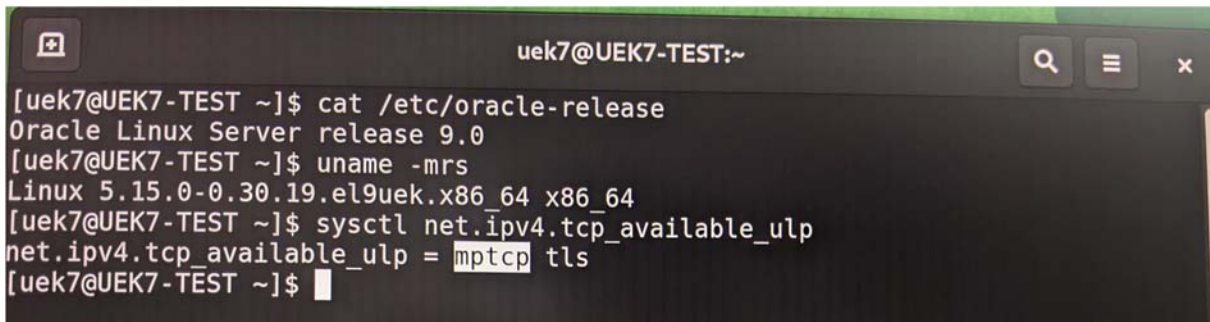
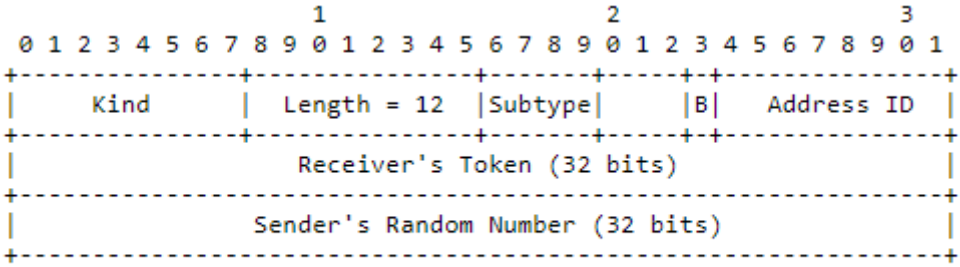


EXHIBIT F

Infringement of U.S. Patent No. 9,350,649 by Oracle MPTCP Products

#	U.S. Patent No. 9,350,649	Oracle MPTCP Products
7a	7. A method for establishing and sending data during a multipath connection, comprising:	The Oracle MPTCP Products perform the claimed method utilizing, for example, devices having Oracle Solaris, Oracle Linux 8.3, Oracle Linux 8.4, Oracle Linux 8.5, Oracle Linux 8.6, Oracle Linux 9, and devices that operate in a similar manner. By way of example, Oracle Linux 9 is charted herein. The Oracle MPTCP Products perform a method for establishing and sending data during a multipath connection as described below:  <pre> [uek7@UEK7-TEST ~]\$ cat /etc/oracle-release Oracle Linux Server release 9.0 [uek7@UEK7-TEST ~]\$ uname -mrs Linux 5.15.0-0.30.19.el9uek.x86_64 x86_64 [uek7@UEK7-TEST ~]\$ sysctl net.ipv4.tcp_available_ulp net.ipv4.tcp_available_ulp = mptcp tls [uek7@UEK7-TEST ~]\$ </pre>
7b	the data-source device receiving multipath connection requests from a plurality of data-requesting devices, each of the multipath connection requests involving a particular multipath connection to a data-requesting device;	The Oracle MPTCP Products include the data-source device receiving multipath connection requests from a plurality of data-requesting devices, each of the multipath connection requests involving a particular multipath connection to a data-requesting device as described below: Network interfaces are identified by a name that combines a prefix and a suffix. The prefix depends on the type of network interface: Ethernet network interfaces: en; Wireless local area network (LAN) interfaces: wl; Wireless wide area network (WAN) interfaces: ww Associating a New Subflow with an Existing MPTCP Connection: The exchange of keys in the MP_CAPABLE handshake provides material that can be used to authenticate the endpoints

#	U.S. Patent No. 9,350,649	Oracle MPTCP Products
		when new subflows will be set up. Additional subflows begin in the same way as initiating a normal TCP connection, but the SYN, SYN/ACK, and ACK packets also carry the MP_JOIN option. Host A initiates a new subflow between one of its addresses and one of Host B's addresses. The token -- generated from the key -- is used to identify which MPTCP connection it is joining, and the HMAC is used for authentication. The Hash-based Message Authentication Code (HMAC) uses the keys exchanged in the MP_CAPABLE handshake, and the random numbers (nonces) exchanged in these MP_JOIN options. MP_JOIN also contains flags and an Address ID that can be used to refer to the source address without the sender needing to know if it has been changed by a NAT. <pre> Host A Host B ----- ----- MP_JOIN MP_JOIN [B's token, A's nonce, [B's HMAC, B's nonce, A's Address ID, flags] B's Address ID, flags] <- ACK + MP_JOIN ACK [A's HMAC] <- </pre>
7c	the data-source device associating a connection ID to each multipath connection request;	The Oracle MPTCP Products include the data-source device associating a connection ID to each multipath connection request as described below: Token: A locally unique identifier given to a multipath connection by a host. May also be referred to as a "Connection ID".

#	U.S. Patent No. 9,350,649	Oracle MPTCP Products
		Host A initiates a new subflow between one of its addresses and one of Host B's addresses. The token -- generated from the key -- is used to identify which MPTCP connection it is joining, and the HMAC is used for authentication. The Hash-based Message Authentication Code (HMAC) uses the keys exchanged in the MP_CAPABLE handshake, and the random numbers (nonces) exchanged in these MP_JOIN options. MP_JOIN also contains flags and an Address ID that can be used to refer to the source address without the sender needing to know if it has been changed by a NAT.
7d	the data-source device using multipath information about each of a plurality of network-edge devices that is capable of joining in a particular multipath connection to a particular data-requesting device, the particular multipath connection having a particular connection ID;	The Oracle MPTCP Products include the data-source device using multipath information about each of a plurality of network-edge devices that is capable of joining in a particular multipath connection to a particular data-requesting device, the particular multipath connection having a particular connection ID as described below: The exchange of keys in the MP_CAPABLE handshake provides material that can be used to authenticate the endpoints when new subflows will be set up. Additional subflows begin in the same way as initiating a normal TCP connection, but the SYN, SYN/ACK, and ACK packets also carry the MP_JOIN option. In the first MP_JOIN on the SYN packet, illustrated in Figure 5, the initiator sends a token, random number, and address ID.  Figure 5: Join Connection (MP_JOIN) Option (for Initial SYN)

#	U.S. Patent No. 9,350,649	Oracle MPTCP Products
7e	the data-source device notifying each of the network-edge devices that is participating in the particular multipath connection associated with the particular connection ID; and	The Oracle MPTCP Products include the data-source device notifying each of the network-edge devices that is participating in the particular multipath connection associated with the particular connection ID as described below: HOST B sends an ACK in response to MP_Join for each additional subflow. <pre> Host A Host B ----- ----- MP_JOIN -> [B's token, A's nonce, A's Address ID, flags] <- ACK + MP_JOIN -> [A's HMAC] <- ACK </pre>
7f	the data-source device sending data associated with the particular connection ID to the network-edge devices participating in the particular multipath connection associated with the particular connection ID.	The Oracle MPTCP Products include the data-source device sending data associated with the particular connection ID to the network-edge devices participating in the particular multipath connection associated with the particular connection ID as described below: At a high level, an MPTCP implementation will take one input data stream from an application, and split it into one or more subflows, with sufficient control information to allow it to be reassembled and delivered reliably and in order to the recipient application.