6. CONCLUSION AND ROAD-MAP

6.1 CONCLUSION

Transmission Control Protocol (TCP) often employs transport layer protocol for connecting two hosts on a network. Specifically, it establishes communication among hosts with a single network interface. However, with technological advancements, most modern gadgets now feature many network interfaces, including Ethernet and WiFi. The original Transmission Control Protocol (TCP) has been extended with Multipath Transmission Control Protocol (MPTCP) so that several network interfaces can establish a single connection. The TCP headers have not been changed in any way by Multipath TCP. Instead, the extra Multipath TCP data has been sent in the TCP header's options field. Options abound with multipath TCP like MP_CAPABLE, which determines whether or not the hosts are MPTCP-capable, and MP_JOIN, which adds a new sub-flow to an existing connection. For MPTCP, the network connections between two hosts are called sub-flows. When an MPTCP session begins, the first sub-flow is initiated using a shared key pair. The generated sub-flows are then verified using these keys. It's crucial to remember that the keys are presented only as strings of plain text.

Multipath TCP has increased throughput and redundancy and opened up new security flaws. Analysis of security risks has revealed that exchanging authentication keys in plain text is a major vulnerability. With these keys, attackers can design a variety of assaults to hijack the session, unavailable the service, identify the throughput of other networks, etc.

ADD_ADDR vulnerability and eavesdropper in the initial handshake have been identified and implemented to demonstrate the attack scenario with MPTCP. Various security solutions have been analyzed concerning security and performance. However, research in the area is still required to minimize the risk of attacks due to key exchange in clear form during the initial handshake.

In light of this, the primary focus of this study is to investigate potential replacement strategies for enhancing the security of keys exchanged during the initial handshake. Using the Linux kernel's implementation of MPTCP, we proposed a secure key exchange model for MPTCP (SKEXMTCP) based on identity-based encryption (IBE). The keys transferred during the 3-way handshake in SKEXMTCP are encrypted using IBE [25]. The IBE asymmetric encryption technique reduces the burden of exchanging keys before sending data using a random character

sequence as the public key. In this case, the same idea can be utilized at the initial handshake to exchange the session keys. Session keys can be encrypted using the public parameters and exchanged during the initial handshake; the other party's keys can then be decrypted using the private key obtained from the IBE-PKG (Private Key Generator).

The proposed SKEXMTCP improves the security of MPTCP against the ADD_ADDR vulnerability and eavesdroppers during the initial handshake by encrypting the keys during conncetion establishment. The overhead and security complexity of the proposed model is evaluated for MPTCP. The research indicates that the proposed method improves the security of MPTCP security without adding further complexity to the protocol.

6.2 ROAD MAP FOR FUTURE WORK

In TCP, the header is 64 bits; this can be extended in the future to improve the security of MPTCP. The fundamental constraint of MPTCP is the available option size for MPTCP. Boosting the difficulty of the BDH and producing a new ID for the encryption add security to the scheme. Different classes of assaults may be demonstrated to be secure in the future by extending the current approach. In addition, a proposal for private key generation based on unique ID generation via IP address and port can be made. The lightweight IBE solution is a viable option for devices with limited storage and processing power. Moreover, using an immutable ledger for the private key generation can enhance the security level with computational resources. One potentially fruitful area of study in MPTCP security is using machine learning and deep learning techniques to automate the detection and avert attacks on data-flow-managed protocols.