CUBE

Autonomous Car Network Security Platform based on Blockchain

November 16, 2017

Executive Summary

CUBE is a platform for securely protecting network security of existing automobiles and autonomous vehicles with blockchain technology. Nowadays, automobiles are evolving rapidly as connected vehicles rather than traditional mechanical parts. As navigation routing, traffic information, and etc are becoming basic features of vehicles, connected units inside vehicles such as ECU, Brake Control Unit(BCU), Wheel Control Unit(WCU) and so on are also becoming vulnerable to malicious attack.

More than 30% of the functions of autonomous vehicles depend on communication. The traffic control centers monitor whether the wheel and brake control units are operating without failure. The navigation required for vehicle operation also depends on communication.

With the above in mind, one of the most important concerns related to autonomous vehicles is securing them against malicious network attacks. To date, no fundamental defense mechanism has been developed against malicious attacks on networks. This risk is the biggest problem for autonomous vehicles.

CUBE solves the security problems of autonomous vehicles using blockchain technology.

The key to Blockchain is that technology ensures trust. CUBE uses block-chain technology to ensure the security of autonomous mobile networks.

In the operation of autonomous vehicles, many IOTs provide information to autonomous vehicles. The attacker seeks to gain access to the network between an autonomous car and IoT or traffic center. In such instances, the hash of the infected binary differs from the hash included in the multisig transaction which is signed by the SW provider and the OEM.

Thus, the vehicles can readily detect such an attack before installing the infected SW update. Distribution of a false update by claiming to be the OEM or SW update provider could be ceased since the overlay nodes are aware of the Public Key of the OEM and the SW update provider. Therefore, the attacker cannot claim to be either of these entities as it requires the private key associated with the Public Key of the relevant entities.

CUBE also uses AI and machine learning to create a defense against possible malicious attacks in the future. In the end, malicious attacks are typically a combination of attacks that have already been introduced, so CUBE provides its own defense against malicious attack algorithms by using tens of millions of combinations of possible attack scenarios, which deep learning generate.

Finally, CUBE uses quantum hashing cryptography technology to improve security. Currently, blockchain technology uses hash as the core of its security; however, if a computer’s processing speed increases dramatically, there is no guarantee of security within an hour. CUBE's quantum hashing cryptography is based on the properties of quantum and already provided technical basis. This approach will serve as an upgrade to the security of the entire blockchain.

To secure the protection of Token contributor and to make transparent company operation, following will be conducted.

Introduction

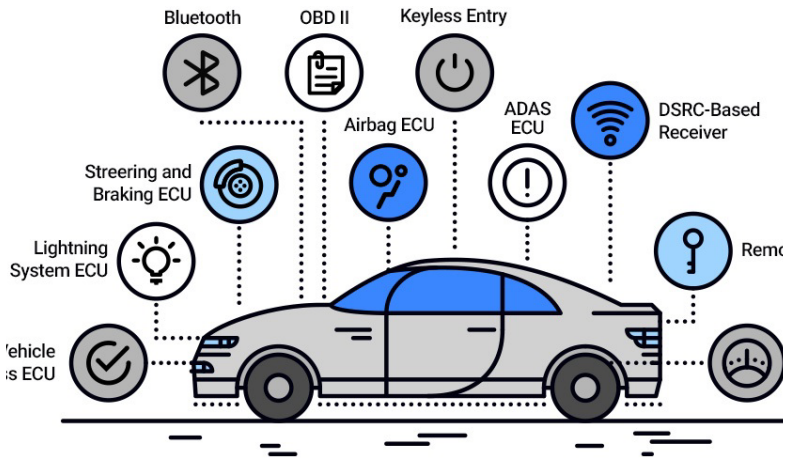
Connected car and Autonomous cars are controlled not only by camera recognition, Ladar, but also network. An autonomous car should always be connected to a network for receiving accurate location data, a vehicle to vehicle data (V2V), traffic data, IOT assist data, and so on. If an autonomous car is connected to LAN, it could be much simpler to detect hacking danger. But, lots of network connections with wireless network increase the hacking danger dramatically. This high degree of connectivity makes it particularly challenging to secure smart vehicles. Malicious entities can compromise a vehicle, which endangers not only the security of the vehicle but also the safety of passengers.

CUBE is the security platform to prevent the hacking danger on the basis of the blockchain. We all remember the devastating WannaCry ransomware attack that struck organizations around the world in May 2017. The attack spread at a rate of almost 3,600 computers per hour or about one per second. When all was said and done, the ransomware infected more than 300,000 devices. Although infected server and PC suffered lots of pain, an autonomous car can be much more dangerous. A hacked autonomous car could threat the human life. CUBE platform decentralizes the IOT control server, and filter the hacked network data by blockchain private key identification.

Potential Dangers

Autonomous cars are revealed to the potential malicious intended attackers in various channels. There are two kinds of weak channels to be attacked, inter-communication channel, and intra-communication channel. Inter-communication channel includes the communication between outside information delivery sites and autonomous car gateway. It includes traffic information, navigation routing information, and remote firmware upgrade by automakers. Another channel is the intra-communication channel. An autonomous car has the intra network, which includes Electric Central Unit (ECU), Break Central Unit (BCU), Wheel Control Unit (WCU), tire, and so on.

Intra-Communication Channel needs high safety and security. Ironically, to secure the safety of this Intra-Communication-Unit, this network should always be connected with automaker control center to check the status of the autonomous car, and it increases the possibility of a cyber-attack by malicious intended attackers.



Most hackable points of vehicle.

Inter-Communication-Network Security

An autonomous vehicle must acquire as much information as possible about its surroundings to operate the vehicle alone. Such information may be path information for navigation, traffic information, or data for updating an old firmware of an autonomous vehicle. Receiving such a variety of data can be very helpful in operating autonomous vehicles, but at the same time increases the risk of malicious intrusions.

V2V, which means Vehicle to Vehicle communication, is an important function to make an autonomous car safer. Short-range communication systems are vehicle-based data transmission systems configured to transmit vehicle operational data to other nearby vehicles and to receive vehicle operational data from other nearby vehicles. The vehicle-to-vehicle (V2V) transmissions between the

short-range communication systems may be sent via DSRC, Bluetooth, satellite, GSM infrared, IEEE 802.11, WiMAX, RFID, and/or any suitable wireless communication media, standards, and protocols. In certain systems, short-range communication systems may include specialized hardware installed in vehicles (e.g., transceivers, antennas, etc.), while in other examples the communication systems may be implemented using existing vehicle hardware components (e.g., radio and satellite equipment, navigation computers) or may be implemented by software running on the mobile devices of drivers and passengers within the vehicles.

The range of V2V communications between vehicle communication systems may depend on the wireless communication standards and protocols used, the transmission/reception hardware (e.g., transceivers, power sources, antennas), and other factors. Short-range V2V communications may range from just a few feet to many miles, and different types of driving behaviours may be determined depending on the range of the V2V communications. For example, V2V communications ranging only a few feet may be sufficient for a driving analysis computing device in one vehicle to determine that another vehicle is tailgating or cut-off the vehicle, whereas longer communications may allow the device to determine additional types of driving behaviours (e.g., yielding, defensive avoidance, proper response to a safety hazard, etc.)

IOT Security

Many IOT devices are used to operate autonomous vehicles. The most representative is the “Guide Assist IOT,” which will be applied to smart roads. This IOT informs the autonomous car of its current position, receives speed and driving information from the autonomous vehicle, and sends this information to the clients who need it. Of course, information related to privacy will be sent only to some authenticated clients.

Internet of Things (IoT) has an identity problem, namely: a lack of authentication and encryption solutions that can scale to meet the unique demands of IoT

deployments. Strong identity in the form of validly issued Public Key Infrastructure (PKI) certificates is the bedrock of online identity and security. They are used today for everything from securing user access to devices and stored data, to device communications, secure boot and software updates (patching). The Internet of Things demands far more of them to secure far, far more devices and data: from tire pressure sensors on connected cars to ambient temperature sensors in a "smart" building and biometric data collected from an implanted medical device.

In the absence of strong authentication and encryption, and methods to provide data and system integrity, any one of these "connected" features can become a vulnerability and point for compromise, subverting the work and purpose of the device or providing a toehold or pivot point for unauthorized access to a sensitive network. Without strong encryption to protect communications to and from smart endpoints, connected devices are vulnerable both to "brute force" password guessing and snooping on sensitive communications via "man in the middle" attacks.

There is plenty of evidence that this is a clear and present danger. A recent study published by HP estimated that almost three-quarters of connected devices fail to encrypt communications to and from the Internet or local networks. A lack of cryptographically signed firmware leaves devices vulnerable to software-based takeovers.

Intra-Communication-Network Security

Automakers need to communicate with autonomous cars continually. Most important is navigation routing information, which includes traffic information. For complete autonomous car self-driving, the car should receive the routing information from the automaker's traffic management control center. Even though the car has the navigation map data, it should receive the best route information, which comes with traffic information. The potential danger is a malicious attacker with fake traffic information, such as blocking the road or misleading traffic information.

Different communication protocols are developed to support the communication. Among the protocols, Controller Area Network (CAN) as the de factor standard of in-vehicle network communication is such a simple communication protocol supporting to connect sensors and actuators with ECUs, and the adoption of CAN facilitates emerging automotive applications. Quite often important information such as diagnostic, informative, and controlling data is delivered through a CAN bus to serve the automotive services such as self-driving and advanced driver assistance systems (ADAS). The information must be secured for the safety of a driver. However, the growth of networking capability is accompanied by significant security concerns, and unfortunately, the in-vehicular network includes several security flaws. ECUs can obtain any ECU-to-ECU broadcasting messages on the same bus, and they are unable to identify a sender. It is shown in how faked packets can confuse critical components securing driver's safety by malicious attacks such as a packet injection and data manipulation

An autonomous car has much more complicated Electronic Control Unit, which includes Break Control Unit (BCU), Transmission Control Unit (TCU), Wheel Control Unit (WCU), and many other units to control self-driving functions. Being penetrated ECU by malicious attack makes serious danger. The problem is the network between automakers and the gateway of this autonomous car. An automotive company should check each autonomous car's IOT devices to make sure that autonomous car's all ECU works without problem. At the same time, an automotive company should upgrade automotive car's firmware remotely through a network. All of these network connections make vulnerable to the malicious attack.

CUBE Security Platform

CUBE consists of three layers. The first is the BC Layer, the second is the AI Layer, and the third is the Quantum Layer. The first layer, Blockchain Layer, is a layer using the technique of Blockchain, and the AI Layer is a layer using artificial intelligence. The third quantum layer is a layer using Quantum Chritography. The block chain layer will be commercialized in 2018. AI Deep Learning Layer will be commercialized in 2020, and Quantum Hash Cryptography will be commercialized in 2022.

Blockchain Layer

The key to Blockchain is that technology ensures trust. CUBE uses block-chain technology to ensure the security of autonomous mobile networks. But there are various difficulties in applying traditional BC to autonomous vehicle safety. Blockchain instantiations suffers from high overhead and low scalability. The consensus algorithm employed in Blockchain involves solving a hard-to-solve easy-to-verify puzzle that consumes significant computational resources. All transactions and blocks are broadcast to the entire network which results in pronounced packet overhead. Additionally, this raises a scalability issue as the number of broadcast packets increases quadratically with the number of participating nodes.

CUBE solves these limitations of traditional BC technology with hybrid BC, which use both public blockchain and private blockchain. If it is slow, but requires a higher level of security, use Public Blockchain. However, some level of security is required, but if you need fast speed, use private blockchain.

In the operation of autonomous vehicles, many IOTs provide information to autonomous vehicles. The attacker seeks to gain access to the network between an autonomous car and IoT or traffic center, and manipulate the software binary with the goal of injecting malware into a large number of vehicles. In such

instances, the hash of the infected binary differs from the hash included in the multisig transaction which is signed by the SW provider and the OEM.

Thus, the vehicles can readily detect such an attack before installing the infected SW update. Distribution of a false update by claiming to be the OEM or SW update provider could be ceased since the overlay nodes are aware of the PK of the OEM and the SW update provider. Therefore, the attacker cannot claim to be either of these entities as it requires the private key associated with the PK of the relevant entities.

The problem of Conventional Blockchain

Conventional security and privacy methods used in smart vehicles tend to be ineffective due to the following challenges:

- ***Centralization***: Current smart vehicle architectures rely on centralized brokered communication models where all vehicles are identified, authenticated, authorized, and connected through central cloud servers. This model is unlikely to scale as large number of vehicles are connected. Additionally, the cloud servers will remain a bottleneck and a single point of failure that can disrupt the entire network.

- ***Lack of privacy***: Most of the current secure communication architectures either do not consider user privacy, e.g. they resort to exchanging all data of the vehicle without the owner's permission, or reveal noisy or summarized data to the requester. However, in several smart vehicle applications, the requester needs precise vehicle data to provide personalized services.

- ***Safety threats***: A malfunction due to a security breach (e.g., by installing malicious SW) could lead to serious accidents thereby endangering the safety of the passengers and also of other road users in close proximity.

Considering the weakness of conventional security methods, CUBE adopted Blockchain as a key security platform for autonomous car. BlockChain is a distributed database that maintains a growing list of blocks that are chained to each other. BlockChain is managed distributed by a peer to peer network. Each node is identified using a Public Key (PK). All communications between nodes,

known as transactions, are encrypted using PKs and broadcast to the entire network. Every node can verify a transaction, by validating the signature of the transaction generator against their PK. This ensures that BC can achieve trust-less consensus, meaning that an agreement between nodes can be achieved without a central trust broker. All transactions (i.e. communications) in the network are encrypted using asymmetric encryption. Nodes are authenticated using their PKs. Strong communication security and authentication introduced by BC mitigates the risk that the vehicle may be remotely hacked and thus increases the safety of the passengers.

There are various difficulties in applying traditional BC to autonomous vehicle safety. Blockchain instantiations suffers from high (processing and packet) overhead and low scalability and throughput. The consensus algorithm employed in Blockchain involves solving a hard-to-solve easy-to-verify puzzle that consumes significant computational resources. All transactions and blocks are broadcast to the entire network which results in pronounced packet overhead. Additionally, this raises a scalability issue as the number of broadcast packets increases quadratically with the number of participating nodes. The throughput of the Blockchain is defined as the number of transactions that are stored in Blockchain per second. Conventional Blockchains have limited throughput, e.g. Bitcoin throughput is restricted to seven transactions per second due to the complexity of the consensus algorithm.

CUBE as a Hybrid Blockchain

CUBE's autonomous vehicle security is based on decentralized blockchain technology. Each entity is a node, making it basically the same as the security systems running on Ethereum. However, in the case of an autonomous car, there is a limit to the use of existing blockchain methods. Because of the blockchain’s slow speed and scalable problems, the blockchain technique cannot be used “as-is.” For this reason, CUBE has been preparing a hybrid chain.

CUBE is composed of hybrid blockchains that use public and private blockchains together. When datafying a vehicle’s drive, tons of data is generated, yet the data

will vary depending on the method of how to accumulate the data. Even when counting only the driving information and the peripherally recognized information used in datafication, enormous amounts of data, up to 4TB, are generated every day. This, in turn, takes a lot of time to process or share. By contrast, vehicles require fast data processing, transmission, and receipt to prevent accidents.

The security of current autonomous vehicles is inefficient, in terms of velocity and volume, when it comes to handling by the public blockchain. To date, we have seen Hyperledger, the Linux Foundation's Umbrella, and R3, in addition to more than 75 banking consortiums around the world, as examples of private blockchains. CUBE constitutes the private blockchain of autonomous cars, operating as a private blockchain to solve the problems of velocity related to data processing and volume. This, however, requires a very high level of trust in critically important elements, such as firmware upgrades by automakers. Those elements that require such a high level of trust will rely on public blockchains, such as Ethereum.

In summary, CUBE uses its own private blockchain, but also uses a public blockchain for critically important elements; thus, CUBE is a hybrid blockchain that can be used for all aspects of autonomous cars.

The Blockchain Island Theory

The relationship between public and private blockchains is the difference between the internet and an intranet. A public blockchain, such as Ethereum, is a reliable technology that cannot be manipulated. A private blockchain is less secure; it is essential to connect private blockchain to a public blockchain to help to ensure an adequate security level.

Ethereum's blockchain is a sea of trust in which many people are involved. In these seas, private blockchains form islands. Each island increases users' levels of trust by placing bridges to other private blockchains that they need. For instance, the government can form a private blockchain, and it can require a private blockchain consortium, such as R3 and bridges, as the occasion demands.

These private blockchains live in the ecosystem of public blockchains like Ethereum through many bridges while increasing their efficiency by designating fast transaction and the most reliable entities. This is the advantage of a private blockchain.

CUBE Governance Issue

CUBE's governance will come from the participating automakers and owners of autonomous vehicles. Each participant becomes a node and has suffrage in the main decision-making process. The costs of this transaction will be paid to the carmakers as Proof of Share (POS). Automakers will have exclusive rights to data upgrades that require high reliability, such as super nodes firmware for each autonomous vehicle.

DAAP Based on CUBE Platform

CUBE is a hybrid blockchain platform that allows vehicles to easily share reliable data peer to peer (p2p). On this platform, a variety of future Internet of Things (IoTs) (e.g., guidance assist IoT, traffic information assist IoT, collision avoidance IoT, etc.) can be easily shared with trust.

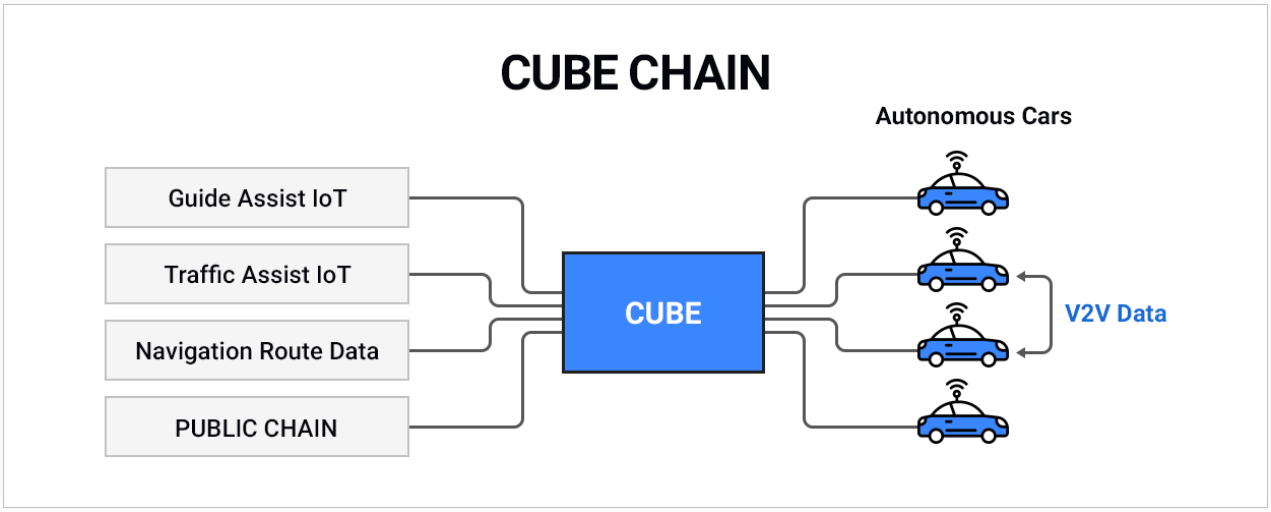
CUBE Blockchain Security Operation

CUBE solves these limitations of traditional BC technology with two concepts of segmentation and permission. In the operation of autonomous vehicles, many IOTs provide information to autonomous vehicles. This information may be transmitted directly between the IOT and the vehicle or may be transmitted through a center that controls the IOT.

The attacker may seek to gain access to the network between an autonomous car and IoT or traffic center, and manipulate the software binary with the goal of injecting malware into a large number of vehicles. In such instances, the hash of the infected binary differs from the hash included in the multisig transaction which is signed by the SW provider and the OEM. Thus, the vehicles can readily detect such an attack before installing the infected SW update. Distribution of a false update by claiming to be the OEM or SW update provider could be ceased since the overlay nodes are aware of the PK of the OEM and the SW update provider.

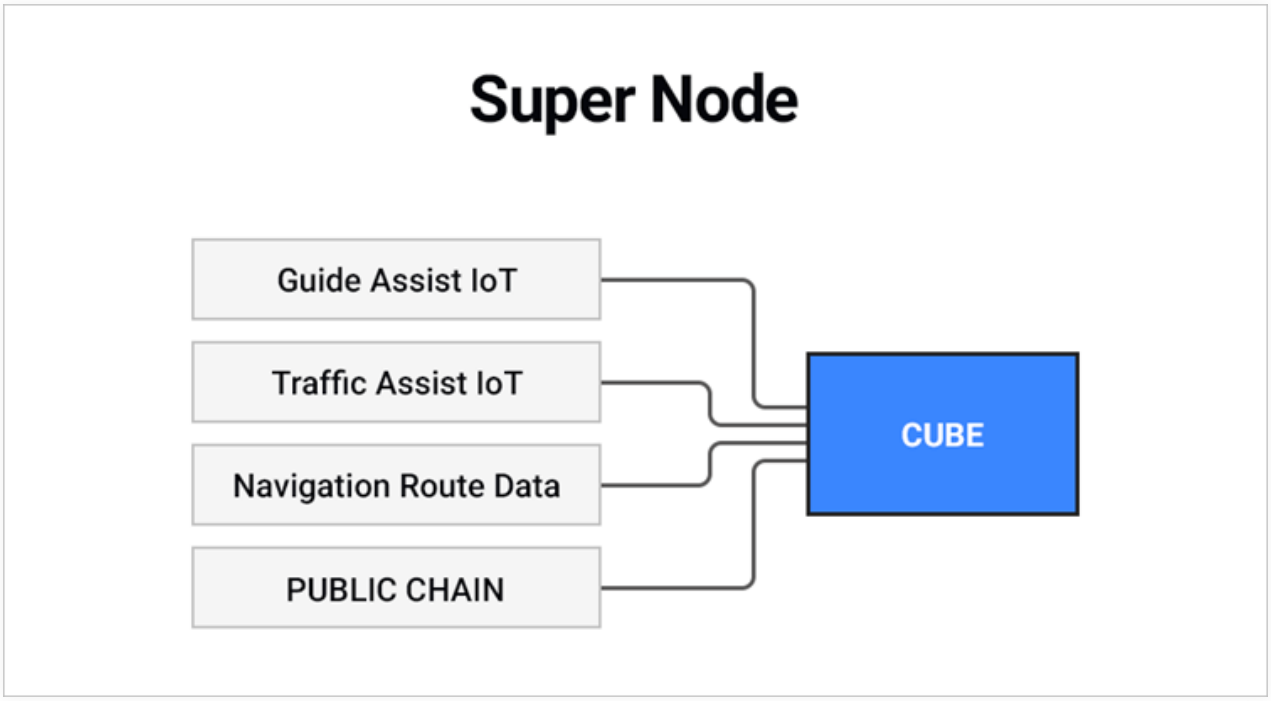
Therefore, the attacker cannot claim to be either of these entities as it requires the private key associated with the PK of the relevant entities.

Regarding distributed Denial of Service (DDoS) attack, to manage a DDoS attack, it is necessary to compromise a large number of vehicles in the overlay. The compromised vehicles send a large number of transactions to a targeted overlay node to overwhelm it. Remind that transactions are broadcast to all OBMs. An OBM forwards a transaction to a cluster member only if the keys in the transaction (i.e. PK.1 and PK.2) match with a key pair in the key list of the OBM. The overlay nodes authorize requesters to access them by uploading a key pair in the key list of the OBM. DDoS attack would not generate a match in the key list and would thus be dropped and not impact the targeted node.



The data sent from the IOT to the car includes information such as location information and road conditions. Conversely, the information sent from the car to the IOT includes speed information and vehicle status information.

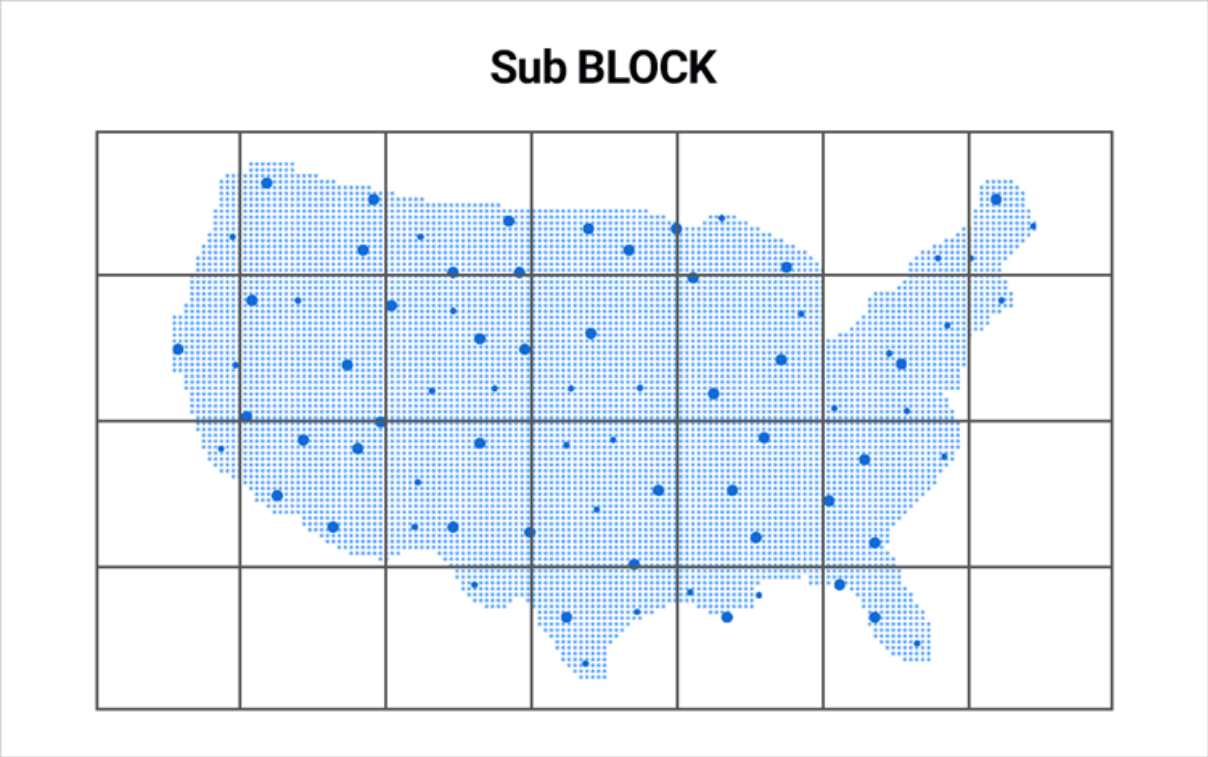
For the safety of autonomous vehicles, CUBE uses Blockchain to handle communication between these vehicles and the IOT. Each of the IOTs becomes a node, and the autonomous vehicle becomes a node. The data transfer between these cars and the IOT is considered transaction.



The essential information used in another autonomous vehicle is self-driving data provided by the automaker, the traffic management center, and so on. The automaker must remotely monitor the status of the vehicle from autonomous vehicles and remotely upgrade the firmware.
The autonomous car should receive only the authenticated data from the authenticated center.
The authenticated center should always be connected with blockchain and should be updated continually. Because most malicious entities disguise themselves with these centers, this node is very important and needs to be trusted.

Blockchain was originally characterized as permission-free, but since communication with autonomous vehicles requires the highest security, only autonomous vehicles should be allowed to communicate with permission. Therefore, unlike the case with normal blockchain, the data should be processed based on permission.

To allow only authorized entities to access a node, CUBE uses the concept of a super node. A super node plays an important role, such as providing information to an autonomous vehicle and upgrading it, unlike a general autonomous vehicle. These super nodes can be specified by the automaker or the government.



As seen above, each sub-block works just like current blockchain. It updates every nodes' transaction, such as traffic information, automakers' hardware upgrade, and navigation route information. Because the size of sub-block is much smaller than blocks, the time to add new block become much faster than the current method.

Hand Shaking

If the autonomous vehicle's driving information needs to be handed over from one base station to another, the data may be interrupted in the meantime. In this case, the two base stations should negotiate so that the data can be transmitted well so that the data is not disconnected. These two protocols are called hand shaking. The handshaking process usually takes place to establish rules for communication when a computer sets about communicating with a foreign device. When a computer communicates with another device like a modem, printer, or network server, it needs to handshake with it to establish a connection.

Handshaking can negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel including information transer rate, coding alphabet, parity, interrupt procedure, and other protocol or hardware features. Handshaking is a technique of communication between two entities.

However, within TCP/IP RFCs, the term "handshake" is most commonly used to reference the TCP three-way handshake.

A simple handshaking protocol might only involve the receiver sending a message meaning "I received your last message, and I am ready for you to send me another one." A more complex handshaking protocol might allow the sender to ask the receiver if it is ready to receive or for the receiver to reply with a negative acknowledgment meaning "I did not receive your last message correctly, please resend it" (e.g., if the data was corrupted en route).

Single-Signature and Multi-Signature

Nodes use transactions to communicate with other nodes in the overlay. There are two types of validation. The number of signatures that must be validated:

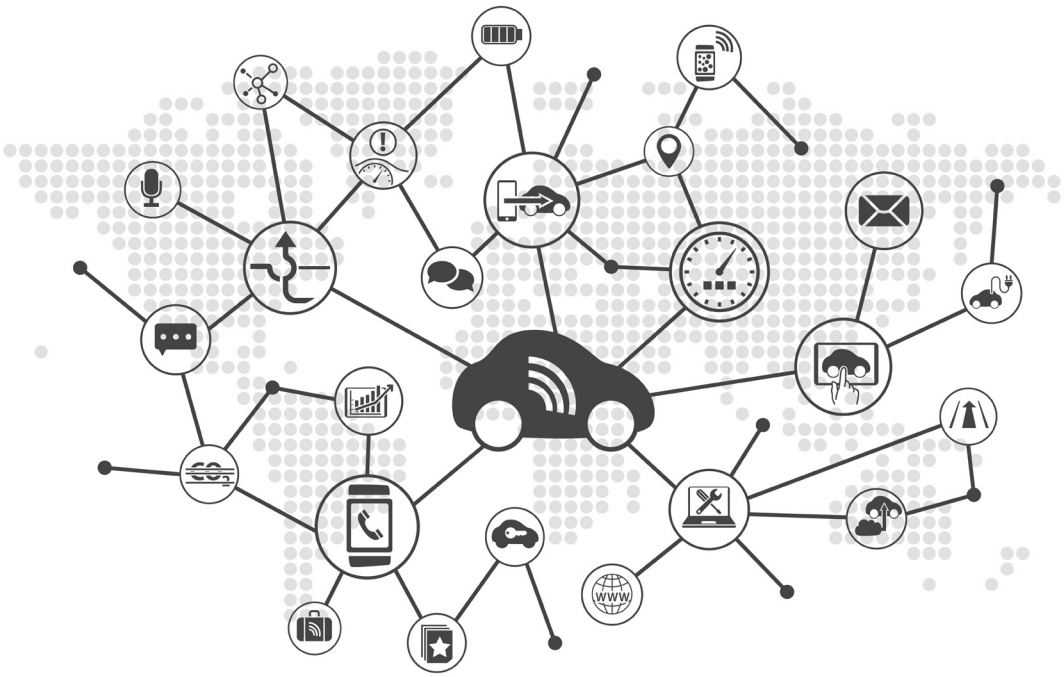
- ***Single signature:*** A single signature is used for simple automotive networks, such as traffic information, and simple road condition information transmission.
  A single signature transaction requires one signature, which is the signature of the transaction generator, to be considered valid. It is used to link subsequent transactions of the same node, thereby creating a transaction ledger for that node. This is followed by the PK and signature (Sig) of the transaction generator.

- ***Multisig:*** Multisig is used for the data transmission, which requires higher security, such as automotive firmware upgrade or patch. A multisig transaction requires two signatures, which are the signature of the transaction generator and recipient, to be considered valid. The subsequent fields contain the PK and signature (Sig) of the transaction generator and recipient.

All transactions are broadcast to all OBMs. An OBM checks the validity of the received transaction by verifying the affixed signature(s). If the transaction is valid, then it is stored in a pool of valid transactions which will be collated to form a block with a pre-defined block size, i.e., the total number of transactions stored in the block. A multisig transaction that arrives at the OBM may yet need to be

signed by the recipient, particularly when the recipient belongs to the cluster of that OBM. Each OBM maintains a list of PK pairs (essentially an access control list) which establishes the nodes that are allowed to communicate with each other. The cluster members (i.e., overlay nodes) upload key pairs to the key list of their OBM to allow other overlay nodes to access them. If the OBM finds a PK pair in its list that matches with the PKs in the transaction (PK.1/PK.2), then it forwards the transaction to the corresponding node that uploaded the key pair. Otherwise, the transaction is broadcast to other OBMs.

AI Deep Learning Layer

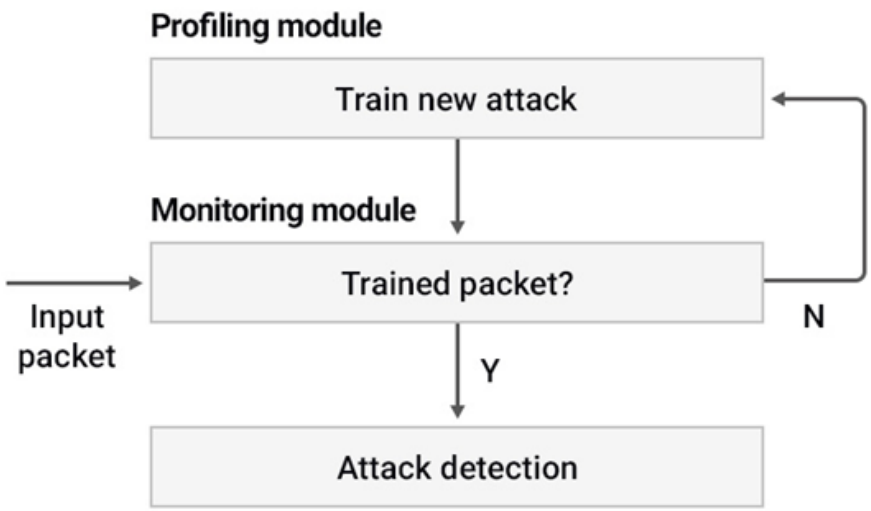


To enhance the level of security, AI (Artificial Intelligence) will be applied at this stage. Recently, attacks on hackers' networks with malicious intent are rapidly evolving. Until now, however, cyber-security technology has been a passive method of collecting vaccines to defend against the attacks that have been presented so far. Cube has developed deep learning network security rather than this passive approach to choose a method based on predictions of malicious attacks rather than the defensive methods so far. Until now, CUBE will continue to learn how malicious attackers have attacked the network and to prevent future attacks.

Intrusion detection system considers a general type of an attack scenario where malicious data packets are injected into an in-vehicle CAN bus. In-vehicular networks are accessed from the mobile communication links such as 4G, IOT network, and WIFI or a self-diagnostic tool such as OBD paired with the driver’s mobile device.

CUBE's self-taught Learning

CUBE's self-taught Learning is a deep learning approach that consists of two stages for the classification. First, a good feature representation is learnt from a large collection of unlabeled data, xu, termed as Unsupervised Feature Learning. In the second stage, this learnt representation is applied to labeled data, xl, and used for the classification task. Although the unlabeled and labeled data may come from different distributions, there must be relevance among them. The below image shows the architecture diagram of self-taught learning. There are different approaches used for Unsupervised Feature Learning, such as Sparse AutoEncoder, Restricted Boltzmann Machine (RBM), K-Means Clustering, and Gaussian Mixtures.



Architecture of Deep Learning Flow

Deep Learning for Classification

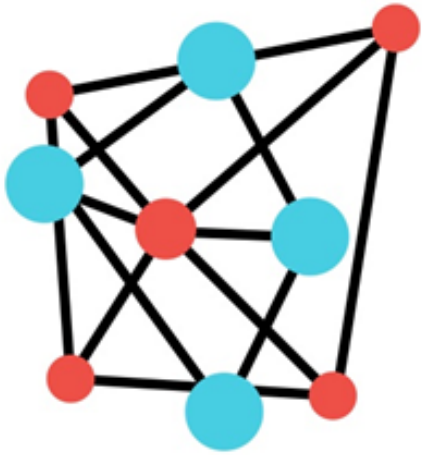
Deep learning refers to a machine learning technique using an architecture comprising some hierarchical layers of non-linear processing stages. Intrusion Detection System using deep neural network for In-Vehicle Network Security can be divided into two types, i.e., a deep discriminative architecture and a deep generative architecture, depending on how the architectures are exploited. The discriminative deep architecture provides abilities for pattern classification with the supervised learning as in the conventional feed-forward artificial neural

networks. The deep structure, namely, deep neural network can be augmented with multiple hidden layers from the artificial neural network structure. However, the augmented neural networks are inefficiently trained using the back-propagation learning with a gradient descent optimization due to the vanishing gradient problem. In the backpropagation, the gradient of the error surface is computed in each layer while the gradient exponentially decreases with the number of the layers, thus causing an extremely slow convergent speed.

To prevent the problem, the deep generative architecture characterizing the correlation of the observed data and the associated classes will be used for initializing parameters of the discriminative architecture, called the unsupervised pre-training scheme. The weight parameters interconnecting nodes in adjacent layers are efficiently trained using a top-down approach by considering the nodes.

After the pre-training, fine-tuning will be performed using the gradient descent method with the supervised learning as in the conventional feed-forward artificial neural network. The deep belief networks as a probabilistic generative model include several layers of stochastic hidden units on top of a single bottom layer of observed data to solve the vanishing gradient problem efficiently.

Quantum Hashing Cryptography



QUANTUM
COMMUNICATION

Blockchain has improved security by using hashes appropriately. However, there is a growing concern that as the performance of computers grows rapidly, hash cryptography can become a limitation. CUBE develops quantum cryptography to prevent malicious attacks against autonomous vehicles. This Quantum Cryptography will contribute not only to the autonomous drive but also to the overall upgrade of the entire Blockchain technology.

CUBE increases the level of security by using the characteristics of Quantum. Security is ensured by quantum's continuously changing characteristics, the characteristic that the data received depends on the angle of polarization, and the characteristics of the data that is destroyed at the moment of malicious attack.

Blockchain's cryptographic hash functions

Blockchain's cryptographic hash functions are used widely in today's cryptographic systems. Ideally, blockchain wants the hash function to have the following properties. First, the digest should be much shorter than the message. Depending

on applications, the following security properties are desirable. (1) Collision resistant: It is computationally infeasible to find a “collision”, i.e., two distinct messages x and x 0, such that h(x) = h(x 0). (2) Preimage Resistance: It is computationally infeasible to invert h. (3) Second Preimage Resistance: Given a message x, it should be computational infeasible to x 0 6= x with h(x 0) = h(x).

In practice, there may be information leakage of the message over time due to information transmission, adversarial attacks, etc. Therefore, it is rather desirable if the hash function is resilient against information leakage. We ask: how many bits ` about the message x can be leaked before the adversary is able to forge the tag h(x) easily? Cleary, ` ≤ m, since if the tag h(x) itself is known to the adversary, he does not need to know more about x to pass the verification. This is rather disappointing, since m is typically much smaller than n. We then ask: what if a quantum tag is used instead? If the leakage is quantum, by the same reasoning, m remains a trivial and rather lower upper-bound on

Quantum hash functions and BlockChain

Quantum key distribution (QKD) is a technique that allows two parties; we can say "Automotive Control Center" and "Autonomous Car," to share a common secret key for cryptographic purposes. In this section, I wish to give a general idea of what QKD is and the techniques it involves. The concepts will be covered in more details in the subsequent chapters.

To ensure the confidentiality of communications, "Automotive Control Center" and "Autonomous Car" agree on a common, yet secret, a piece of information called a key. Encryption is performed by combining the message with the key in such a way that the result is incomprehensible to an observer who does not know the key. The recipient of the message uses his copy of the key to decrypt the message.

A chain then ensures the confidentiality of the transmitted data with two links: the quantum-distributed key and the encryption algorithm. If one of these two links is broken, the whole chain is compromised; hence we have to look at the strengths of both links.

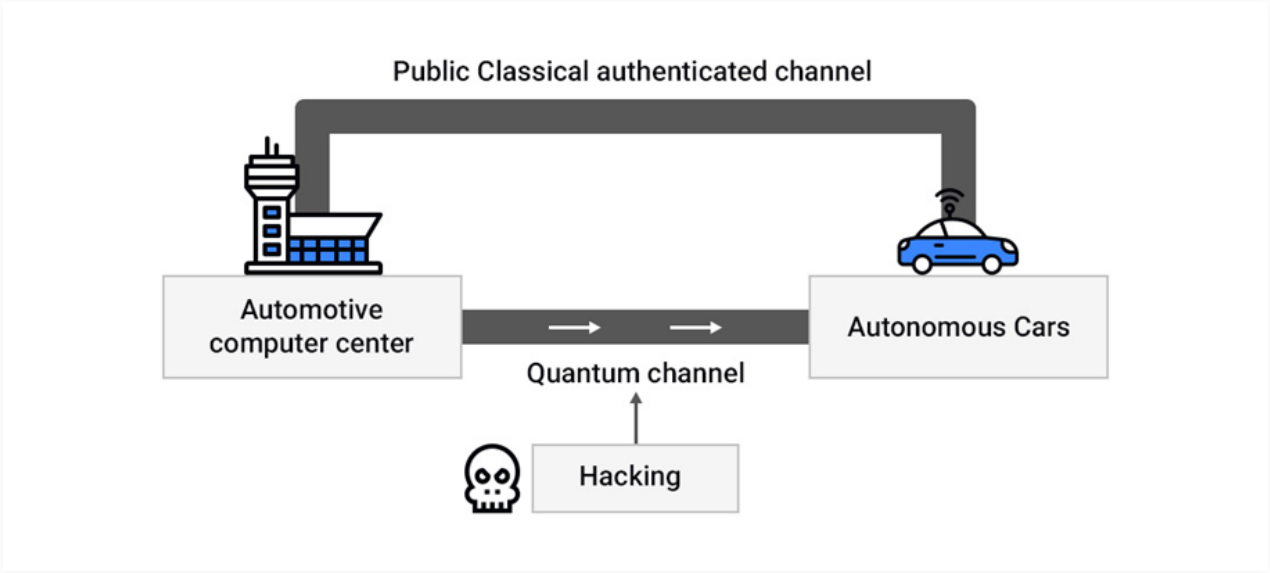
First, how is the confidentiality of the key ensured? The laws of quantum mechanics have strange properties, with the nice consequence of making the eavesdropping detectable. If an eavesdropper, conventionally called "Malicious Attack," tries to determine the key, she will be detected. The legitimate parties will then discard the key, while no confidential information has been transmitted yet. If on the other hand, no tapping is detected, the secrecy of the distributed key is guaranteed.

As the second link of the chain, the encryption algorithm must also have strong properties. As explained above, the confidentiality of data is guaranteed if the encryption key is as long as the message to transmit and is not reused for subsequent messages. This is where quantum key distribution is particularly useful, as it can distribute long keys as often as needed by "Automotive Control Center" and "Autonomous Car."

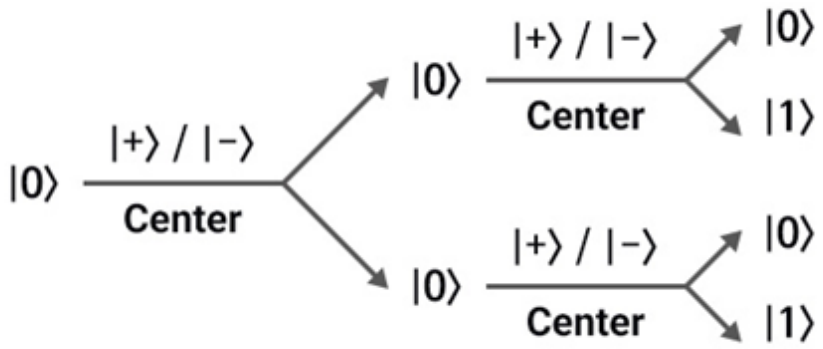
In the quantum carriers, “Automotive Control Center” encodes random pieces of information that will make up the key. These pieces of information may be, for instance, random bits or Gaussian-distributed random numbers, but for simplicity of the current discussion, let us restrict ourselves to the case of “Automotive Control Center” encoding only zeroes and ones. Note that what “Automotive Control Center” sends to “Autonomous Car” does not have to – and may not – be meaningful. The whole point is that an eavesdropper cannot predict any of the transmitted bits. In particular, it may not use fixed patterns or pseudo-randomly generated bits but instead is required to use "truly random" bits.

During the transmission between "Automotive Control Center" and "Autonomous Car," "Malicious Attack" might listen to the quantum channel and therefore spy on potential secret key bits. This does not pose a fundamental problem to the legitimate parties, as the eavesdropping is detectable by way of transmission errors. Furthermore, the secret-key distillation techniques allow "Automotive Control Center" and "Autonomous Car" to recover from such errors and create a secret key out of the bits that are unknown to "Malicious Attack."

After the transmission, "Automotive Control Center" and "Autonomous Car" can compare a fraction of the exchanged key to see if there are any transmission errors caused by eavesdropping. For this process, QKD requires the use of a public classical authenticated channel, as depicted in Fig 6. This classical channel has two important characteristics, namely, publicness and authentication. It is not required to be public, but if "Automotive Control Center" and "Autonomous Car" had access to a private channel, they would not need to encrypt messages; hence the channel is assumed to be public. As an important consequence, any message exchanged by "Automotive Control Center" and "Autonomous Car" on this channel may be known to "Malicious Attack." The authentication feature is necessary so that "Automotive Control Center" and "Autonomous Car" can make sure that they are talking to each other. We may think that “Automotive Control Center” and “Autonomous Car” know each other and will not get fooled if “Malicious Attack” pretends to be either of them.



Quantum key distribution comprises a quantum channel and a public classical authenticated channel. As a universal convention in quantum cryptography, “Automotive Control Center” sends quantum states to “Autonomous Car” through a quantum channel. “Malicious Attack” is suspected of eavesdropping on the line.



So, when “Malicious Attack” tries to eavesdrop, she will get irrelevant results about half of the time and disturb the state. She might decide not to send “Autonomous Car” the states for which she gets irrelevant results, but it is impossible for her to make such a distinction, as she does not know in advance which encoding is used. Discarding a key element is useless for “Malicious Attack” since this sample will not be used by “Automotive Control Center” and “Autonomous Car” to make the key. However, if she does retransmit the state (even though it is wrong half of the time), “Automotive Control Center” and “Autonomous Car” will detect her presence by an unusually high number of errors between their key elements.

Both "Autonomous Car" and "Malicious Attack" have the same difficulties in determining what "Automotive Control Center" sent since they do not know which encoding is used. But the situation is not symmetric in "Autonomous Car" and "Malicious Attack": all the communications required to do the sifting are made over the classical authenticated channel. This allows "Automotive Control Center" to make sure she is talking to "Autonomous Car" and not to "Malicious Attack." So, the legitimate parties can guarantee that the sifting process is not influenced by "Malicious Attack." Owing to this, "Automotive Control Center" and "Autonomous Car" can select only the key elements which are correctly measured.

To detect the presence of an eavesdropper, "Automotive Control Center" and "Autonomous Car" must be able to detect transmission errors. For this, an option is to disclose a part of the sifted key. A given protocol might specify that after a transmission of l+n key elements (e.g., l+n=100 000), numbered from 0 to l+n-1, "Automotive

Control Center" randomly chooses n indexes (e.g., n=1000) and communicates them to "Autonomous Car." "Automotive Control Center" and "Autonomous Car" then reveal the corresponding n key elements to one another to count the number of errors. Any error means there was some eavesdropping. The absence of error gives some statistical confidence on the fact that there was no eavesdropping – "Malicious Attack" might just have been lucky, guessing right the encoding sets or making errors only on the other l key elements. Of course, only the remaining l key elements will then be used to produce a secret key.

If an error is detected, the "Car Control Center" and "Autonomous Vehicle" may decide to abort the protocol because the eavesdropping may cause an error. At least, this prevents the creation of keys that can be known to the enemy. However, this kind of decision can be somewhat strict. In practice, the physical implementation is not perfect, and errors can occur for many reasons other than eavesdroppings, such as noise or loss in the quantum channel, incomplete generation of quantum states, or incomplete detectors. Also, “Malicious Attack” may just eavesdrop a small fraction of the sifted key, making the remaining key elements available for creating a secret key. There should thus be a way to make a QKD protocol more robust against noise.

“Automotive Control Center” and “Autonomous Car” count the number of errors in the disclosed key elements and divide this number by n to obtain an estimate of the expected fraction e of transmission errors in the whole set of key elements; e is called the bit error rate. They can then deduce the amount of information “Malicious Attack” knows about the key elements.

To make the key secret, the idea behind privacy amplification is to exploit what “Malicious Attack” does not know about the key. “Automotive Control Center” and “Autonomous Car” can calculate a function f of their key elements to spread "Malicious Attack"'s partial ignorance over the entire result. The secret key obtained after privacy amplification can be used by “Automotive Control Center” and “Autonomous Car” for cryptographic purposes. In particular, they can use it to encrypt messages and thus create a secret channel.

CONCLUSION

CUBE Technology Market Application Plan

Short-term Strategy – Over-The-Air Market

OTA technology is a technology that can remotely upgrade existing automotive software. Automotive software can improve performance and patch bugs through upgrades. CUBE will provide software diagnostics and installation and bug patches of existing automotive software remotely with OTA technology based on blockchains.

CUBE is developing blockchain based On-Board-Diagnostics (OBD) equipment for existing connected vehicles. Every car has a part to insert an OBD device. Once a vehicle is connected to an OBD device, it can track the vehicle's status from firmware version data to detecting malfunctioning part of the vehicle. This means that vehicles with OBD devices will benefit with time and cost saving effect.

Since firmware updates are performed over the air, car owners do not have to go to car repair shops to get the firmware updated. The vehicle manufacturers can fix the bugs via OTA, while saving enormous amount of money spending on recalling. The main reasons of the recalls are mostly software problem, not the hardware.

However, this OTA technology also has a problem of being controlled centralized. This means that OTA technology is also vulnerable to hacking. If a malicious hacker decides to hack the server while uploading firmware upgrade, it will cause tremendous problem from safety issues to human lives.

Cube produces unhackable OBD devices via decentralization. Since automobiles carry the most valuable matters, perfect security is a matter of the utmost importance.Cube believes blockchaintakes the core value in realizing the perfect security.

Since OBD devices and OTA technology are not very new to the world, Cube expects to announce “Cube OTA Ver1.0” by May of 2018, which is profitable area in relatively short-term basis.

Short- and Long-Term Profit Model

In OTA market, the major customers of Cube will be car manufacturers. By enabling remote control of software in vehicles without safety concerns, Cube continues to fulfill business activities towards core customers and charge OTA fee to manufacturers.

Also in a long term aspects, customer range will be widened once Cube finalizes developing “Cube Autonomous Vehicle Security Model Ver1.0” in 2019. Security platform fee will be charged to the customers. Cube is also planning to engage in affiliate sales to automotive-related stores, such as gas stations, car maintenance shops, automobile supplies stores, car dealers, car-sharing services, carpooling, and public transportation. The merchant will receive the CUBE token, and the merchant who receives the CUBE token and sells the service or the goods in return will be charged anaffiliates fee. The CUBE tokens that gathered in various shape of fee will be reimbursed to the mileage point benefit for users who purchase services or goods with CUBE tokens.In this way, CUBE tokens become an inevitable currency option in the automotive market. Shops will have no other option but to sign up for CUBE affiliation to maximize sales while paying affiliation fees. As a result, the CUBE token will have the real value as currency, and the value will rise even further.