**Discussion and Countermeasure:**

In the section, it demonstrated the threats, the primary challenges and security problems and solutions for them.

It summarised three primary challenges in the previous section:

* Data leakage.
* Cloud provider’s identity.
* Information integrity.

Data leakage refers to that when the self-driving car is driving on the way, and it is probable that hackers have opportunities to intrude the database and steal data during the transmission and exchange of data from the car to the cloud (Zhao and Ge, 2013). Hence, it is necessary to enhance the security level of communications. The AES (Rijndael) algorithm is the optimal method to encrypt and decrypt the data. The AES (Rijndael) utilised 256-bit keys with flexible and efficient operation speed and could be applied in various platforms, especially small mobile devices (Mitali and Sharma, 2014). Shafagh (2015) also stated that compared to DES and RSA, the AES has larger key size than DES and has faster encryption and decryption speed than RSA.

During driving, the car requires connecting to various cloud providers to obtain the updated traffic map and commands. If the cloud provider is not reliable, or even hackers acted like an authentic provider, the gained map and command information may lead to some terrible accidents. Therefore, the self-driving car technique requires verifying the identity of the cloud provider. The key distribution system is the selection to apply to this circumstance, which is able to ensure the cloud provider is reliable. The key distribution centre (KDC) shares a pair of unique master keys with users through the physical transmission, and when users proceed communications, they have to apply for the exclusive session key from the KDC, which is encrypted by the master key of KDC (Levi and Sarimurat, 2017).

Information integrity, as one of the most significant security requirements of IoT, is able to guarantee that all the collected data are reliable and integrated (Xu, Wendt and Potkonjak, 2014). Some sensors of IoT have the data gathering rate with low real-time delay and high bandwidth features, thus, increasing some data integrity techniques to these sensors not only ensures the velocity but also reduces the external interference (Xu, Wendt and Potkonjak, 2014). In order to figure out the integrity issues, we added two protocols to the communication which are suitable for tradition network, but also appropriate for the IoT infrastructure, TLS (Transport Layer Security) and DTLS (Datagram Transport Layer Security).

As the figure shown below, the TLS contains the server, the client and the channel.

* Server: The server is to monitor the connections from the client and generate a TLSChannel instance to operate these connections (Tiburski et al., 2017).
* Client: When caught a connection, TLSClient sends the message to the Middleware Core (Tiburski et al., 2017).
* Channel: Created by the server and generate a handshake with TLSClient (Tiburski et al., 2017).

By contrast, the DTLS consists of the server, the client and the connector.

* Server: The server is to open the connection channel with the client and transfer the data (Tiburski et al., 2017).
* Client: Start the handshake with the server, receive and send message to the server (Tiburski et al., 2017).
* Connector: Guarantee the encryption, decryption, key and message exchange between the server and client. (Tiburski et al., 2017).

In both TLS and DTLS protocols, after the handshake, all the message exchanges and communications are under the protection.

However, these countermeasures still have limitations. For instance, the encryption and anonymisation technique of AES (Rijndael) will encounter the brute-force attack easily due to the mathematical property (Pawar and Ghumbre, 2016). Meanwhile, Ebrahim, Khan and Khalid (2014) stated that although the AES(Rijndael) does not show any limitations, the implementation of inverse cipher through it has not illustrated adequate performance on the smart card.

For key distribution, the security level of the KDC depends on the protocol, and some simple protocols may lead to the man-in-the-middle attack (Salman et al., 2016). Secondly, the storage limitation is also the potential problem for some self-driving cars (Salman et al., 2016). Supposed there are M pairs of self-driving vehicles and clouds which are going to communicate, so we have M users and M\*(M-1)/2 communications. Then, the number of session keys and master keys which need to store in the KDC increase to M\*(M-1)/2 and M respectively as one communication uses one session key. Apparently, the total of keys stored in the KDC is M+M\*(M-1)/2, which requires a considerable memory space. In addition, for each self-driving car, it has to store one master key and M-1 session keys. If the M is too large, the limitation of storage spaces for both KDC and self-driving car require resolving.

Moreover, the overhead of TLS and DTLS should be taken into consideration. When the TLS and DTLS first implemented, it requires buying additional equipment since it is designed for web-based applications (Tiburski et al., 2017). Through the experiment, Tiburski et al. (2017) also demonstrated the overhead increases after applying TLS and DTLS is not only relevant to the deployment of the security layer infrastructure but also the transmission and sequence of messages.

**Conclusion:**

To sum up, the report introduced the architecture and relationship of IoT-based self-driving car and vehicular cloud, and analysed challenges and their preliminary solutions for this industry. The result of the analysis illustrated that although there exists shortage and challenges, with the application of the cryptographic techniques, the vehicular cloud, the self-driving car and related techniques are increasingly mature.

The connection between the self-driving car and vehicular cloud may encounter large amounts of security challenges. For example, confidentiality, authentication, information integrity, verification and access control. Therefore, to figure out these issues, we compared the common cryptographic algorithms and techniques in the market and selected appropriate ones for different issues. Based on the analysis results, we applied AES for solving confidentiality problems, KDC for figuring out authentication issues and TLS/DTLS for settling integrity challenges.

However, these solutions are not perfect and have limitations. It is probable that these limitations become the block of the development of the self-driving car and vehicular cloud industry. The disadvantages of them should be considered to figure out in the future.