Chapter 6
Cryptographic Techniques for Automotive Technology

Md. Iftekhar Salam and Madhusudan Singh

Abstract Modern vehicles now can be connected over vehicular network that gives the opportunity for building an intelligent transportation system. These networks provide many attractive features to provide comfort and safety. However, this also raises a new set of security concerns. These security features must not be limited to protecting confidential information, but also needs to address safety critical systems such as brake, accelerator or steering etc. this article has presented the cryptography mechanism overview for automotive cybersecurity.

Keywords Automotive cryptography · Stream cipher · Symmetric encryption

6.1 Automotive Cybersecurity

Not long ago, in 2015 two automotive security researchers Charlie Miller and Chris Valasek demonstrated a remote attack on the Jeep Cherokee by taking control of the Jeep's brakes and accelerator. It turns out that the researchers exploited a vulnerability in the Jeep's infotainment system over a cellular network. Imagine an attacker hacking into the electronic component of the vehicle to get access to the control of the braking system; the result would be devastating. Addressing the security of vehicular network is a challenging issue [1]. In Fig. 6.1, we have shown need of security in vehicle such as vehicle location, communication channel (Wi-Fi, cellular networks etc.) embedded insurance aggregator, vehicle valuation, driving behavior of vehicle, vehicle tracking, virtual breakdown on-demand, deals accident information, and virtual mechanic.

Cryptographic techniques play an important role to provide different security features in a vehicular network. Security of such network depends on various things,

Md. I. Salam
Department of Information and Communication Technology, Xiamen University, Xiamen, Malaysia

M. Singh (✉)
School of Technology Studies, Endicott College of International Studies, Woosong University, Daejeon, Republic of Korea
e-mail: msingh@wsu.ac.kr

M. Singh, *Information Security of Intelligent Vehicles Communication*,
Studies in Computational Intelligence 978,
https://doi.org/10.1007/978-981-16-2217-5_6

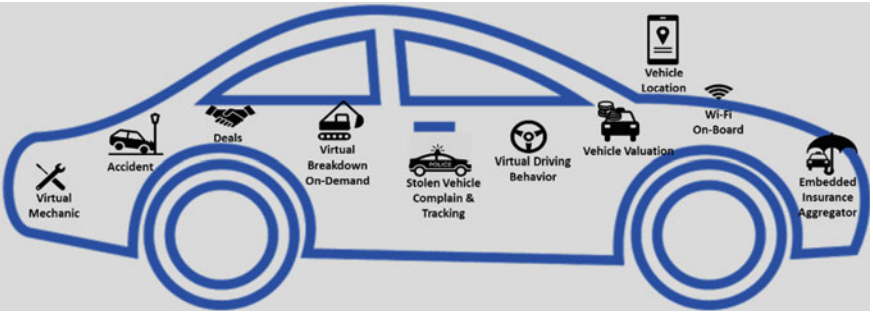


Fig. 6.1 Need of security in vehicle issues

but cryptography plays an important role as the basic building block for providing security in vehicular networks. Cryptographic techniques can be used to support the security goals such as authentication, confidentiality, integrity assurance etc., in V2X communications, where a vehicle is communicating with other vehicles (V2V) or roadside infrastructure (V2I) [1]. Cryptography bolsters the validation that permits these applications to confide in each other, which clearly is key since they include human lives in huge amounts of metal moving at high speeds. Crypto algorithms are additionally an incredible method to encrypt V2X interchanges. A model is guaranteeing that a vehicle's continuous area data has not been controlled. Without cryptography, it's conceivable that a programmer could send counterfeit message that could cause, for example, mishaps by activating programmed slowing down at high speeds.

In general, there are mainly two types of cryptographic primitives: symmetric primitives and asymmetric primitives. Symmetric cryptosystems are commonly used for providing security services of confidentiality and integrity assurance. Asymmetric cryptosystems are more commonly used for providing user authentication and key distribution.

- ***Confidentiality*** can be achieved using a secure symmetric key cipher that provides encryption and decryption functionalities. These symmetric ciphers are mainly categorized into two types: block ciphers and stream ciphers. Over the years there have been several developments of secure block ciphers and stream ciphers. Currently, the most common block cipher algorithm is Rijndael, which was selected as the Advanced Encryption Standard (AES). Block ciphers have different modes of operation to provide different services. Examples of block cipher mode of operations providing confidentiality include AES Cipher Block Chaining (AES-CBC) mode [3], AES Counter (AES-CTR) mode. Similarly, stream ciphers can also be used to provide confidentiality. Examples of stream cipher-based confidentiality algorithms include Salsa20, Trivium.
- ***Integrity*** assurance can be achieved using a message authentication code (MAC) algorithm or using a cryptographic hash function. There have been several developments of secure algorithms, e.g., Hash-based Message Authentication Code

(HMAC), Poly1305, to provide integrity assurance of the transmitted data. Modes of block cipher, e.g., Cipher Block Chaining MAC (CBC-MAC), can also be used to provide integrity assurance. Similarly, there are some stream cipher-based constructions, e.g., ZUC, which provide integrity assurance.

Generally, stream ciphers are faster than their block ciphers counterpart, and are most suitable for real time application. On the other hand, block ciphers are more suitable for processing bulk amount of data such disk encryption. In regard to the security goal of confidentiality stream cipher may seem to be a more suitable choice for automotive cybersecurity. This is due to requirement of several facts such as real time processing, low computational overhead, and faster processing.

The rest of the chapter is organized as follows. Section 6.2 presents an overview of symmetric cryptosystem-based stream ciphers. Section 6.3 discusses current research trend on symmetric ciphers. Section 6.4 discussed standard IEEE 1609.2. Section 6.5 presents the future of automotive cryptography. Section 6.6 conclude an overview of current automotive cryptography and future research directions.

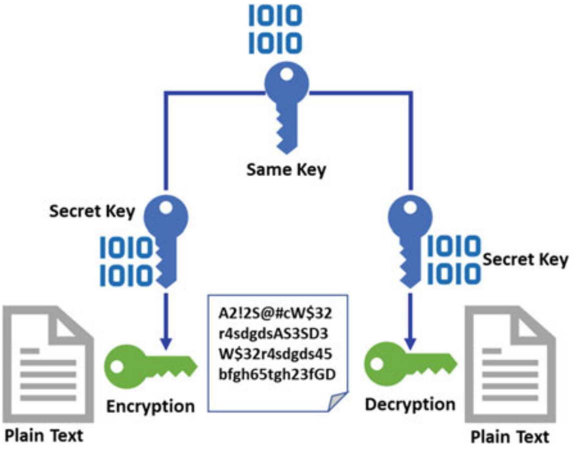
6.2 Symmetric Cryptosystem Based on Stream Ciphers

Stream ciphers are broadly utilized in the cryptographic algorithms for giving confidentiality to negotiate the transmission of data between two clients. A stream cipher ordinarily partitions the message into progressive characters and works on each character independently to encrypt/decrypt the message. In view of the size of the character, a stream cipher can be either bit based or word based depending on the length of the character and its size. In the bit-based stream cipher, the cipher works on each bit independently. In the word-based stream cipher, each character comprises of a gathering of bits called a word and the cipher works on these words to encrypt/decrypt a message. Before going to the details of a stream cipher [2]. Figure 6.2 has shown the process of symmetric cryptosystem based on stream ciphers.

6.2.1 Notation and Terminology

- ***Keystream generator*** A component that generates pseudo-random characters.
- ***Secret key, 'K'*** An input component which entered in the keystream generator, which is only known to the sender and receiver.
- ***Initialization vector (IV), 'V'*** An input to the keystream generator, which is usually publicly available information. The initialization vector usually varies from message to message.
- ***Keystream, 'Z'*** Stream of output bits/words from the keystream generator.
- ***Plaintext, 'P'*** Stream of plaintext message bits/words before encryption.
- ***Ciphertext, 'C'*** Stream of the ciphertext message bits/words after encryption.

Fig. 6.2 Process of symmetric cryptosystem



- ***Associated Data, 'D'*** Stream of the associated data bits/words. This part of a message does not require confidentiality; but requires integrity assurance.
- ***Message, 'M'*** Message can be either plaintext or ciphertext or associated data.
- ***Tag, 'τ'*** A specific length sequence generated by the tag generation algorithm. The tag is computed on the message value and is used to determine whether the message has been modified during transmission.
- ***Encryption algorithm*** A process that converts the plaintext into ciphertext using a secret key.
- ***Decryption algorithm*** A process that converts the ciphertext into plaintext using a secret key.
- ***Internal state*** Memory locations where information is stored.
- ***Internal state size*** Amount of information that the internal state of the cipher can hold.
- ***Initialization*** The initialization phase loads and disuses the key and initialization vector are stacked into the internal state of the keystream generator. The state obtained after the initialization procedure is called the initial state.
- ***State updates function*** the process to update the contents of the internal state.
- ***Output functions*** the process to compute the keystream bits using the contents of the internal state.

6.2.2 *Operations to Provide Confidentiality*

Confidentiality guarantees that the message isn't unveiled to an unauthorized entity. This can be accomplished by utilizing an encryption/decryption algorithm. Here, we quickly characterize the encryption/decryption method for accomplishing confidentiality utilizing a stream cipher. Figure 6.1 shows the general development of a stream cipher. Initially, the secret key, K, and initialization vector, V are stacked into

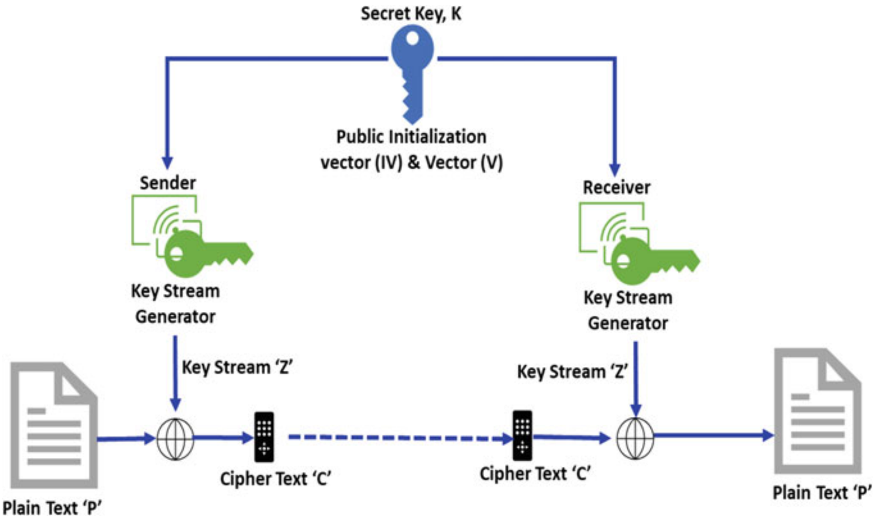


Fig. 6.3 Confidentiality using stream cipher

the internal state of the keystream generator as a major aspect of some initialization phase [3]. Following this, the keystream generator is operated for a predetermined number of iterations without creating any keystream bits. After the initialization phase, the internal state of the keystream generator comprises the initial state and is prepared to produce the keystream bits.

For keystream generation, the internal state of the keystream generator is refreshed utilizing a state update function and a keystream bit/word, 'Z', is computed at every iteration utilizing the output function of the cipher. The keystream generator is utilized to generate a keystream sequence by repeating this procedure. In conclusion, as appeared in Fig. 6.3, the encryption algorithm utilizes a combining function to consolidate the keystream Z with the plaintext P and outputs the ciphertext 'C'. Regularly, the keystream is combined with the message utilizing bitwise XOR function. Stream ciphers using the XOR function as the combining function are called binary additive stream ciphers. Upon encryption of the plaintext, the ciphertext is transmitted through the insecure channel. At the receiver end, the keystream is generated in a similar fashion and then the decryption algorithm combines the ciphertext C with the keystream Z to retrieve the plaintext.

6.2.3 Operations to Provide Integrity Assurance

Integrity assurance of a message gives the beneficiary an assurance that the information has not been altered during transmission. Data integrity assurance can be accomplished by producing a Message Authentication Code (MAC) tag. Here, we quickly portray a method for accomplishing integrity assurance utilizing a stream figure. In a stream cipher-based MAC tag generation scheme as appeared in Fig. 6.4, the input message M is collected into the internal state of the cipher subsequent to performing out the initialization phase. Following this, the cipher is iterated for a predefined number of steps in the finalization phase without creating any output bits. Toward the end of the finalization phase, the tag generation function takes input from a portion of the internal state bits and output the MAC tag τ [4].

The MAC tag is affixed and afterward transmitted with the message. Upon receipt of the message, the receiver processes the MAC tag τ′ for the got message M′ and compares it with the received MAC tag. On the off chance that the got MAC tag coordinates the registered MAC tag at the recipient (τ = τ′), at that point the receiver expects that the message has not been adjusted during transmission. In the event that the got MAC tag doesn't coordinate the MAC tag computed by the recipient, at that point the receiver expects that the message has been adjusted during transmission through the channel and therefore disregard the received message.

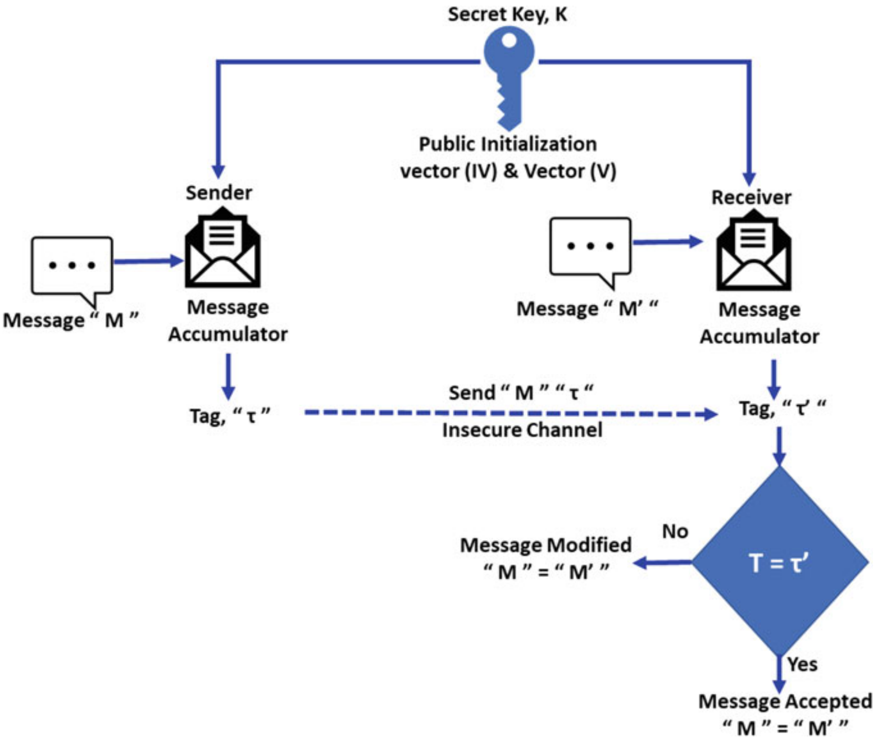


Fig. 6.4 Integrity assurance using stream cipher

6.3 Current Research Trend on Symmetric Ciphers

There are several cases where both confidentiality and integrity assurance are needed. The idea of providing both data confidentiality and integrity assurance is not new. A scheme that provides both security goals is called an authenticated encryption (AE) scheme. Individual algorithms can be combined to construct an authenticated encryption scheme. A generic composition to shape such authenticated encryption conspire was portrayed by Bellare and Namprempere, and Katz and Yung. These generic compositions are two-pass schemes, in this manner require two passes over the data: one giving secrecy and the other, integrity confirmation. These AE schemes also require two different keys, one for each of their component mechanisms. The computational cost of the two-pass scheme is about twice of the single pass scheme.

Authenticated encryption is one of the hot research topics these days. More importantly, we require authenticated encryption for lightweight devices such as sensor network, IoT devices. The lightweight cryptographic primitives will preferably be suitable for the automotive cybersecurity as well.

In view of this, starting from 2013 the “Lightweight Cryptography Project” was initiated in view to to design lightweight cryptography by National Institute of Standards and Technology (NIST). The aim of the lightweight cryptography project is to evaluate and standardize cryptographic algorithms which are suitable for resource constrained environment. This standardization process aims to select a portfolio of secure and efficient lightweight cryptographic algorithms.

6.3.1 Asymmetric Cryptosystem

Asymmetric cryptography [5], otherwise called public key cryptography is for the most part utilized for key conveyance and to give the security administrations of non-repudiation and client verification in vehicular networks. Asymmetric cryptosystems can likewise be utilized to give the security objective of confidentiality; nonetheless, slower and resource-hungry similar to the symmetric cryptosystem as shown in Fig. 6.5.

In this scheme, a couple of keys are utilized: a public key and a private key. In spite of the fact that the keys are different, the keys are numerically related. One key is called public key since it is known to everybody, while the other one is called private key just known to the proprietor. Consequently, Asymmetric Key Cryptography is otherwise called Public Key Cryptography.

- **Asymmetric cryptography for providing confidentiality**
  In this plan, each client must have a couple of various keys: a private key, and a public key. One implied for encryption of the information while the other implied for its decryption. The public key is placed into a public storehouse while the private key is put away as a mystery. Even though public and private keys are mathematically related, it is computationally unrealistic to get one from another.

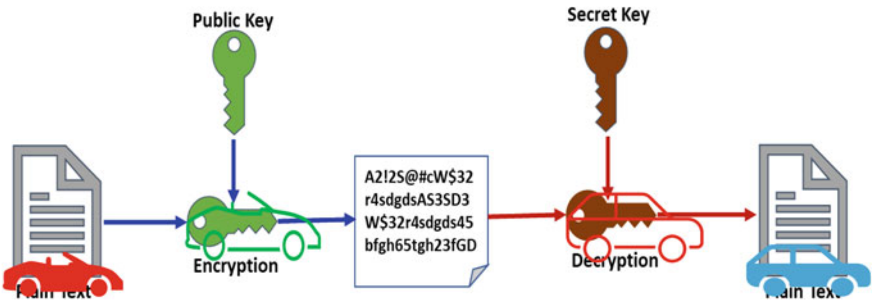


Fig. 6.5 Asymmetric cryptosystem

At the point when a client needs to send information to another client, at that point, they get the public key of another client from the store, encrypts the information, and afterward transmits it. Another client at that point utilizes their private key for decryption.

One inquiry you may pose, that, how these keys are identified with one another but then, it is difficult to get one from another. The appropriate response lies in mathematical ideas and concept. Numerically, it is conceivable to plan a cryptosystem whose keys have this property. Symmetric cryptography was fine government and military organizations, however with the spread of secure computers networks, an alternate sort cryptography strategy was required to address the issue as shown in Fig. 6.6.

- **Asymmetric cryptography for user authentication**
  A Digital Signature can be produced utilizing asymmetric cryptography to give client verification and non-repudiation in vehicular systems. This is a method that connects an element with the digital data which is supposed to transmit and receive. This affiliation is freely verifiable by the recipient just as any outsider or third-party interaction. Digital Signature is a cryptographic worth which can be determined from the data and some secret (key shared or belonged) known to the

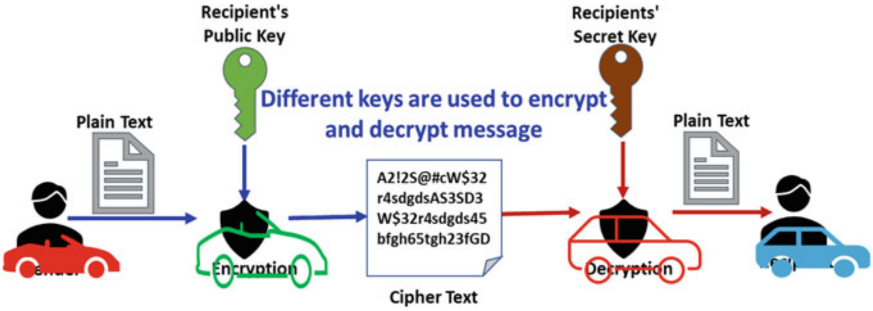


Fig. 6.6 Encryption and decryption of asymmetric key encryption

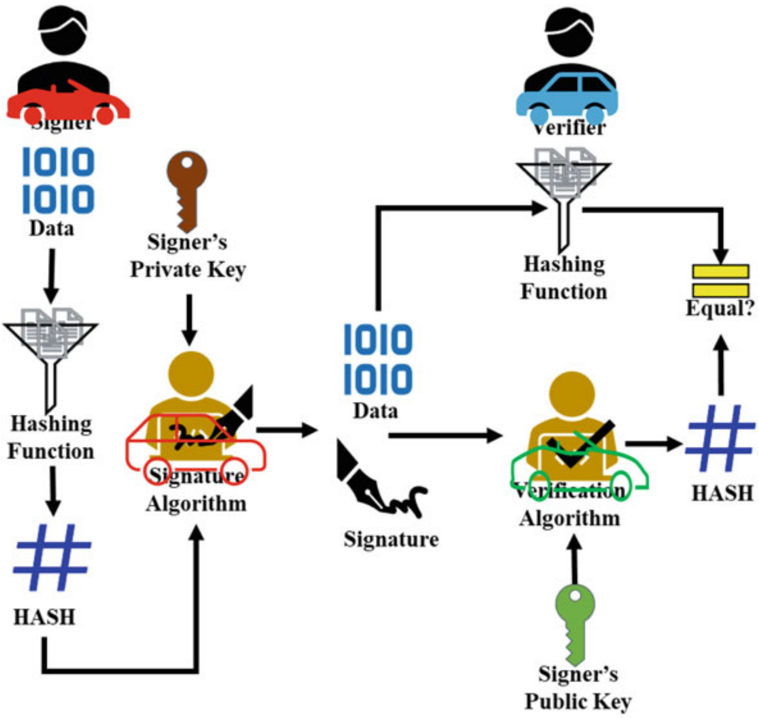


Fig. 6.7 Model of digital signature

signer. The digital signature scheme depends on public-key cryptography. The model of Digital Signature scheme is illustrated in the following Fig. 6.7.

Each beneficiary associated with this plan has an open private key pair. The private key utilized for signing is named as 'signature key' while the Public key is named as 'verification key.' Signer gives data to the hash function and creates a hash value and this hash value id directed to the signature algorithm. Hash value as well as signer's private key called signature key are then given to the signature algorithm which delivers the digital signature of the given hash [6].

At the receiver end, the verifier at that point provides a digital signature and the verification key is also known as the public key into the verification algorithm, and output is acquired. Verifier additionally runs a similar hash work on got data to produce a hash value [7]. At that point, this hash value and output from the verifier algorithm are looked at and compared. Furthermore, in light of the compared result, the legitimacy of the digital signature is checked.

Since Digital Signature is made by the private key of the sender and nobody else can have this key. Along these lines, the sender can't repudiate marking the data in the future. It ought to be seen that the hash of the data is utilized for signing the as opposed to signing the data itself. Utilizing hashes improves the productivity of the scheme.

6.4 IEEE 1609.2 Cryptography Mechanism

The IEEE 1609.2 [8] standard defines the use of the encryption techniques described below for the security of messages to be communicated.

For the sake of convenience in the following description, when communicating between two vehicles or a vehicle and an infrastructure, it is assumed that the entity transmitting the message is W_A and the entity receiving the message is W_B. We have use W_A and W_B as two communication nodes in description for IEEE 1609.2 cryptography mechanism at below.

6.4.1 Digital Signature

When W_A sends an authenticated message to W_B, W_A sends a message attaching a digital signature, which is a cryptographic checksum generated using its own private key, W_B can confirm that the message is sent from W_A by verifying the received message and the digital signature as the public key of W_A.

Such a digital signature is a technique particularly effective for security in communication with an entity that does not have a previous connection instance, such as broadcasting in a dynamically changing environment.

The 1609.2 standard backings the Elliptic Curve Digital Signature Algorithm (ECDSA) characterized as a computerized signature technology and alternatively remembers extra data for the digital signature.

6.4.2 Asymmetric Encryption Algorithm

The asymmetric encryption algorithm permits clients who don’t share the secret key to impart safely. In this technique, a couple of a private key and a public key are utilized. A public key is revealed to a user taking an interest in the communication; however, a relating private key can be known uniquely to the owner.

When W_A encrypts a Message to be sent to W_B using the public key of W_B and then transmits it, W_B decrypts it using its corresponding private key. Generally, asymmetric cryptosystems have a disadvantage in that computation is more complicated than symmetric cryptosystems, and the method is used as a symmetric cryptosystem algorithm to increase efficiency. That is, W_A encodes a message utilizing a haphazardly produced secret key, encodes the secret key with the public key of W_B, and transmits the encoded message along with the message. In the asymmetric encryption procedure, just the short secret key is encoded, Symmetric encryption technology encrypts the entire message, which can increase efficiency.

6.4.3 Key Generation and Key Validity

ECDSA and ECIES, the private key and public key utilized are created by the definition in FIPS 196 − 3 * 9 as a key pair, and the utilization of a top-notch irregular number generator is requested.

The validity of the key pair is determined by IEEE Std. It is evaluated according to the criteria defined in 1363 – 2000 * 10.

6.4.4 Symmetric Encryption Algorithm

W_A may encrypt and transmit a message using a secret key, and W_B may decrypt the received encrypted message using the same cryptographic key. It also uses cryptographic keys to generate cryptographic checksum or message integrity check (MIC) to ensure authenticity and integrity.

The 1609.2 [8] standard uses AES-CCM (the propelled encryption standard-counter with cipher block chaining instrument) as the Symmetric algorithm, and AES-CCM is characterized in detail in NIST SP 800 – 38 C * 9 [9].

6.4.5 Implicit Certificate

Verifiable Certificate is a variation of a public-key certificate, which is a data structure containing the computerized signature esteem created by the certificate authority's private key for cryptographically and safely interfacing the identity data of the authentication holder and the public key.

Utilizing a digital certificate is known as the most ideal approach to build up identity in network information interchanges, and the authentication gives connectivity between the individual data and the public key. The key pair is utilized for the 7 digital signature required for key trade and approval of exchanges to set up secure communications. Along these lines, digital certificates are fundamental components in public key infrastructure (PKI).

Since the conventional public-key certificate contains a duplicate of the public key and the digital signature of the affirmation authority, the public key in the confirmation of the digital signature is utilized as in the individual recognized in the certificate knows the related private key and is the main party that realizes the private key expressly affirmed. It is along these lines called an explicit certificate. On account of a verifiable certificate, the public key is remade from the certificate, and the main party that realizes the related private key is called implicit regarding the individual recognized in the certificate.

In the case of an explicit certificate, the size of the certificate can be quite large, the infrastructure for the used key protection, the memory for storing and manipulating certificates, and the significant investment in securing the bandwidth required for repeatedly sending certificates to multiple objects need. On account of an implicit certificate, the encryption part of the certificate is extensively littler than the proven exact certificate. limited certificates are exceptionally helpful in situations where there are restricted assets where a great deal of memory or bandwidth isn't accessible. IEEE Std.

In Europe, the European Telecommunications Standards Institute (ETSI) has characterized the security prerequisites for the between vehicle communication dangers utilized in ITS through specialized report TR102893 [10] as follows:

- ***Integrity [In]***

Objective		Functional security requirements
ID	Text	
In1	Data in ITS-S needs to be secured from unwanted or illegal change	Only allows authorized applications to modify or delete ITS-S security parameter and LDM information
		Only the above and authorized users can modify or remove service profile information
In2	Information transferred or received by a registered ITS user to be protected during transmission	An ITS-S shall implement one or more methods to enable it, if requested by an ITS user
In3	Management Information inside an ITS-S to be secured	In1 satisfies requirement
In4	Management Information transferred or received by ITS-S to be protected during transmission	Ln2 satisfies requirement

- ***Availability [Av]***

Objective		Functional Security Requirements
ID	Descriptions	
Av1	Any malevolent action in the ITS-S ought not to confine access to and the activity of ITS services by approved clients	must have ability to detect attack patterns of denial of service

- ***Confidentiality [Co]***

Objectives		Functional security requirements
ID	Description	
Co1	Parties without authorization (non-users) must not have information revealed to them	Assigning tags of restriction on information
		Restricted information to be encrypted
		Authentication from recipient
		Authentication needed from sender for recipient
Co2	Information in ITS-S to be kept away from someone without access	Only allow ITS application to access information
		Only authorized users to have access
Co3	Data on identity and administration capacities of an ITS client not to be uncovered to any outsider	Same as Co2
Co4	The executives Information moved to be shielded from individuals without approval	Only users can get management information
		Only get information from legitimate source
Co5	The management Information in ITS-S to be shielded from individuals without approval	entry restricted to only some users with authorization
		Provide a way for user to get access
Co6	By tracking communication between users, it should be impossible to locate them	User identity to be kept secret and not be added to location data in unlimited multicast address
		The above may be permissible for unicast or limited multicast address
		While transmitting should protect information
Co7	By tracking communication between users, it should be impossible to calculate the route taken	Shall have option to use multiple identifiers
		When used there will be no link between the identifiers

- ***Accountability [Ac]***

Objective		Functional security requirements
ID	Descriptions	
Ac1	Changes in security application and parameters should be auditable	All changes and request for them in security shall be recorded

- ***Authentication [Au]***

Objective		Functional security requirements
ID	Text	
Au1	Should not be possible for unauthorized users to alias themselves as legitimate users of ITS- S	Only authorized ITS-S get access to services
		Can validate identity of vehicle in the emergency
		In the emergency vehicle it should be possible to get temporary access to services
		ITS-S should be able to identify itself to emergency vehicle
		ITS-S will only be allowed to send massages if authorized in the situation
		Unauthorized messages to be ignored
Au2	Should not be workable for an ITS-S to get and process the management and configuration data from an unapproved client	In1 and Au2 satisfy requirement
Au3	Restricted ITS services like emergency warnings to be available only to authorize ITS users	Only currently authorized users will be allowed to transmit messages
		Authorizations will be either time limited or can be explicitly removed

6.5 Future of Automotive Cryptography

Automotive cybersecurity is a very important issue as this can be linked with the safety of the vehicles on the road. Lightweight cryptographic primitives could be the potential answer to ensure automotive cybersecurity. Customary cryptographic calculations, for example, public key foundation, elliptic curve cryptography, HASH functions, and symmetric key cryptography may not be applied legitimately in vehicular systems because of their high portability and dynamic system topology. Currently National Institute of Standards and Technology (NIST) has published a call for algorithms to be considered for lightweight cryptographic project. These lightweight cryptographic cipher proposals will go through a thorough three round evaluation phases over the next few years. These evaluation phases will consider both public analyses of the algorithms and the analyses provided by NIST internal committee.

6.6 Conclusion

In this article, we can get to know about the cryptography mechanism that can be used in automotive cyber security such as symmetric, asymmetric cryptography, we have get the overview of stream cipher, encryption and decryption of asymmetric key encryption an also digital signature. National Institute of Standards and Technology (NIST) is working on what's called post-quantum cryptography: public key cryptography that's resistant to quantum computers. So, we need ways of updating cryptography in vehicles such that when post-quantum options become available, they can be implemented in all vehicles rather than just new ones. The open acknowledgment for new innovation in vehicular systems must be guaranteed by advancing the security and protection of users.

Acknowledgements This research was funded by Woosong University Academic Research in 2021.

References

1. A.K. Jadoon, L. Wang, T. Li, M.A. Zia, *Lightweight Cryptographic Techniques for Automotive Cybersecurity*, vol. 2018. https://doi.org/10.1155/2018/1640167
2. M. Singh, Secure ID-based routing data communication in IoT. EAI Endorsed Trans Internet Things 18(6), 153566. ISSN 2424-1399. https://doi.org/10.4108/eai.15-1-2018.153566
3. B. Naik, D. Singh, A.B. Samaddar, H.-J. Lee, Security attacks on information-centric networking for healthcare system, in *19th International Conference on Advanced Communication Technology (ICACT)*, South Korea, Feb. 19–22, 2017, pp. 436–441
4. E. Schoch, F. Kargl, On the efficiency of secure beaconing in VANETs, in *3rd ACM Conference on Wireless Network Security (WiSec 2010), Proceedings*, March 2010
5. M.I. Salam, K.K.-H. Wong, H. Bartlett, L. Simpson, E. Dawson, J. Pieprzyk, Finding state collisions in the authenticated encryption stream cipher ACORN, in *Proceedings of the Australasian Computer Science Week Multiconference (ACSW'16).* Association for Computing Machinery, New York, NY, USA, Article 36, pp. 1–10. https://doi.org/10.1145/2843043.2843353
6. M. Singh, I. Singh, IEEE E-Learning, securing intelligent transportation systems (2020). https://ieeexplore.ieee.org/courses/details/EDP587
7. M. Singh, Requirement engineering for intelligent vehicles at safety perspective. EAI Endorsed Transactions on Smart Cities, **2**(6), (2017)
8. IEEE standard for wireless access in vehicular environments–security services for applications and management messages, in *IEEE Std 1609.2-2016 (Revision of IEEE Std 1609.2-2013)*, pp. 1–240, 1 March (2016). https://doi.org/10.1109/IEEESTD.2016.7426684
9. M. Dworkin, Recommendation for block cipher modes of operation: The CCM mode for authentication and confidentiality, National Institute of Standards and Technology (NIST), Technology Administration U.S. Department of Commerce, NIST Special Publication 800-38C-[Updated 2007]. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf
10. ETSI TR 102 893 - V1.2.1 - Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA) (2017). https://www.etsi.org/deliver/etsi_tr/102800_102899/102893/01.02.01_60/tr_102893v010201p.pdf