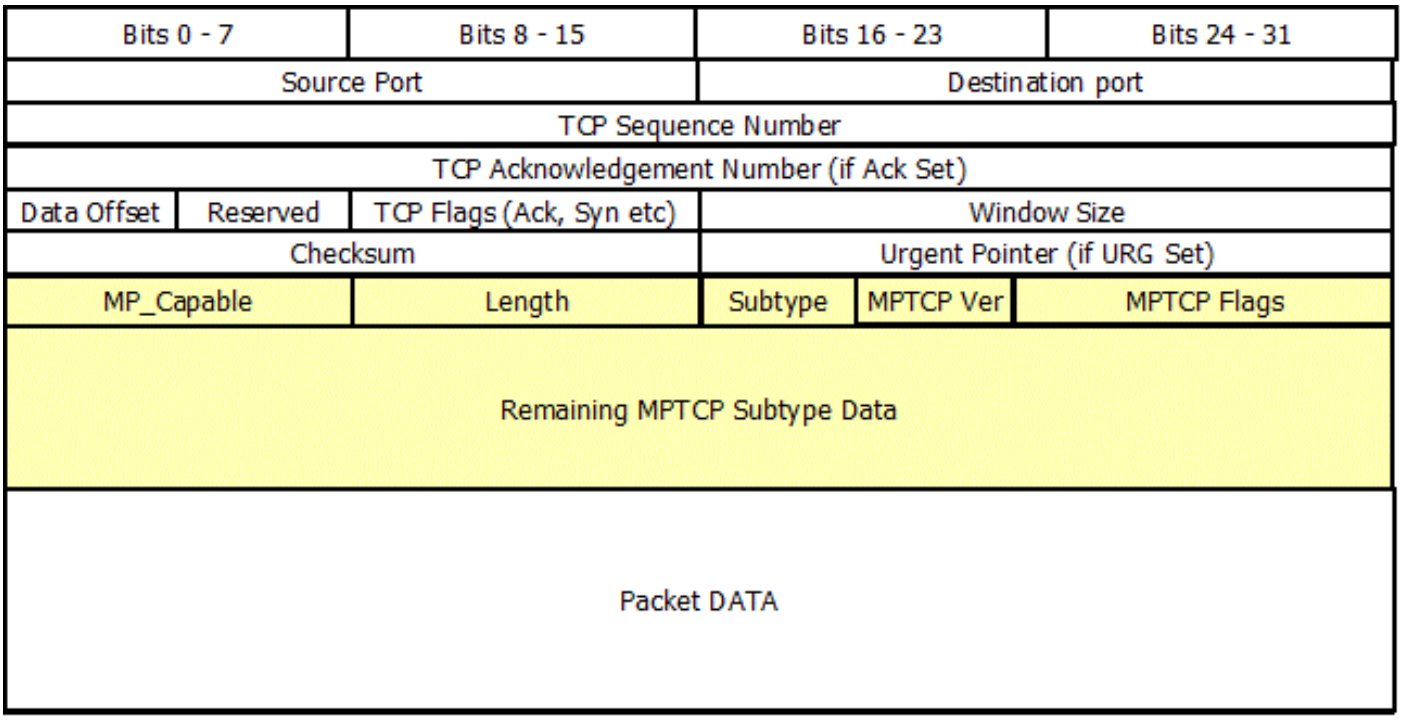
MPTCP Cheatsheet

MPTCP Header:



MPTCP Subtype	HEX	Flags?	Other Likely fields of interest
MP_CAPABLE	0x0		
MP_JOINs	0x1		
DSS	0x2		
ADD_ADDR	0x3		
REMOVE_ADDR	0x4		
MP_PRIO	0x5		
MP_FAIL	0x6		
MP_FASTCLOSE	0x7		

Getting the MPTCP Sequence Numbers:

Key	64 Bit number supplied by host	
Initial DSN (ISDN):	SHA1(key)[-64:]	Binary mode hash, network byte order
Initial DSS		
Subflow DSS	mapping likely starts at ISDN[0:32] + TCP ISN + 1	TCP Seq is 32 bits, + 1 for the SYN
MP_JOIN		

MP_JOIN Authentication (RFC 6824 Fig 8)

A		B
TCP_SYN, MP_JOIN (TokenB, NonceA) ->		
	<-	TCP_SYN_ACK, MP_JOIN(HMAC(Key=KB+KA, Msg = NonceA + NonceB), NonceB)
TCP_ACK, MP_JOIN(HMAC(Key=KA+KB,Msg=NonceB+NonceA) ->		
	<- TCP_ACK	
Token = ConnectionID = SHA1(Key)[0:32] of Other Party's key. (Capture from either steps 2 or 3 in the first handshake)		

Detecting MPTCP things

Passive:	Usage	Inbound Connection Attempts	Detect inbound connection attempts - Look for the SYN packets with MPTCP Header	TCP(SYN) TCP Option= 30 ** 00 …
		Successful Handshake	(Pre-viability) Look for Ack Packets with MPTCP Option header	TCP(ACK) TCP Option = 30 ** 00 …
		Valid Handshake	MPTCP Option header Look for Ack Packets with the MPTCP Option Header	TCP(ACK) TCP Option = 30 ** 00 …
		MPTCP Joins	TCP SYN Packets with MPTCP TCP Option and an MP_JOIN subtype	TCP(SYN) TCP Option = 30 ** 01 …
	Attacks	MPTCP Simple Fragmentation	Non look for non sequential last 32 of DSS numbers	
		Cross Path Fragmentation	MPTCP packets with the last 32 bits of the DSS numbers that don't align with the same stream's TCP sequence numbers (cross-stream fragmentation)	
		MPTCP Address Hopping	MPTCP Stream where the initial handshaking connection is no longer valid	
		MPTCP Resilience Attacks	MPTCP Streams where connections that are killed are reestablished (implementations don't seem to do this)	
Active:	Supporting Listeners	Checksum test		
		Join Failure		
	Supporting Software	Socket-level tests	Try it on an MPTCP capable kernel -	Does it work single stream, multiple stream, if you close the initial, if you change network address
	Path Tests	Tracebox Test	1 - Send a TCP connection attempt 2 - Send MPTCP Connection attempt	Responses: 1:NONE & 2:NONE = Path Down 1:ICMP & 2:ICMP = Blocked or Unroutable 1: TCP_RST & 2: TCP_RST = Port Blocked 1: TCP_SYN & 2: TCP_SYN: = TCP Only or Forced Downgrade OR options Stripped) 1:TCP_SYN & 2: TCP_RST MPTCP Connections Killed

Responding to MPTCP

Downgrade	Strip MP_CAPABLWE from TCP_Syn Packets	This only works if the hosts don have other paths which don't do this (as they will be preferred and you will be blacklisted)
	Insert MP_FAIL after handshake OR Attempt Sending Infinite DSS mapping	Questionable if it will work
Intercept	Intercept individual Subflows	Need either block MP_Joins or pass them through untouched. The authentication is designed to protect against MitM and replay attacks
	Intercept MPTCP Handshake	Foolproof, but requires tracking and linking state across entire connection, may leave you vulnerable to resource exhaustion attacks
	Alter TCP Subflow Data	Will cause MPTCP Checksum failure and result in your path being dropped for any other which doesn't touch data. Difficult to fake checksum as it includes the pseudoheader (related to authentication keys) and would require state tracking with risks as above.
	Flip the checksum flag	[NEED TO CONFIRM THIS WORKS]
Kill	Kill all subflows (RST)	This only works if you kill every subflow on every path, otherwise you will remove all visibility
	Send MP_FASTCLOSE	This will kill entire MPTCP connection (equiv to TCP RST), but it requires capturing and using the full 64bit key from the initial handshake.