1. INTRODUCTION

With an emerging era of Industry 4.0, the internet of things, blockchain technology, cyber security, edge computing, data analytics, robotics, and machine learning have empowered various industrial sectors and started focusing heavily on the performance and quality of services (QoS) of real-time applications. The revolution of industries has changed the day-to-day lifestyle of humans worldwide by offering services anywhere-anytime on various computing devices such as smart television, wearables, smart phones, etc. which increases the demand for high-performance and reliable communication technologies. Moreover, most modern machines/devices have numerous network interfaces (Wireless Fidelity (Wi-Fi), Global System for Mobile Communication (GSM), Ethernet, etc.) and multiple Internet Protocol (IP) addresses, making them "multi-homed" and "multi-addressed". One of the most promising and widely adopted connection-oriented transport layer protocols, Transmission Control Protocol (TCP), was introduced in 1981 [1]. TCP offers reliable communication between two hosts by binding the communication hosts to the network interface over the packet-switching network.

Nonetheless, most cutting-edge applications rely on TCP as a transport layer protocol for reliable communication; however, in the current scenario, the inefficient use of network resources results from the exclusive reliance of the computing machines on the use of a single communication interface. Moreover, switching from one network to another may interrupt the ongoing communication, leading to data loss and requiring data transfer from scratch. In addition to the increasing rate of devices connected over internet, the demand for performance with fault tolerance is also increased.

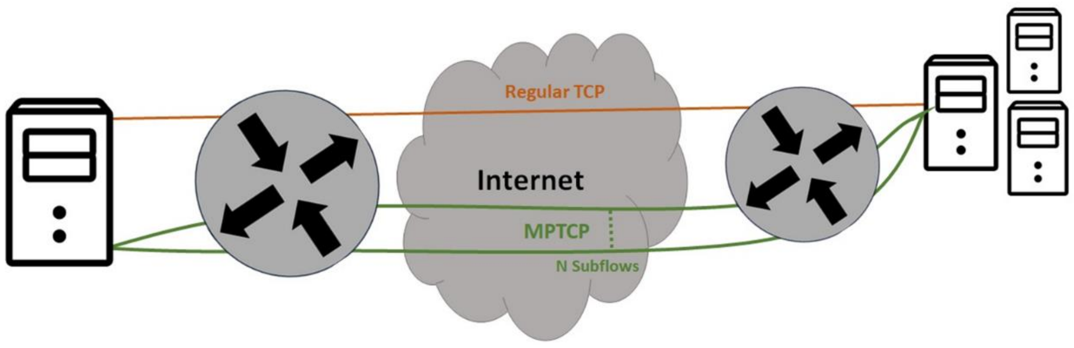


Figure 1.1 TCP vs. MPTCP connection scenario **[2]**

As shown in Figure 1.1, to achieve higher throughput, reliable communication, backward compatibility, and fault tolerance, the researchers from the Internet Engineering Task Force (IETF) have designed a Multipath Transmission Control Protocol (MPTCP) which enables the usage of numerous disjoint routes together to fulfil the requirement of the current scenario [3]. MPTCP permits the practice of multiple network interfaces over a sole connection by establishing multiple sub-connections known as a sub-flow.

As a promising expansion of TCP, MPTCP is a novel transport layer protocol that aggregates available bandwidth to increase throughput and decrease delay for a more stable and reliable connection. As an added bonus, it's resilient thanks to gentle path transitions in the event of a failure. To facilitate and avoid the rejection of packets by middleboxes, MPTCP uses a TCP header. The MPTCP-specific information is being communicated by setting various options in the TCP header, such as MP_CAPABLE, MP_JOIN, ADD_ADDR, etc., making MPTCP backward compatible. The backward-compatible nature of MPTCP has increased the deployment of MPTCP by many industries and academia compared to other extensions of TCP. Many industries and academic institutions are looking towards integrating 5G and MPTCP to improve the overall network performance by simultaneously allowing Wi-Fi and mobile data mobile devices [4]. Mobile and vehicular networks, data centre networks, robotics communication, software-defined networks, etc., are just a couple of the application domains that attract professionals from both the commercial and governmental sectors to MPTCP [5].

However, MPTCP offers numerous benefits over TCP, the use of MPTCP options opens the door to many assaults that are not TCP-specific. SYN flooding attacks, DoS attacks, and session hijacking via Man-in-the-Middle (MitM) assaults can be initiated by exploiting the vulnerabilities of various MPTCP options to steal data from a victim's computer. Nevertheless, because these keys are later utilized for sub-flow authentication in MPTCP, their exchange in unencrypted form during the 3-way handshake opens the door to additional safety risks. This article focuses on the MPTCP security measures taken to prevent attacks exploiting the ADD_ADDR vulnerability and the security of the keys transferred during the initial handshake. Many researchers have studied MPTCP's security challenges, because of the protocol's capability for multi-homing and multi-addressed hosts for data transfer increases security risk while also providing better bandwidth and fault-tolerance.

Many probable solutions have been proposed and implemented by researchers to offer security over attacks like session hijacking, Denial of Service (DoS) attack, SYN flooding, etc., including Transport Layer Security (TLS) [6], tcpcrypt [7], hash chained based solution for MPTCP [8] and sum hash chained-based encryption for MPTCP [9], key exchange using Software Defined Network (SDN) for MPTCP [10], secure and light-weight solution for MPTCP [11]. Some of the solutions mentioned above increase overhead, and others open the doors for attacks which again creates issues for the MPTCP. Suppose an attacker is present during the handshake; they can easily access sensitive information and launch various attacks without going through the necessary authentication procedures required each time a new sub-flow is created, a new address is advertised, a sub-flows priority is adjusted, etc.

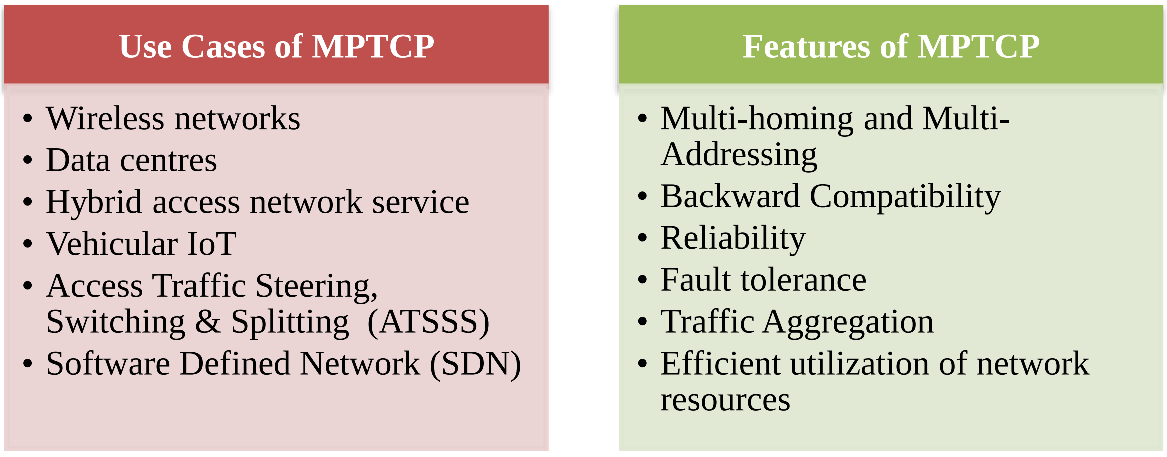
However, few researchers have focused on the security of MPTCP for many years aiming to improve its performance in areas like congestion control, scheduling, and applications (data centre networks, automotive networks, deep learning, etc.). The importance of guaranteeing the security of data transmissions inspired researchers to focus on this topic.

This chapter briefly introduces the MPTCP protocol at the transport layer, its option, security issues, and their impact on security goals. It also tells the motive, problem statement, and research contributions behind this work.

1.1 MULTIPATH TCP (MPTCP)

MPTCP is one of the most promising extensions of TCP compared to other extensions like Transactional Transmission Control Protocol (T/TCP), Wireless Transmission Control Protocol (WTCP), etc. [3]. However, many researchers have proposed extensions to enhance performance, fault tolerance, and load balancing [12]. However, none of them got widespread adoption due to various limitations such as backward compatibility, performance degradation, security concerns, etc. It is required to meet the current demands and to resolve the issues with the existing protocol for the widespread adoption and deployment of the new protocol. Furthermore, it should be constructed so that firewalls and middleboxes may accept it. MPTCP was designed with goals like improved throughput, resiliency, and backward compatibility [13]. MPTCP achieves higher performance by bandwidth aggregation of multiple available paths, resiliency by proper load balancing and data retransmission over less congested paths, and backward compatibility using the TCP header.

Currently, MPTCP is adopted by many companies to satisfy the quality of service, fault tolerance, and higher bandwidth requirement. Additionally, MPTCP Linux kernel implementation can be installed independently on Linux, mac Operating System (OS), Android, and Apple iOS operating systems. The Siri voice assistant app, Maps, and music app on Apple iOS have been using MPTCP since 2013 [14]. Many android smartphone companies, like Samsung, LG, Huawei, etc., are also using MPTCP to enable multiple internet services to offer high-speed internet connectivity simultaneously. Moreover, data centre, software-defined networking, Internet of Things (IoT), satellite communication systems, and mobile unmanned aerial vehicle (UAV) heterogeneous networks are the most important use cases of MPTCP as shown in Figure 1.2.



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 1.2 Use Cases of MPTCP

The organization that has led mobile communication standards, the third-generation partnership project (3GPP), has exhibited the prototype for integrating MPTCP with Fifth-Generation (5G) networks to facilitate Wi-Fi and 5G simultaneously [4]. By combining the use of multiple network interfaces in mobile devices as well as in servers, datacentres, and end machines, MPTCP improves performance in terms of bandwidth and fault tolerance. These characteristics of MPTCP have attracted the attention of major tech companies like Apple, Tessars, 3GPP, Samsung, Huawei, etc. Tessares has installed MPTCP proxies to support hybrid access networks in multiple networks [15] to help people in remote areas access the internet and other modern conveniences. Android smartphones from Samsung, Huawei, and LG use MPTCP to provide simultaneous access to multiple network interfaces, including Wi-Fi and Forth Generation (4G)/long-term evolution (LTE), for faster download speeds. Because of MPTCP's improved throughput and fault tolerance during Wi-Fi/Mobile network handover, Apple has been using it for Apple Maps, Apple Music, and Siri since 2013 [16]. MPTCP also addresses speed and robustness concerns in automotive IoT systems [5]. Load control in data centres via several independent channels is a key application area for MPTCP. Although proposed data centre architectures vary from conventional systems in link organization, they all share that servers are linked via several independent channels [17]. Furthermore, MPTCP can be combined with SDNs to facilitate enhanced communication between government and commercial satellite communication systems and mobile UAV networks [18] [19].

MPTCP is built on the top of TCP at the transport layer of TCP/IP stack, as shown in Figure 1.3, so the header format and protocol stack both remained unchanged. The "option" field of the TCP header is utilized to transmit the necessary details of MPTCP communication [20].

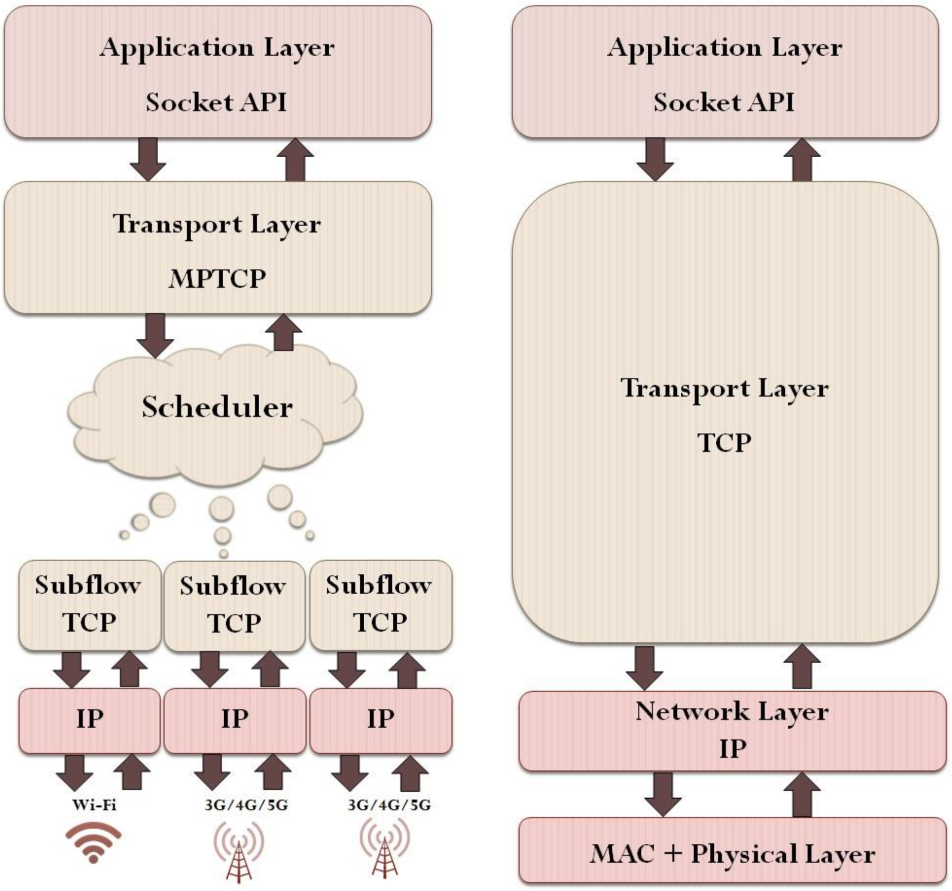


Figure 1.3 TCP vs. MPTCP in TCP/IP protocol stack

By enabling the MP_CAPABLE option in the TCP header, MPTCP allows the conventional TCP 3-way handshake protocol to initiate a connection between hosts [3] [21]. By enabling the MP_CAPABLE option in an SYN packet, the sender indicates that they can receive MPTCP packets. Sending an SYN+ACK message by setting the MP_CAPABLE option in a TCP header to the reply of the SYN packet indicates that the recipient supports MPTCP. If the receipt doesn’t support MPTCP, it sends normal SYK+ACK, which will downgrade the connection to the TCP connection. After receiving the SYC+ACK, the sender will again send the ACK. Sender enables MP_CAPABLE option in the ACK packet only if the receiver is MPTCP supported; otherwise, the connection will be considered as normal TCP.

Here, during the 3-way handshake, the sender and receiver exchange keys in plaintext that will be used to produce the Hash Message Authentication Code (HMAC) during subsequent sub-flow establishment over the connection, thereby authenticating the communicating hosts [22].

MPTCP follows the four-way handshake procedure to establish a new sub-flow over an ongoing connection, as shown in Figure 1.4 [3]. The HMAC calculated with the use of the shared keys from the initial handshake is sent in an SYN packet with the MP_JOIN option set. By enabling the MP_JOIN option in the TCP header of the SYN packet, it can be indicated that the new connection established with handshake must be part of the ongoing connection as a sub-flow and differentiate it from the normal connection establishment. After exchanging the four packets as shown in Figure 1.4, the new sub-flow will be added over the ongoing connection.

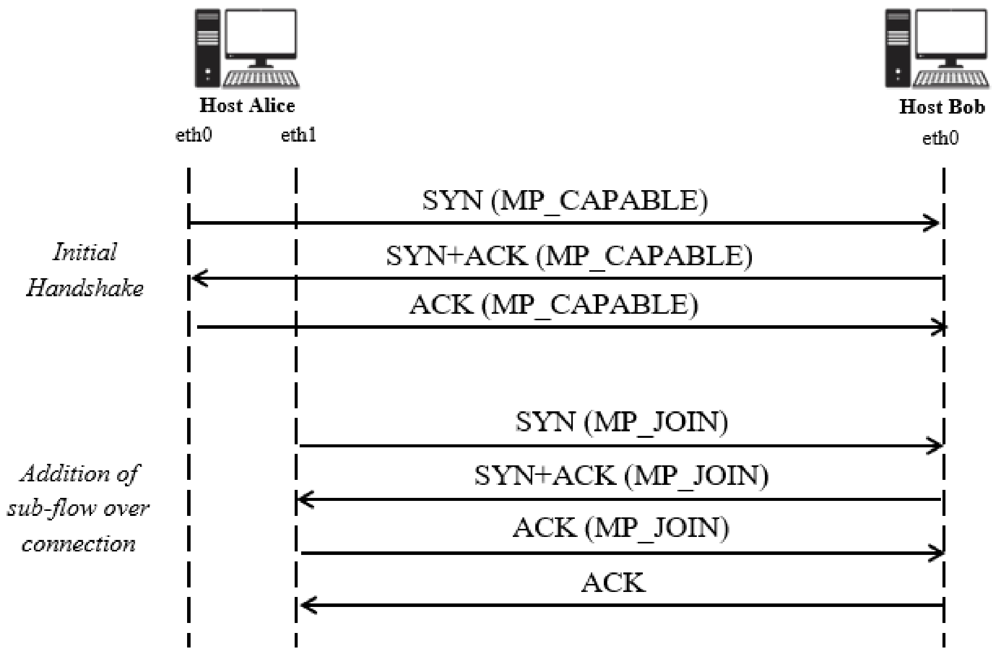


Figure 1.4 MPTCP connection establishment and addition of sub-flow **[22]**

Another option, ADD_ADDR, is also available to advertise the network interface with MPTCP [3]. By configuring the ADD_ADDR field in the TCP header, any communicating network interface can use any of the available IP addresses. In MPTCP version 1, host authentication uses the HMAC computed from the shared keys shared during the connection establishment process [21].

Moreover, REMOVE_ADDR allows the removal of the address during the connection lifetime. MP_PRIO option in the TCP header enables the modification of sub-flow priority to change the direction of data transmission [3]. Data Sequence Signal (DSS) is be added to indicate the data sequence number. Moreover, few other options are available to immediately terminate the connection, to reset the connection, etc.

However, authentication between the communicating hosts is required at the time of establishment of the new sub-flows over the ongoing connection and to advertise the available network interface. A major security flaw in the current implementation of MPTCP is that the authentication keys are transmitted in plaintext during the initial 3-way handshake [22]. If an adversary obtains these keys, they can create a new communication channel between themselves and the server, blocking the legitimate client from accessing the server.

1.2 SECURITY ISSUES WITH MPTCP

IETF has identified significant threats to MPTCP like DoS attack, ADD_ADDR attack, etc. in Request for Comment (RFC) 7430 [23]. Many other researchers have also identified various threats to MPTCP such as side channel attack, traffic diversion attack, data signal manipulation attack, etc. by performing experiments on MPTCP. The eavesdropper during the initial handshake is one of significant concern in MPTCP which exposes the security keys to the attacker that may lead to many securities concerns in future [23]. If an attacker obtains access to the initial keys, they can take control of the connection later. In below subsections, some major security threats to MPTCP are discussed briefly.

1.2.1 ADD_ADDR ATTACK

In a client-server scenario with MPTCP support, the new sub-flow can be established by following 3-way handshake initiated by client with MP_JOIN option. The available network interfaces can be announced to other hosts using the ADD_ADDR option of MPTCP. However, suppose a server has free or new interfaces available. In that case, it can advertise those interfaces to the host so that a client can subsequently establish a new sub-flow. In MPTCP version 0, a host must broadcast a new IP address and address ID (unique number) in an ADD_ADDR packet to announce a sub-flow [3]. To connect to the server as a genuine user over the ongoing connection, an attacker must only counterfeit the ADD_ADDR message because doing so does not require authentication [24].

1.2.2 DENIAL OF SERVICE (DoS) ATTACK

To establish the new sub-flow over the ongoing connection between hosts Alice and Bob, Alice sends SYN+MP_JOIN packet to Bob, which contains a token that Alice has calculated from the authentication keys exchanged during connection initiation handshake [3]. Keys are exchanged in plaintext during the connection establishment, allowing an attacker to create a valid token that can be used with SYN+MP_JOIN to send the sub-flow request to host Bob and begin the joining procedure [23]. The SYN + MP_JOIN message establishes a state in the host that receives it for additional sub-flows. HMAC is computed using a 32-bit random nonce and a 32-bit token sent in the SYN + MP_JOIN that permits the recipient to recognize the MPTCP session. Because this data is not retransmitted during the ACK of the 3-way handshake, it is required to establish a state after receiving an SYN + MP_JOIN.

However, depending on the implementation, there may be a restriction on storing these partially opened connections per MPTCP connection [23]. Therefore, the receiver can become exhausted by sending numerous MP_JOIN messages from various source addresses to initiate a DoS attack using MP_JOIN [22].

1.2.3 OTHER ATTACK

The other possible attacks on MPTCP are SYN flooding attacks, data sequence manipulation, traffic diversion attacks, cross-path inferences attacks, etc. The SYN flooding attack can be initiated by sending SYN+MP_JOIN messages using various ports and IP addresses pair, which exhausts the server's resources [23]. It is also. possible for an on-path attacker to spoof the originating IP address of an SYN/JOIN packet by remaining in the network during the exchange. A traffic diversion attack is possible because an attacker can use the MP_PRIO option to reroute traffic on a compromised sub-flow and gain access to sensitive information [25]. The flooding and DoS attack can be initiated by manipulating the data sequence at the connection level acknowledgment [26].

In RFC 7430, the IETF has proposed many probable solutions against the mentioned attacks [23]. In addition, other researchers have proposed improvements to strengthen the security of MPTCP based on encryption, hashing, opportunistic security, and many more, which will be discussed in the next chapter.

1.3 MOTIVATION FOR THIS WORK

Currently, most computing devices have multiple network interfaces; unlike TCP, MPTCP allows 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 resource utilization, backward compatibility, and performance by using TCP header; it opens the doors to many threats.

- The keys communicated in clear form during the connection establishment will be utilized later for new sub-flow authentication, address advertisement, etc.
- The same keys can be used to exploit the vulnerability of MPTCP and initiate attacks to hijack the session to gain access to information exchanged.
- Session hijacking via address advertisement, DoS attack, and host authentication during sub-flow addition over a connection can be mitigated by using secure technique for the communication of key during connection establishment.
- The security enhancement of MPTCP may resolve the issues related to latency and connection drop and attract many industries, like wireless communication, smart healthcare, data centers, etc., to take the benefits of the features of MPTCP.

1.4 PROBLEM STATEMENT AND OBJECTIVES

Most security flaws of MPTCP stem from the insecure key exchange protocol during the connection establishment procedure. The IETF has proposed several solutions to various security issues. Some solutions were created using the concepts outlined in RFC7430, while others were created by merging several security protocols that will be covered later.

The primary motivation behind this study is exploring a method for authenticating MPTCP sub-flow creation without compromising the protocol's existing performance and security.

1.4.1 PROBLEM STATEMENT

- To design and implement the solution that offers security against session hijacking using ADD_ADDR packet sequence classified as an off-path active attack and eavesdropper present during connection establishment process by protecting the keys communicated during the connection establishment procedure.

1.4.2 OBJECTIVES

- Identify and test the vulnerabilities of MPTCP during communication and show its impact on communication.
- Identify and test the off-path active attacks on the communication network.
- To design or develop some model for securing the MPTCP communication from off-path active attackers so that it doesn’t degrade the performance of MPTCP than TCP.
- Test the designed model against security and performance.

1.5 RESEARCH CONTRIBUTIONS

- MPTCP security risks are assessed and demonstrated, including how the ADD_ADDR packet is susceptible to manipulation, which allows an attack to be launched to hijack a session, and how the keys exchanged during the connection establishment procedure can be intercepted and manipulated to hijack a session [27].
- The performance and security levels of the various probable solutions to several MPTCP security threats are evaluated and examined [27].
- The available solutions to the different MPTCP security threats are assessed by comparing performance and security levels [27] [28].
- Identity Based Encryption (IBE)'s efficiency and safety are compared to the elliptic curve cryptosystem (ECC) concerning required and security complexity [28].
- Secure Key Exchange Model for Multipath TCP (SKEXMTCP) using IBE is the model designed and assessed concerning its security and performance. This model allows the key to be exchanged securely during the initial handshake without the need to exchange keys before the connection is established [28].

1.6 THE OVERALL STRUCTURE OF THE THESIS

The overall thesis has been organized into seven chapters.

Chapter 1, as discussed above, was about MPTCP and its options, the various security threats to the MPTCP, and various encryption techniques which can be used for security. Moreover, the chapter discusses motivation, problem statements, objectives, and research contributions. The remaining of the chapters are ordered as under:

Chapter 2 gives an introduction to the necessary concepts of the MPTCP. It provides essential details of MPTCP and its options, Cryptography and network security basics, Elliptic Curve Cryptography (ECC) and the fundament details of IBE.

Chapter 3 gives a detailed review of the security threats to the MPTCP protocol in the form of a literature study. It also discusses the literature review of various approaches proposed by various researchers for enhancing the security of MPTCP.

Chapter 4 discusses the detailed architecture and implementation of the proposed key exchange model for MPTCP during the initial handshake and its working with MPTCP.

Chapter 5 details the proposed key exchange model's experimental evaluation, performance, and security comparison.

Chapter 6 concludes the whole work and states the possibility of future work.

Chapter 7 lists the publications that emerged through this research work.

Chapter 8 lists the references for the entire content.