLS	Multipath Transport Layer Security
PKG	Private Key Generator
PR_{Alice}	Private Key of Alice
PR_{Alice_Master}	Shared key used to generate the private key of Alice
PR_{Bob}	Private Key of Bob
PR_{Bob_master}	Shared key used to generate the private key of Bob
PU_{Alice}	Public key of Alice
PU_{Bob}	Public key of Bob
QoS	Quality of Service
SCMTCP	Secure Connection Multipath TCP
SDN	Software Defined Network
SKEXMTCP	Secure Key Exchange model for MPTCP
SMTCP	Secure Multipath TCP
TCP	Transmission Control Protocol

TLS	Transport Layer Security
VM	Virtual Machine

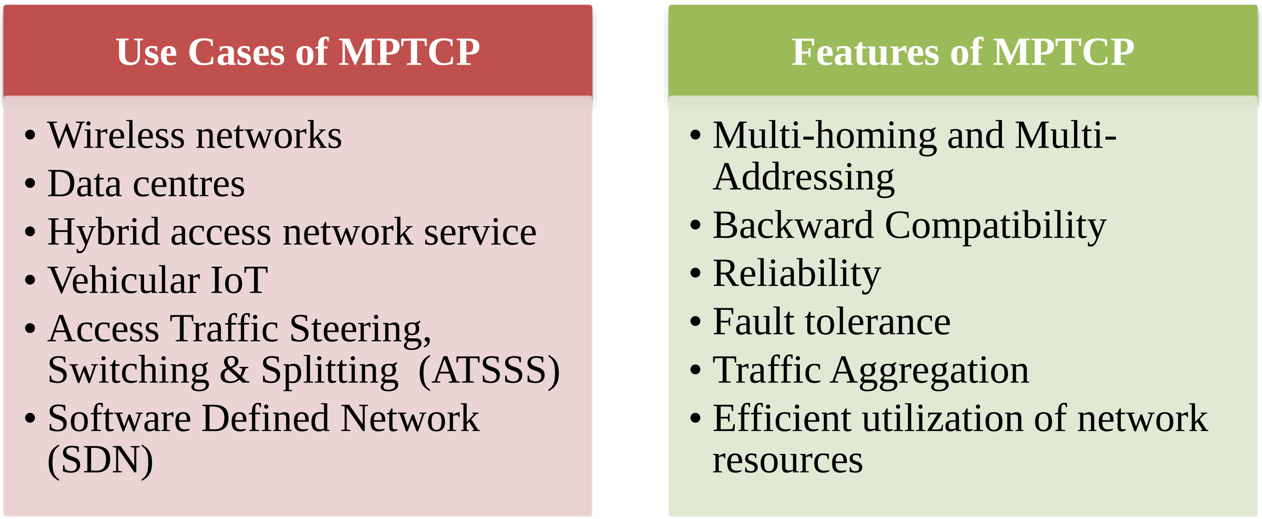
1. INTRODUCTION

This chapter briefly introduces about Multipath Transmission Control Protocol (MPTCP), its working and vulnerabilities of MPTCP using which attackers can generate the threat. It also tells the motive behind this work, the problem statement and what are the research contributions of this work.

1.1. Multipath TCP and Its Security Issues

In the era of an Industry 4.0, the internet of things, big data analytics, blockchain technology and artificial intelligence have empowered a variety of industrial sectors and started focusing heavily on the performance and Quality of Services of real time applications. The revolution of industries have changed the day-to-day lifestyle of humans across the word by offering the services anywhere-anytime using communication technologies on various computing devices which increases the demand of high performance and reliable communication technologies. Moreover, most of the computing devices are multi-homed as connected with more than one networks and multi-addressed by equipped with multiple network interfaces like Wi-Fi, GSM, Ethernet, etc. Transmission Control Protocol (TCP) [1] is the best suited transport layer protocol used by most of the state-of-art applications for reliable communication but it restricts connection to use of single network interface by binding the IP addresses [2]. Moreover, if the host tries to switch to another network interface during on-going communication with the server, the current TCP connection will be dropped as device will get connected to another network and new IP address will be assigned to it.

In order to utilize the network resources efficiently and offer the reliable and high performance communication, MPTCP [3, 4, 5] has been implemented over TCP. Multipath TCP facilitate all the applications which uses TCP as a transport layer protocol with backward compatibility, fault tolerance with stable connection by smooth handover among multiple paths, and higher performance by bandwidth aggregation. MPTCP, an initiative of Internet Engineering Task Force (IETF), attracts various industries and academia to offer solutions in areas like mobile communication, vehicular networks, datacenter networks, robotics communication, software defined networks, etc. Many companies like Apple, Tessars, 3GPP, Samsung, Huawei, etc. are adopting MPTCP for performance improvement in terms of bandwidth and resiliency by combining usage of multiple network interfaces in mobile devices as well as in servers, datacenters, and end machines [6, 7, 8, 9, 10, 11] . Figure 1 shows the use cases and features of MPTCP.



Use Cases of MPTCP

- Wireless networks
- Data centres
- Hybrid access network service
- Vehicular IoT
- Access Traffic Steering, Switching & Splitting (ATSSS)
- Software Defined Network (SDN)

Features of MPTCP

- Multi-homing and Multi-Addressing
- Backward Compatibility
- Reliability
- Fault tolerance
- Traffic Aggregation
- Efficient utilization of network resources

Figure 1 Use Cases and Features of MPTCP [12, 13, 9, 14, 15]

MPTCP supports the multi-homing and multi-addressed nature of hosts for data transmission, which opens the door for security threats while availing the low latency communication over stable connection. MPTCP uses the TCP header to incur positive impact on the traditional TCP aware applications to achieve the goal of backward compatibility by using various options in TCP header like MP_CAPABLE, MP_JOIN, ADD_ADDR etc. but they are more vulnerable to various attacks than TCP [4, 16].

MPTCP uses 3-way handshake process for connection establishment. MP_CAPABLE option used in TCP header during the initial 3-way handshake for connection establishment indicates that the host supports the MPTCP connection. The MP_JOIN option is available to add another subflow with the pre-established connection. ADD_ADDR is another option available with MPTCP, which can be used to inform another host about availability of a new IP address.

These options of MPTCP make it vulnerable to various threats [17] by allowing an attacker to gain access to the MPTCP session. The session can be hijacked either by forging the keys communicated during the 3-way handshake or by adding the forged address using ADD_ADDR packets or by using MP_JOIN packets on communicating host. During the communication between Alice and Bob, the attacker may initiate the session hijacking attack and create the illusion that the Bob will assume that the new subflow will be established with legitimate user Alice only and Alice will think that request is coming from Bob but in backend the subflow will be established with attacker from both the ends and attacker will be successful in implementing man-in-the-middle attack. By using the compromised subflow, the attacker can monitor, manipulate or gain access over whole connection by terminating the legitimate subflow. Moreover, the key exchanged in plaintext during the initial handshake in MPTCP welcomes many security threats because these keys are used for sub flow authentication during the addition of new subflow over connection, or advertisement of availability of new network interface in the future, which leads to SYN flood attack, MP_JOIN attack, SYN/MP_JOIN attack, Session Hijacking, traffic diversion attack, etc.

The breach of security of one of the paths may lead to the hijacking of whole connection which will have a negative effect on the confidentiality, integrity and availability of communication or data.

1.2. Motivation for This Work

Currently, most of all the computing devices are available with multiple network interfaces, unlike TCP, MPTCP allows to utilize more than one network interface for communication between hosts to offer higher bandwidth, resiliency and backward compatibility. However, MPTCP solves the issues related to network resources utilization, backward compatibility and performance by using TCP header, it opens the doors for many threats.

- The keys exchanges in clear form during the initial handshake are being used in future for authentication of new subflow, advertisement of the new address, etc.
- The same keys can be used to exploit the vulnerability of MPTCP and initiate the attacks to hijack the session to gain access of information exchanged.
- Secure key exchange during initial handshake can resolve the security issues related to session hijacking using address advertisement, Denial of Service (DoS) attack, and authentication of host at the time of subflow adding over connection.
- The security enhancement of MPTCP may resolve the issues related to latency and connection drop and attract many industries like wireless communication, smart healthcare, datacenters, etc. to take the benefits of the features of MPTCP.

1.3. Problem Statement, Objectives, Research Contributions

1.3.1. Problem Statement

- To design and implement the solution which offers the security against session hijacking using ADD_ADDR packet sequence which falls under the category of off path active attack and eavesdropper in initial handshake by securing the key exchange during the initial handshake.

1.3.2. Objectives

- Identify the vulnerabilities of MPTCP and test the vulnerability during the communication to identify the impact of the same on communication.
- Identify and test the off path active attacks on communication network.
- To design or develop some model for securing the MPTCP communication from off path active attackers in such a way that it doesn’t degrade the performance of MPTCP than TCP.
- Test the designed model against security and performance.

1.3.3. Research Contributions

- MPTCP security threats are examined, how ADD_ADDR packet can be used to launch session hijacking attack as well as the keys exchanged during initial handshake can be captured and used to initiate session hijacking is demonstrated and the analysis of the data lost due to session hijacking has been carried out.
- The existing solutions to the various security threats of MPTCP are analyzed in terms of security level and performance comparison.
- The performance and security of IBE is compared with the elliptic curve cryptosystem (ECC).
- SKEXMTCP using IBE, the model to exchange the key securely during initial handshake without exchanging keys prior to the connection establishment, is proposed and evaluated in terms of security and performance.

2. LITERATURE STUDY

MPTCP uses various options in TCP header like MP_CAPABLE, MP_JOIN, ADD_ADDR, MP_PRIO, etc. for backward compatibility. These options of MPTCP make it vulnerable to various threats by allowing an attacker to gain access to the MPTCP session which will affect the confidentiality, integrity and availability of communication. Many researchers are actively working on the security of MPTCP in order to increase the adoption of MPTCP in various areas to achieve efficient utilization of network resources, higher performance and resiliency but still security of the MPTCP is an open issue.

2.1. Security Threats of MPTCP

MPTCP offers backward compatibility to the TCP aware application by using the TCP header for the communication. The various options in TCP header like MP_CAPABLE, MP_JOIN, ADD_ADDR, MP_PRIO, etc. are used to share the information related to MPTCP. These options used by MPTCP opens the new doors for attackers to initiate the attacks using various threats [17]. 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 on MPTCP Linux kernel implementation figured out by IETF in their draft [17] are ADD_ADDR attack, DoS attack on MP_JOIN, SYN flooding amplification, and eavesdropper in initial handshake, SYN/JOIN attacks. Other than these attacks, traffic diversion attacks are another category of attacks which can be implemented by exploiting vulnerability of MPTCP option MP_PRIO [18]. The significant threats to MPTCP are as shown in Table 1.

Table 1 Significant threats to MPTCP

Attack	Category *	Active/ Passive	Referen ces	Security Goals Impacted #	Remarks
Eavesdropper in initial handshake	P	Active	[17]	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	[17]	C, I, A	By packet forging, an attacker can send the spoofed packet to

Attack	Category *	Active/ Passive	Referen ces	Security Goals Impacted #	Remarks
					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.
DoS attack on MP_JOIN	F	Active	[17]	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	[17]	A	By using the SYN packet, the server will be exhausted; thus, the client will not be served.
Traffic diversion attack	F	Active	[18]	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	[19]	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	[17]	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	[20]	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.

2.2. Session hijacking attack using ADD_ADDR option

ADD_ADDR is the option available with MPTCP used to inform communicating host 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) [16] 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 as shown in Figure 4.

Figures 2–4 show the packet exchange scenario for connection establishment and advertisement of new IP address.

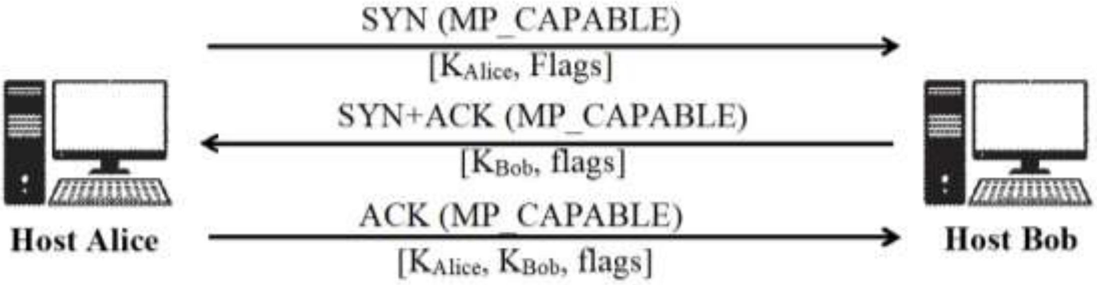


Figure 2 MPTCP Options in Version (v0) Connection Establishment with MP_CAPABLE.

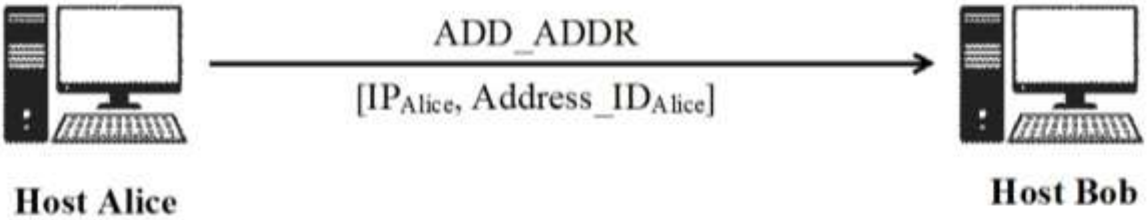


Figure 3 MPTCP Options in Version (v0) advertising new addresses with ADDR.

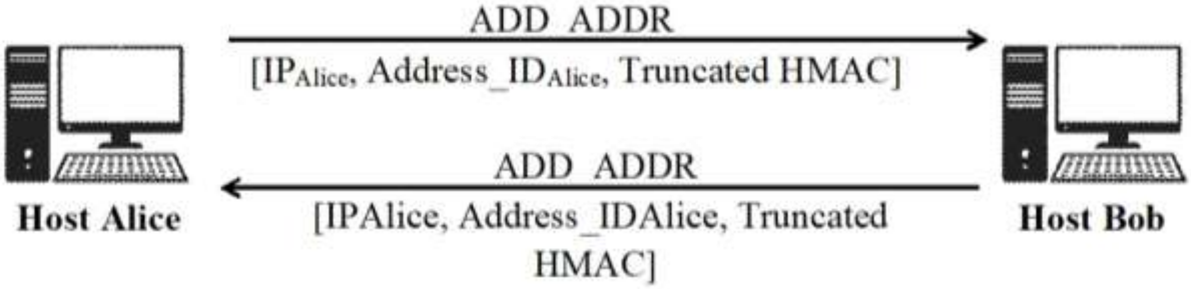

ADD_ADDR
[IP_{Alice}, Address_ID_{Alice}, Truncated HMAC]
ADD_ADDR
[IPAlice, Address_IDAlice, Truncated HMAC]
Host Alice
Host Bob

Figure 4 Advertisement of the new IP address with ADD_ADDR in MPTCP Version (v1).

The steps to exploit the ADD_ADDR vulnerability to initiate the attack to hijack the connection are demonstrated in Figure 5. The session hijacking using ADD_ADDR attack 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 3. 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 5 [2].

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 sub-flow will be established between Alice and Eve. Now, the actual situation and illusionary scenario is represented in Figure 6. Eve can change the priority of the subflow by sending the MP_PRIO packet to hijack the whole session.

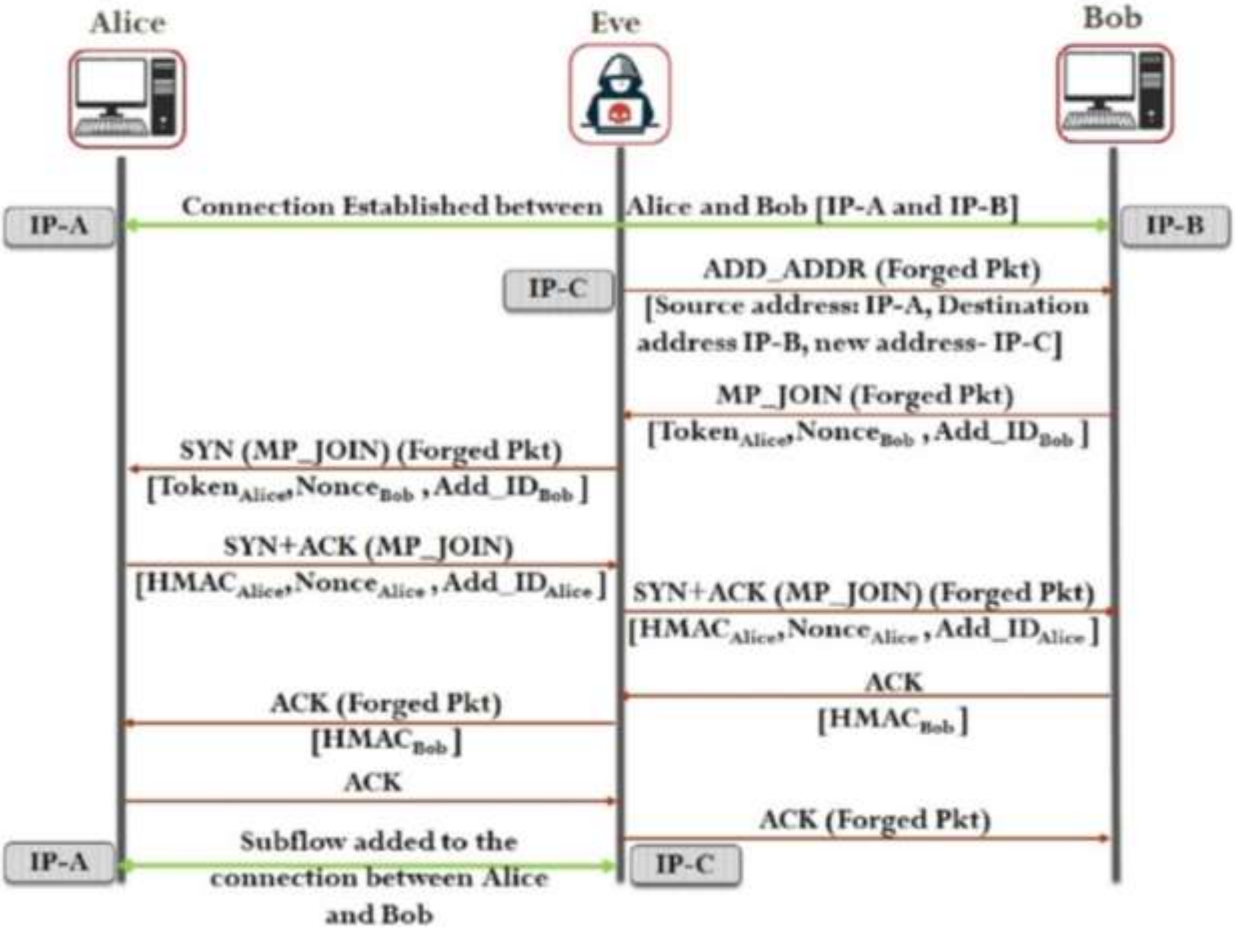

Alice
Eve
Bob
IP-A
Connection Established between Alice and Bob [IP-A and IP-B]
IP-B
IP-C
ADD_ADDR (Forged Pkt)
[Source address: IP-A, Destination address IP-B, new address- IP-C]
MP_JOIN (Forged Pkt)
[Token$_{Alice}$,Nonce$_{Bob}$,Add_ID$_{Bob}$]
SYN (MP_JOIN) (Forged Pkt)
[Token$_{Alice}$,Nonce$_{Bob}$,Add_ID$_{Bob}$]
SYN+ACK (MP_JOIN)
[HMAC$_{Alice}$,Nonce$_{Alice}$,Add_ID$_{Alice}$]
SYN+ACK (MP_JOIN) (Forged Pkt)
[HMAC$_{Alice}$,Nonce$_{Alice}$,Add_ID$_{Alice}$]
ACK
[HMAC$_{Bob}$]
ACK (Forged Pkt)
[HMAC$_{Bob}$]
ACK
ACK (Forged Pkt)
IP-A
Subflow added to the connection between Alice and Bob
IP-C

Figure 5 Use of ADD_ADDR vulnerability to initiate the attack to compromise the connection.

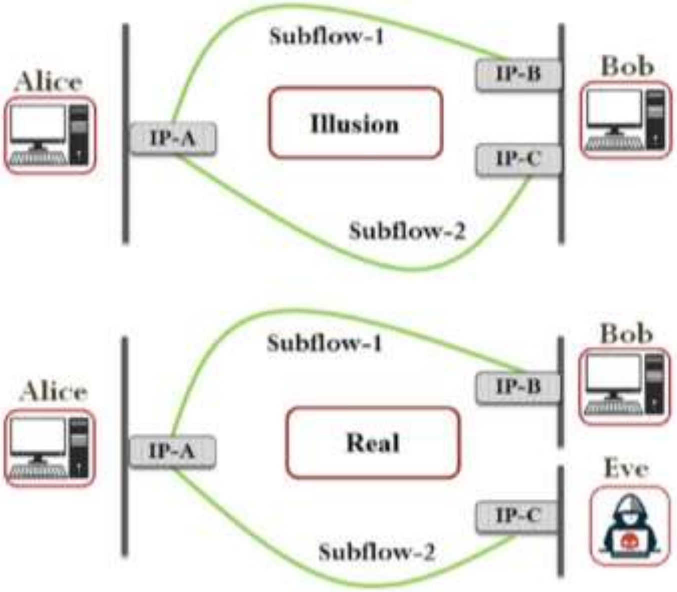


Figure 6 Real vs. illusion for Alice and Bob during the session hijacking attack.

The Experimental implementation of the session hijacking by Man in the Middle attack (MitM) is performed on Linux kernel implementation of MPTCP v0 [3] and Eavesdropper in initial handshake is performed on MPTCP v1. The experimental setup with required topology was created as shown in below Figure 7 using Oracle virtualBox. The simulation of the attack [21] is performed using two virtual machines with Linux kernel implementation of MPTCP. The use of virtualBox leads to fast experimental setup, reliability of experiments and no risk of damaging or crashing of kernel. To configure this "client server" scenario, the required custom kernel and tools are available on official website of MPTCP (http://www.multipath-tcp.org). In order to initiate both the attacks, the attacker host is simulated with scapy tool which supports MPTCP that is used for capturing and injecting packet on the network. Extended version of scapy specially designed for MPTCP [22] is available by Nicolas Matre on https://github.com/nimai/mptcp-scapy repository. Scapy tool provides functionalities for sniffing, modification, capturing and matching the request-response which is important for initiating an attack.

Here in Figure 7, client VM, server VM and host machine are configured with custom MPTCP Linux Kernel. Here client must be equipped with more than one network interfaces to setup MPTCP scenario while server can be equipped with one or more network interface. Here, host machine is configured with three tap interfaces (virtual network interfaces) to implement multi-homing environment [21].

Session hijacking experiment was performed successful in shown scenario for chat application and file transfer application developed using JAVA socket API on MPTCP Linux kernel v0. Here for gathering prerequisite information such as source-destination IP addresses, SEQ no, ACK no etc. average 2-3 packets need to be captured. Average 6:3 packets are required for client to server to initiate attack successfully. Success rate of session hijacking attack is 77%.

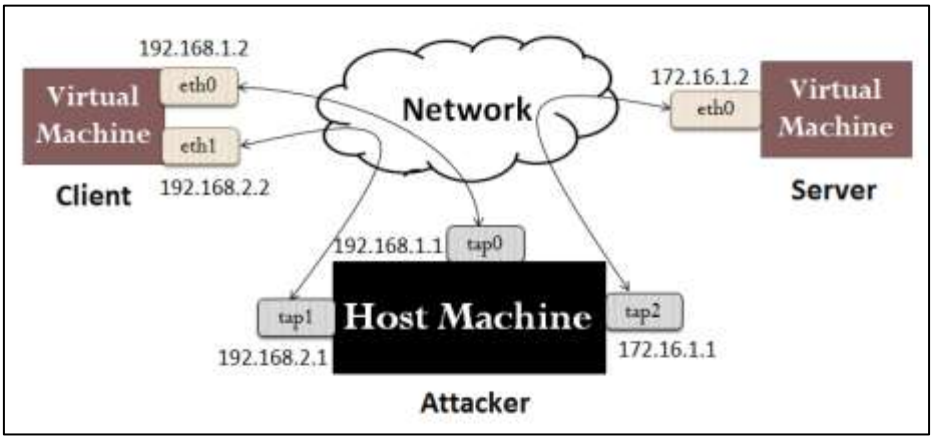


Figure 7 Network simulations for experiments [16]

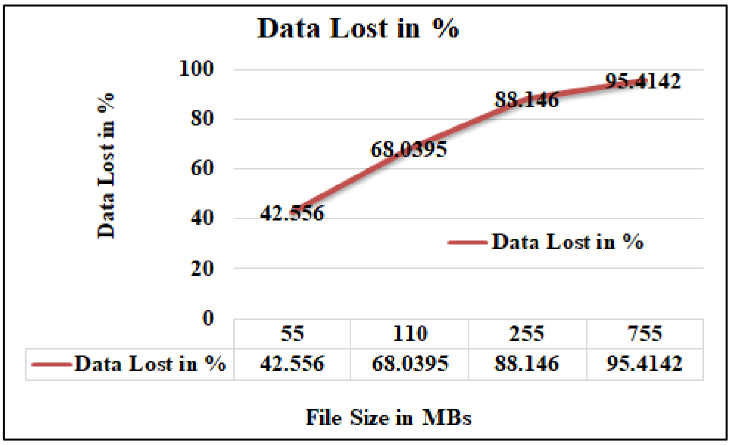


Figure 8 Data lost analysis during different sized file transfer with hijacking attack

The attack is performed multiple times for capturing different data formats as well as for different file sizes. The above Figure 8 and 9 shows the data lost in percentage (%) while transferring different size of file and different types of file from client to server respectively.

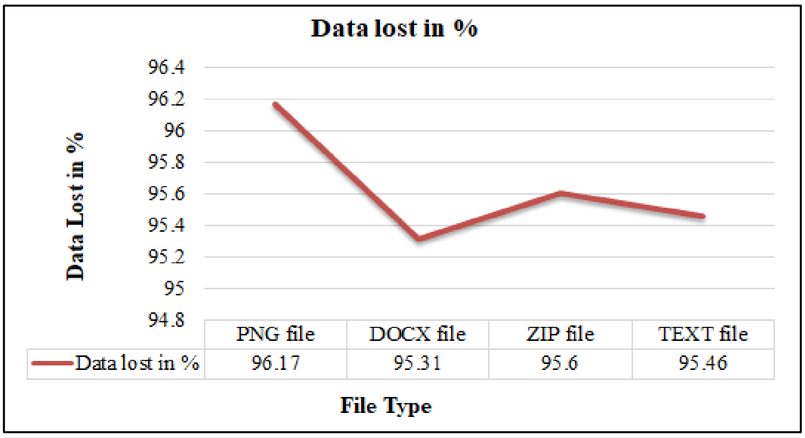
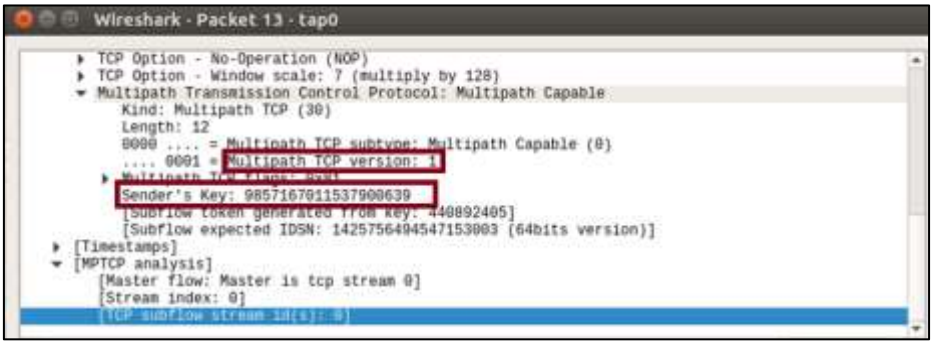
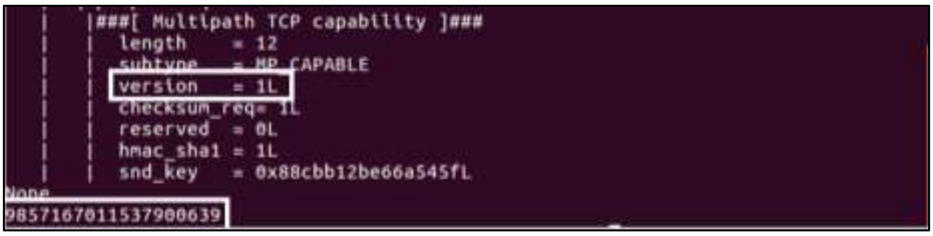


Figure 9 Data lost analysis in % during file transfer of different format



```
  ▸ TCP Option - No-Operation (NOP)
  ▸ TCP Option - Window scale: 7 (multiply by 128)
  ▾ Multipath Transmission Control Protocol: Multipath Capable
      Kind: Multipath TCP (30)
      Length: 12
      0000 .... = Multipath TCP subtype: Multipath Capable (0)
      .... 0001 = Multipath TCP version: 1
    ▸ Multipath TCP flags: 0x81
      Sender's Key: 9857167011537900639
      [Subflow token generated from key: 440892405]
      [Subflow expected IDSN: 14257564945471530003 (64bits version)]
▸ [Timestamps]
▾ [MPTCP analysis]
    [Master flow: Master is tcp stream 0]
    [Stream index: 0]
    [TCP subflow stream id(s): 0]
```

Figure 10 Wireshark capture for Eavesdropper at initial handshake capturing keys in clear form



```
   |    |###[ Multipath TCP capability ]###
   |    |   length    = 12
   |    |   subtype   = MP_CAPABLE
   |    |   version   = 1L
   |    |   checksum_req= 1L
   |    |   reserved  = 0L
   |    |   hmac_sha1 = 1L
   |    |   snd_key   = 0x88cbb12be66a545fL
None
9857167011537900639
```

Figure 11 Extracting keys captured during initial handshake to perform ADD_ADDR attack by using python script

2.3. Probable Security Solution for 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, etc. can be used to solve the issue related to the key exchange during the initial handshake, but MPTCP uses the TCP header in which only 64 bits can be occupied for key. In this section, the comparison between the available solutions is covered to show the re-search path in the area of MPTCP security.

The hash chain-based solution proposed by [23] 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 [24] 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 [25] 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 [26] 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 of computation power.

In [27], 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 [28], 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 [16] 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 hand-shake can initiate the session hijacking by calculating the HMAC for the authentication using the keys exchanged during the initial handshake.

Table 2 [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 2 Existing security solutions for MPTCP

Reference	Year	Solution	Remarks
[23]	2011	Hash chain-based solution	It does not offer security against on-path active attackers.
[25]	2014	Tcpcrypt	It does not authenticate the public key and is vulnerable to man-in-the middle attacks.
[29]	2016	Multipath Transport Layer Security (MPTLS)	Computation overhead during initial handshake. Need to modify the packet sequence.
[26]	2015		
[28]	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.
[24]	2017	Sum chain-based solution	Vulnerable to time-shifted attack.
[30]	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.
[16]	2018	ADD_ADDR2	Vulnerable to time-shifted attack.
[14]	2019	Key exchange through SDN	Single point of failure.
[31]	2020	Secure connection Multipath TCP (SCMTCP)	For each new connection request, it generates the unique key for each option, which increases the computational overhead.
[32]	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. PROPOSED WORK

3.1. 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 ADDR_ADDR attack and eavesdroppers in the initial handshake during the communication over MPTCP, the SKEXMTCP is proposed which uses the identity-based encryption scheme to exchange the security parameters of 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).

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.

3.1.1. 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 it-self 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}.

3.1.2. 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 12 shows the packet sequence and the parameters passed with each packet of the initial three-way handshake according to the proposed scheme.

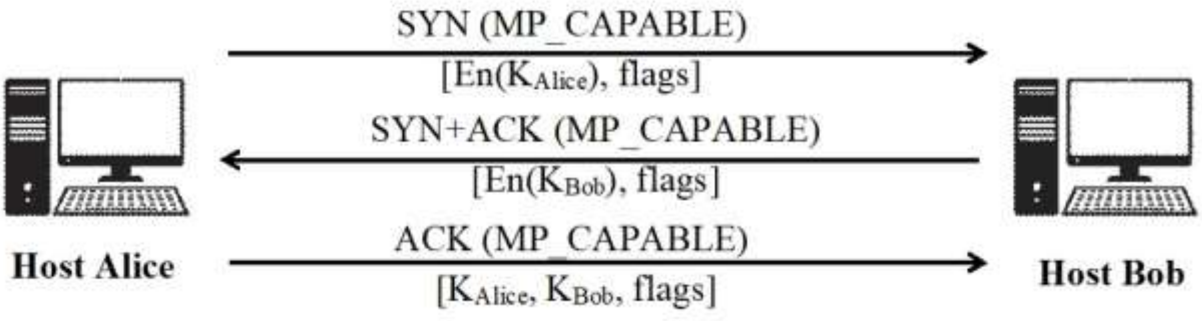


Figure 12 Three-way handshake with proposed scheme.

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 13.

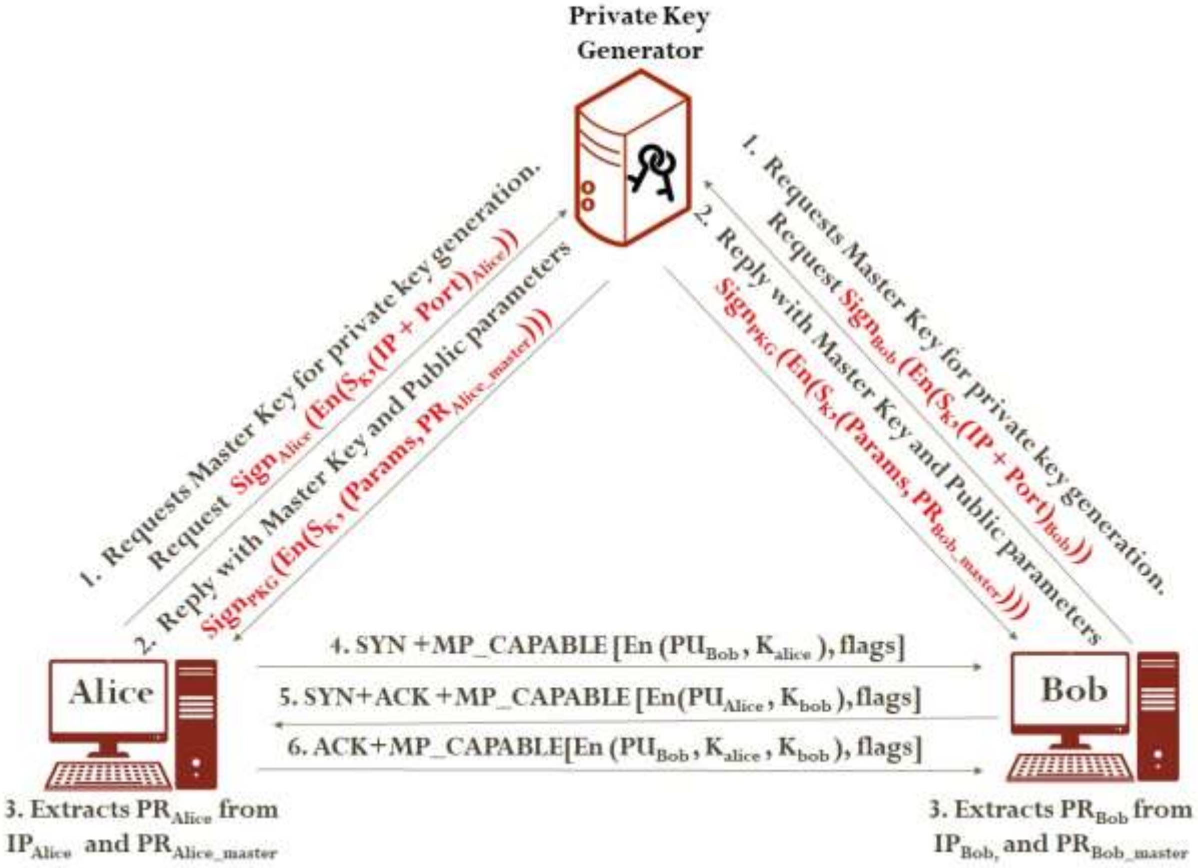

Private Key Generator
1. Requests Master Key for private key generation.
Request $Sign_{Alice}$ (En(S_K,(IP + Port)$_{Alice}$))
2. Reply with Master Key and Public parameters
$Sign_{PKG}$(En(S_K, (Params, PR_{Alice_master})))
1. Requests Master Key for private key generation.
Request $Sign_{Bob}$ (En(S_K,(IP + Port)$_{Bob}$))
2. Reply with Master Key and Public parameters
$Sign_{PKG}$ (En(S_K,(Params, PR_{Bob_master})))
Alice
Bob
4. SYN +MP_CAPABLE [En (PU_{Bob}, K_{alice}), flags]
5. SYN+ACK +MP_CAPABLE [En(PU_{Alice}, K_{bob}),flags]
6. ACK+MP_CAPABLE[En (PU_{Bob}, K_{alice}, K_{bob}), flags]
3. Extracts PR_{Alice} from IP_{Alice} and PR_{Alice_master}
3. Extracts PR_{Bob} from $IP_{Bob,}$ and PR_{Bob_master}

Figure 13 Secure key exchange during a three-way handshake using IBE.

4. TESTING AND RESULT ANALYSIS

The proposed work has been evaluated in terms of performance and security. In this section, the proposed work is compared with existing solutions in terms of performance and security.

4.1. Experimental Evaluation of Proposed Model

The proposed scheme is tested with MPTCP using the Linux kernel implementation of MPTCP. The Oracle Virtual Boxes are used to set up the scenario of the proposed scheme by creating two virtual machines (VMs), client and server, as shown in Figure 14. 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.

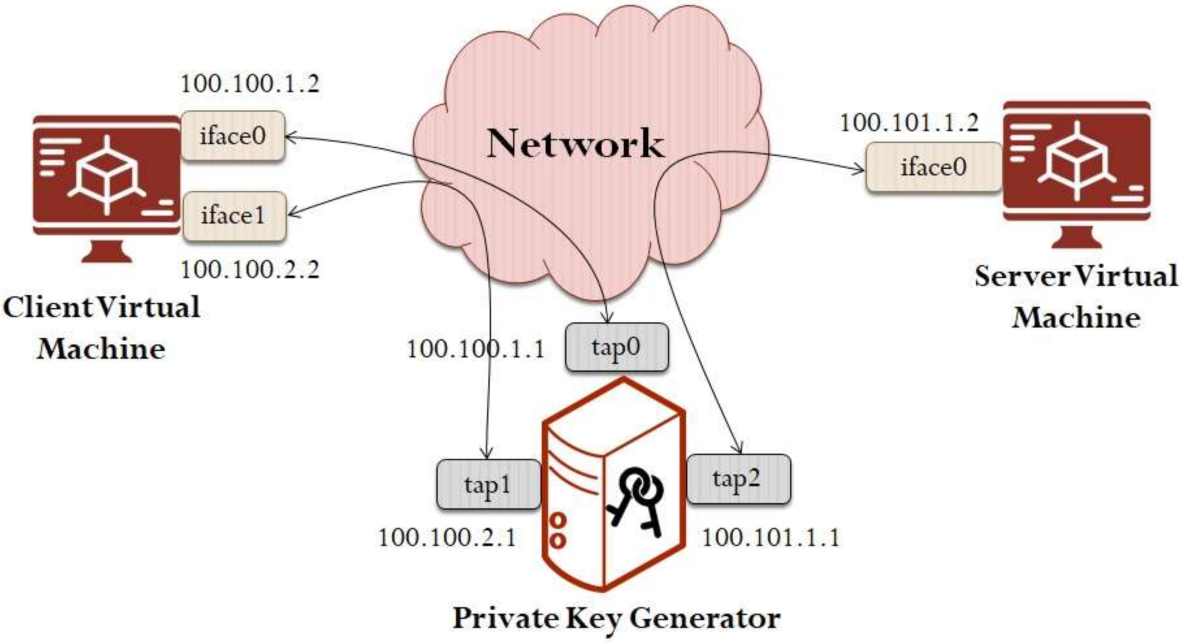


Figure 14 Experimental setup for testing the proposed work

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 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.

4.2. 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 3 shows the comparison of whether the various solutions offer security against various attacks or not.

Table 3 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.

4.3. 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.

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 *O(N)= O(N1),* then

$$N1 \ll N1 \times N1$$

Table 4 shows the comparative analysis in terms of the number of bytes required for key exchange and delay for packet exchanges [32, 28] 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. 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 4 Comparative evaluation of existing security solutions.

	Proposed Solution [33]	**SCMTCP** [31]	**Secure and Lightweight Subflow Scheme** [32]	**SMPTCP** [34]	**MPTLS** [35]	**Hash Chain** [23]	**MPTCP** [16]
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

	Proposed Solution [33]	**SCMTCP** [31]	**Secure and Lightweight Subflow Scheme** [32]	**SMPTCP** [34]	**MPTLS** [35]	**Hash Chain** [23]	**MPTCP** [16]
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

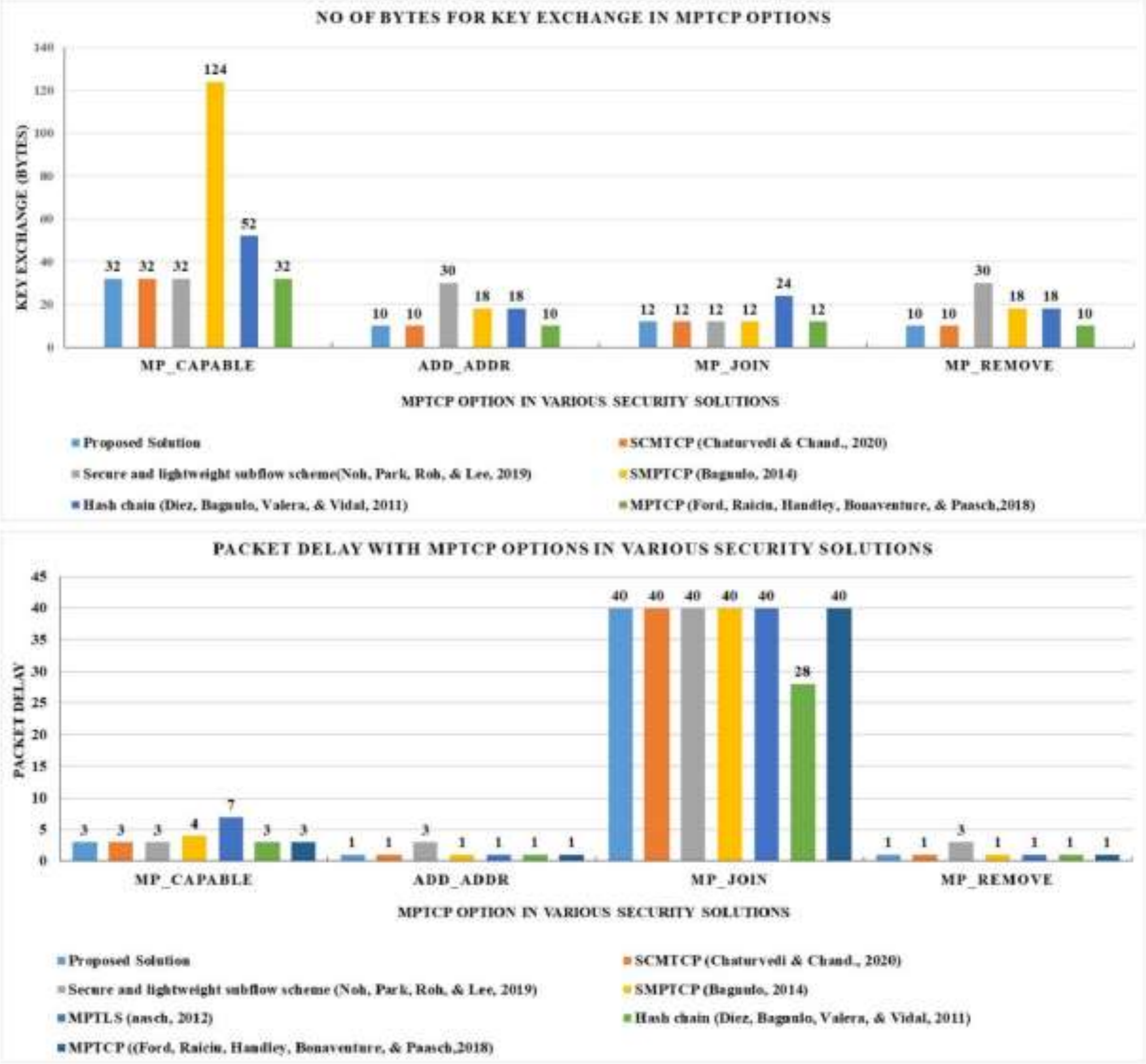


Figure 15 Comparative study of bytes required in key exchange with various MPTCP options in security solutions [16, 31, 32, 34, 35, 23].

5. CONCLUSION AND FUTURE DIRECTION

5.1. Conclusion

The traditional transport layer protocol TCP doesn't fulfill the requirements of current network scenario so multipath TCP is the best suitable transport layer protocol which can be deployed easily due to its backward compatibility but security is one the most crucial requirement for today's era. In the proposed research work, the ADD_ADDR option can be used to launch the session hijacking attack has been demonstrated and it can be analyzed that more than 90% of data can be compromised by changing the priority of legitimate subflow and redirecting the whole data on compromised path. Moreover, the keys exchanged during the initial handshake can be recovered by eavesdropper and the same can be used in future to launch off path active attacks. In order to solve the mentioned issue, we have proposed a secure key exchange model for MPTCP which uses identity based encryption (IBE) to encrypt the session keys exchanged during the 3-way handshake. The use of identity based encryption avoids the requirement of certificate authority & key exchange in prior to the communication which is beneficial for MPTCP in terms of performance and security as well. The use of public key cryptosystem for key encryption increases the overhead on the 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 and it shows that the proposed solution enhancing the security of MPTCP and does not create any overhead on the existing protocol. The secure key exchange during initial handshake prevents the ADD_ADDR attacks by using HMAC of the keys to authenticate the packet.

5.2. Road Map for Future Work

The proposed system can be extended by proposing the algorithm to generate the unique identity using IP address and port pair which can make the proposed system more complex. The light-weight encryption techniques can be proposed and tested in future with MPTCP for resource constraint environment. Moreover, the sessions can be secured by setting the priority of each subflow of the connection, the model can be trained to detect the compromised subflow.

6. REFERENCES

[1] J. Postel, "Transmission Control Protocol, RFC 793," September 1981.

[2] K. Popat and V. V. Kapadia, "Multipath TCP Security Issues, Challenges and Solutions," in *Information, Communication and Computing Technology.Communications in Computer and Information Science*, 2021.

[3] S. Barré, C. Paasch and O. Bonaventure., "Multipath TCP: from theory to practice," in *International Conference on Research in Networking.*, Berlin, Heidelberg, 2011.

[4] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, "RFC 6824 TCP Extensions for Multipath Operation with Multiple Addresses," 2013.

[5] A. Ford, C. Raiciu, M. Handley and O. Bonaventure, "Architectural Guidelines for Multipath TCP Development (RFC6182)," Internet Engineering Task Force (IETF), March 2011.

[6] "5G & Wi-Fi: from coexistence to convergence," tessares.

[7] O. Bonaventure, "Important Milestone for Multipath TCP," Tessares, 10 April 2020. [Online]. Available: https://www.tessares.net/important-milestone-for-multipath-tcp/. [Accessed 1 1 2022].

[8] O. Bonaventure, "The first Multipath TCP enabled smartphones," Multipath-TCP, 10 12 2018. [Online]. Available: http://blog.multipath-tcp.org/blog/html/2018/12/10/the_first_multipath_tcp_enabled_smartphones.html. [Accessed 1 1 2022].

[9] O. Dharmadhikari, "5G Link Aggregation with Multipath TCP (MPTCP)," CableLabs Logo, 2019.

[10] "Opening the way to 4G / 5G Wi-Fi Convergence (NEW)," tessares, 2020.

[11] "Use Multipath TCP to create backup connections for iOS," Apple, 2017.

[12] L. Chao, C. Wu, T. Yoshinaga, W. Bao and Y. Ji., "A Brief Review of Multipath TCP for Vehicular Networks," *Sensors,* vol. 21, no. 8, p. 2793, 2021.

[13] O. Bonaventure, "Multipath TCP in the datacenter," 11 12 2018. [Online]. Available: http://blog.multipath-tcp.org/blog/html/2018/12/11/multipath_tcp_in_the_datacenter.html. [Accessed 1 1 2022].

[14] R. Melki, A. Hussein and A. Chehab, "Enhancing Multipath TCP Security Through Software Defined Networking," in *2019 Sixth International Conference on Software Defined Systems (SDS),* 2019.

[15] Q. Zhao, P. Du, J. Mena and M. Gerla, "A Multi-path TCP Solution for Software-Defined Military Heterogeneous Network," in *IEEE Wireless Communications and Networking Conference (WCNC),* 2018.

[16] A. Ford, C. Raiciu, M. Handley, O. Bonaventure and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses RFC6824(if approved)," draft-ietf-mptcp-rfc6824bis-12, 2018.

[17] M. Bagnulo, C. Paasch, F. Gont, O. Bonaventure and C. Raiciu, "Analysis of residual threats and possible fixes for multipath TCP (mptcp)," (No. RFC 7430), 2015.

[18] A. Munir, Z. Qian, Z. Shafiq, A. Liu and F. Le, "Multipath TCP traffic diversion attacks and countermeasures," in *IEEE 25th International Conference on Network Protocols (ICNP),* Toronto, ON, Canada, 2017.

[19] M. Z. Shafiq, F. Le, M. Srivatsa and A. X. Liu., "Cross-path inference attacks on multipath tcp," in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, 2013.

[20] V. A. Kumar, D. Das and S. M. IEEE., "Data sequence signal manipulation in multipath TCP (MPTCP): The vulnerability, attack and its detection.," *Computers & Security,* vol. 103, p. 102180, 2021.

[21] F. DEMARIA, "Security Evaluation of Multipath TCP, Analyzing and fixing Multipath TCP vulnerabilities, contributing to the Linux Kernel implementation of the new version of the protocol (PhD Thesis)," March 2016.

[22] O. Bonaventure, "Multipath TCP: An annotated bibliography.," *ICTEAM, UCL,* 2015.

[23] J. Díez, M. Bagnulo, F. Valera and I. Vidal, "Security for multipath TCP: a constructive approach.," *International Journal of Internet Protocol Technology,* vol. 6, no. (3), pp. 146-155., 2011.

[24] A. Krishnan, P. P. Amritha and M. Sethumadhavan, "Sum Chain Based Approach against Session Hijacking in MPTCP," in *7th International Conference on Advances in Computing & Communications, ICACC-2017*, Cochin, India, 2017.

[25] A. Bittau, D. Boneh, M. Hamburg, M. Handley, D. Mazieres and Q. Slack, "Cryptographic protection of TCP Streams (tcpcrypt)," Internet-Draft draft-ietf-tcpinc-tcpcrypt-03, 2014.

[26] O. Bonaventure, "MPTLS : Making TLS and Multipath TCP stronger together," 2015.

[27] M. Jadin, G. Tihon, O. Pereira and O. Bonaventure, "Securing multipath TCP: Design & implementation," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*.

[28] D.-Y. Kim and H.-K. Choi, "Efficient design for secure multipath TCP against eavesdropper in initial handshake," in *International Conference on Information and Communication Technology Convergence (ICTC),* Jeju, South Korea, 2016.

[29] A. Hamza, M. I. Lali, F. Javid and M. u. Din., "Study of MPTCP with Transport Layer Security," in *Proceedings of the 3rd International Conference on Engineering & Emerging Technologies (ICEET),* Superior University, Lahore, PK, 7-8 April, 2016.

[30] T. Kato, S. Cheng, R. Yamamoto, S. Ohzahata and N. Suzuki, "Protecting Eavesdropping over Multipath TCP Communication Based on Not-Every-Not-Any Protection," in *SECURWARE 2017 : The Eleventh International Conference on Emerging Security Information, Systems and Technologies*, 2017.

[31] R. K. Chaturvedi and S. Chand., "Multipath TCP security over different attacks.," *Transactions on Emerging Telecommunications Technologies,* vol. 31, no. 9, p. 4081, 2020.

[32] G. Noh, H. Park, H. Roh and W. Lee, "Secure and Lightweight Subflow Establishment of Multipath-TCP.," *IEEE Access,* vol. 7, pp. 177438-177448, 2019.

[33] A. AS, P. K, K. VV, Q. MRNM, A. N and M. REA, "Efficient Key Exchange Using Identity-Based Encryption in Multipath TCP Environment," *Applied Sciences,* vol. 12, no. 15, p. 7575, 2022.

[34] M. Bagnulo, "Secure MPTCP, draft-bagnulo-mptcp-secure-00," ietf-bagnulo-mptcp-secure-00, 2014.

[35] C. a. O. B. aasch, "Securing the MultiPath TCP handshake with external keys," Work in Progress, draft-paasch-mptcp-ssl-00, 2012.

[36] Q. Zhao, P. Du, J. Mena and M. Gerla, "Software Defined Multi-Path TCP Solution for Mobile Wireless Tactical Networks," in *2018 IEEE Military Communications Conference (MILCOM),* Los Angeles, CA, USA, 2018.

[37] W. Stallings, Cryptography and network security, 4/E, Pearson Education India, 2006.

[38] M. Ramadan, Y. Liao, F. Li, S. Zhou and H. Abdalla, "IBEET-RSA: Identity-based encryption with equality test over RSA for wireless body area networks," *Mobile Networks and Applications* , vol. 25, no. 1, pp. 223-233, 2020.

[39] C. Raiciu, M. Handley and D. Wischik, "Coupled congestion control for multipath transport protocols (RFC 6356)," Internet Engineering Task Force (IETF), 2011.

[40] K. Popat and D. V. Kapadia, "Recent Trends in Security Threats in Multi-Homing Transport Layer Solutions," *International Journal of Advanced Science and Technology,* vol. 29, no. 5, pp. 5641-5648, 2020.

[41] K. J. Popat, J. Raval, S. Johnson and B. Patel., "Experimental Evaluation of Multipath TCP with MPI," in *In Proceedings of the Third International Symposium on Women in Computing and Informatics,* 2015.

[42] C. Pearce and S. Zeadally, "Ancillary Impacts of Multipath TCP on Current and Future Network Security," *IEEE Internet Computing,* vol. 19, no. 5, pp. 58-65, 2015.

[43] F. Mallouli, A. Hellal, N. S. Saeed and F. A. Alzahrani, "A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom),* 2019.

[44] J. Ma, F. Le, A. Russo and J. Lobo, "Detecting distributed signature-based intrusion: The case of multi-path routing attacks.," in *IEEE Conference on Computer Communications (INFOCOM),* 2015.

[45] R. Khalili, N. Gast and M. Popovic, "Opportunistic linked-increases congestion control algorithm for MPTCP," 2013.

[46] B. A. Forouzan and D. Mukhopadhyay, Cryptography and network security, New York, NY, USA: Mc Graw Hill Education (India) Private Limited, 2015.

[47] G. Detal, S. Barré, B. Peirens and O. Bonaventure, "Leveraging Multipath TCP to create Hybrid Access Networks," in *SIGCOMM*, Los Angeles, CA, USA, 2017.

[48] Y. Cao, M. Xu and X. Fu, "Delay-based congestion control for multipath TCP," in *2012 20th IEEE international conference on network protocols (ICNP)*, 2012.

[49] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM journal on computing,* vol. 32, no. 3, pp. 586-615, 2003.

[50] O. Bonaventure, M. Handley and C. Raiciu, "An overview of Multipath TCP," *login,* vol. 37, no. 5, pp. 17-23, 2012.

[51] "Elliptic-curve cryptography," [Online]. Available: https://en.wikipedia.org/wiki/Elliptic-curve_cryptography. [Accessed 01 04 2022].

7. PUBLICATIONS

[1] "An novel scheduling scheme of Multipath TCP for MPI using modified RR algorithm, " Proceedings of 1st International Conference on Emerging Technologies in Computer Engineering in Research maGma An International Multidisciplinary Journal, March 2018.

[2] Khushi Popat, Viral Kapadia,"Recent Trends in Security Threats in Multi-Homing Transport Layer Solutions," International Journal of Advanced Science and Technology", *International Journal of Advanced Science and Technology*, vol. 29, no. 5, pp. 5641 - 5648, May 2020. [Publisher: Science and Engineering Research Support Society, Scopus Indexed, SJR: 0.11]

[3] Popat, K., Kapadia, V.V. (2021). Multipath TCP Security Issues, Challenges and Solutions. In: Bhattacharya, M., Kharb, L., Chahal, D. (eds) Information, Communication and Computing Technology. ICICCT 2021. Communications in Computer and Information Science, vol 1417. Springer, Cham. https://doi.org/10.1007/978-3-030-88378-2_2
[Scopus Indexed, Proceeding is published as a book chapter in Springer]

[4] Ali Almuflih; Khushi Popat; Viral V. Kapdia; Mohamed Rafik Qureshi; Naif Almakayeel; Rabia Emhamed Al Mamlook, " Efficient Key Exchange Using Identity-Based Encryption in Multipath TCP Environment," Appl. Sci. 2022, 12, 7575. https://doi.org/10.3390/app12157575 [Scopus, Web of Science- SCIE, Impact factor: 2.838, cite score: 3.7]