Context-Bound Cybersecurity Framework for Resisting Eavesdropping in Vehicle Networks

Longjiang Li[1(✉)], Bingchuan Ma[1], Yonggang Li[2], and Yuming Mao[1]

[1] Department of Network Engineering, SICE,
University of Electronic Science and Technology of China,
Chengdu 611731, China
{longjiangli,ymmao}@uestc.edu.cn, 806166563@qq.com
[2] Chongqing University of Posts and Telecommunications,
Chongqing 400065, China
lyg@cqupt.edu.cn

Abstract. Wireless channels that are widely adopted between autonomous vehicles are vulnerable to eavesdropping or interferences, so that attacks on cybersecurity may lead to serious consequences, such as losing control of vehicles. In particular, the cryptographic methods used for information security rely on the strict privacy of keys, which is often difficult to guarantee in a wireless environment. This paper proposes a context-bound cybersecurity framework, which protects communication from eavesdroppers by encrypting critical data with a dynamic context among vehicles. The context is synchronized among vehicles through a progressive encoding method, which makes it difficult for third parties to learn the entire context by eavesdropping through the channels, especially in the case of mobility. The normal vehicles may extract a security key from the context to encrypt and decrypt key data, but it is impossible or overwhelmingly expensive for the third parties to decode the data transmitted due to the lack of the context. Besides, the proposed framework also provides a promising way to resist the upcoming quantum computers, because it will become more and more difficult for third parties to collect the complete context as the context continues to update.

Keywords: Cybersecurity · Context · Vehicle networks · Cryptography

1 Introduction

With the development of 5G and Internet of Things(IoT), the boundaries of communication are increasingly blurred, making data-secure sharing face unprecedented challenges, including message cracking [35], data leakage, data tampering,

Funded by the National Natural Science Foundation of China (61273235), and the Defence Advance Research Foundation of China (61400020109).

H. Gao et al. (Eds.): ChinaCom 2020, LNICST 352, pp. 519–536, 2021.
https://doi.org/10.1007/978-3-030-67720-6_36

integrity sabotage, unauthorized access, etc. [16,25,29]. As one of the important application scenarios in IoT, autonomous driving has intensely high requirements for data security, and consequently faces the same dilemma. Especially, autonomous vehicles take wireless signal as the main access carrier, which makes the data transmission process face various threats of attack, such as channel monitoring, information content tampering, counterfeiting, man-in-the-middle forwarding, blocking and replaying [3,11,25,32].

Researches on secure communication can be divided into two categories: physical layer security and network layer cryptography. To approach the secrecy capacity, the former requires a wiretap code whose length is infinite [21], which limits its practical application. In contrast, the latter has no such restriction. Related technologies [32] mainly protect data security by integrating various security cryptography algorithms into data distribution, routing and storage procedures. According to the implementation method, the secure cryptography algorithm can be divided into public key cryptography, symmetric cryptography, anti-quantum cryptography, chaotic cryptography, quantum cryptography, lattice-based cryptography, etc. [8,10,27,30,33]. However, with the development of quantum computers [4], traditional security cryptography methods, such as RSA, are in danger of being cracked. Although post-quantum cryptography [14] has the potential to resist quantum computer attacks, it requires a lot of computational overhead [8,14], which is arduous to apply in a vehicle network with limited computing power [20]. Besides, from a mathematical point of view, most of the existing mainstream security cryptography methods are essentially based on computational complexity problems, and their effectiveness depends only on the privacy of the key [28]. Once the key is cracked or compromised, the communication is no longer secure. Moreover, there are great key exposure risks in the process of key generation, distribution and use due to the emergence of side-channel attacks [7,18,19], cold boot attacks [15], etc. Especially, in the scenario of virtual car platoons, many cars and devices are connected to the network, so that the distribution and management of keys are extremely challenging [4,9,12].

In order to solve those security problems, we propose a context-bound cybersecurity framework (CBCF) for vehicle networking scenarios, which is inspired by the application of blockchain technology that chains records for immutable transactions [17,24]. The main idea to bind the coding process to a cybersecurity context that is a chained structure of private information shared between the source and the sink. The mobility of vehicles brings huge dynamics to the communication process, making it virtually impossible or extremely expensive for a third party to completely grasp the communication context, so that the possibility of key exposure is greatly reduced. On the contrary, through progressive coding, the members of a vehicle network can synchronize the context in time, so as to use the correct key to encrypt or decrypt messages. In essence, the framework utilizes the difficulty of the third party in grasping the whole context to overcome the key exposure problem, and further provide support for cybersecurity in the vehicle network.

Our main contributions are summarized as follows:

- We propose a context-bound cybersecurity framework that can be extended in multiple ways to solve the key exposure problem and protect the data security in vehicle networks by dynamically binding the cybersecurity context.
- Two typical implementations of the proposed framework are given as iterative functions. Besides, we demonstrate how to use the progressive coding method to synchronize the context between source and sink vehicles, and a theoretical analysis for the performance of these two implementations is also provided.
- We conducted experiments to illustrate the ciphering and deciphering process of our framework in vehicle networks. The results show that the ciphertext has good randomness and statistical correlation characteristics with the plaintext.

The rest of the paper is structured as follows. Section 2 presents the system model and introduces the basic idea. In Sect. 3, two typical implementations of context-bound cybersecurity framework are given, whose cybersecurity and performance are also analyzed. Section 4 simulates the encrypted communication process in vehicle networks, and evaluates the statistical characteristics of plaintext and ciphertext. Section 5 discusses the framework's resistance to attacks, scalability, and it's relationship with blockchain. Finally, Sect. 6 concludes the paper.

2 Proposed Context-Bound Cybersecurity Framework

In this section, we first describe the system model and then propose the context-bound cybersecurity framework for vehicle networks.

2.1 Typical Vehicle Networking Scenarios

We consider the problem of encrypted communication in the cross roads scenario, as shown in Fig. 1. This model consists of three major components: Road Side Unit(RSU), vehicles equipped with On Board Unit(OBU) and eavesdroppers.

- RSU: When two of the vehicles cannot communicate directly due to problems such as distance or channel quality, the RSU will serve as a relay node to help them maintain the continuity of the communication process. In order to provide promising services and avoid channel congestion, RSUs are usually deployed at intersections or on roads with heavy traffic [2].
- Vehicles: Vehicles can be regarded as high-speed mobile nodes equipped with OBUs, which makes them capable of communicating with other vehicles [36]. In a vehicle network, every vehicle will establish context-bound security channels with each of the others to maintain the synchronization of information inside the network, and resist third-party eavesdropping attacks at the meanwhile.
- Eavesdroppers: Fig. 1 demonstrates two types of eavesdropper, mobile vehicle eavesdropper E_v, and roadside fixed eavesdropper E_p. The former can eavesdrop on the communication messages of other vehicles during the movement, while the latter can only intercept an exceedingly small part of the messages briefly beside the road.

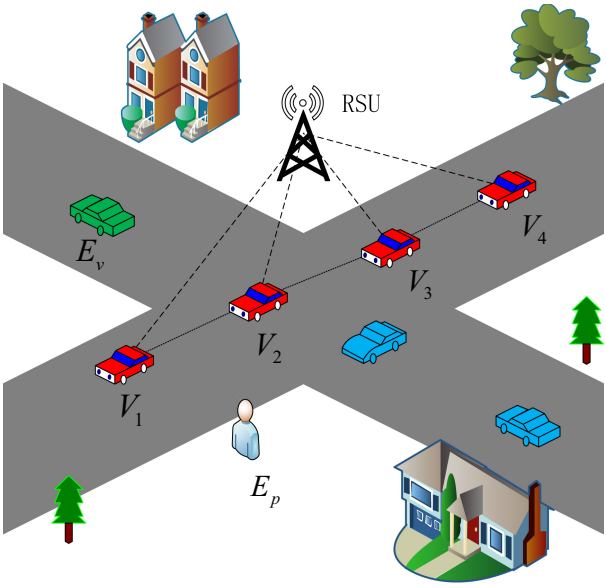


Fig. 1. Encrypted communication in vehicle networking scenarios

2.2 The Main Idea

As shown in Fig. 1, each communication process consists of a source vehicle, a dynamic wireless channel, and a sink vehicle, in which the source sends a series of messages to the sink through the channel. The typical solution is to encrypt the transmitted message by employing a symmetric encryption method, such as AES, with a series of privacy keys that may be distributed by public key cryptography, such as RSA, or physical layer key generations. However, keys may be leaked to eavesdroppers due to various attacks or algorithm limitations. Once the keys are compromised, any messages encoded with these keys are no longer secure to eavesdroppers.

In order to overcome these problems, the main idea is to bind the encrypting and decrypting process to a cybersecurity context. As shown in Fig. 2, the cybersecurity context is constructed by collecting various private random information, such as keys, shared by both parties. For an eavesdropper, it is much harder to master the dynamic context than to steal a single key, so the security level of the cryptosystem is greatly improved.

To simplify the presentation, we assume that the channel is error-free, reliable and orderly. We call the cybersecurity context as Sink Anchor (SA), because the context represents some kind of information that should only be fully accessible between the sink and the source. The message sequence is represented as a series of variables $<x_0, x_1, ..., x_{i-1}, x_i, ...>$, where x_i is called a Sink-Anchoring Coding unit (abbreviated as a SAC unit), denoted by $x_i \in \Lambda$. Sink Anchor (SA) is denoted by $\psi \in \Psi$. The encoding process at the source and the decoding process

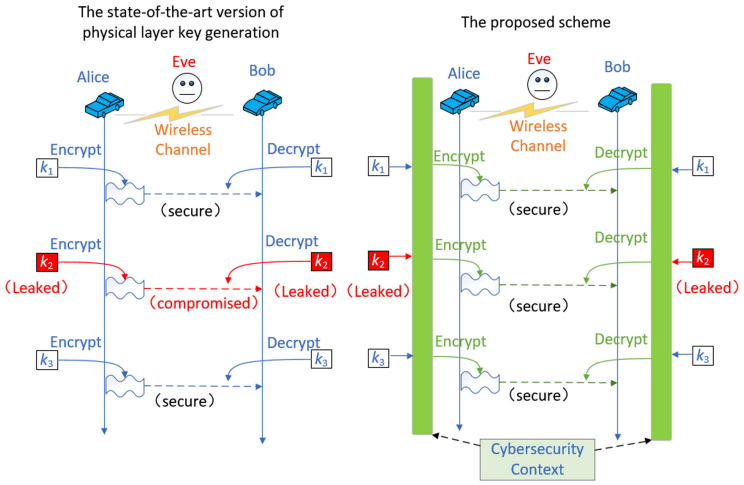


Fig. 2. Comparison between the proposed scheme and the state-of-the-art version of physical layer key generation methods

at the sink can be expressed as a pair of function $<f : \Psi \times \Lambda \rightarrow \Lambda, g : \Psi \times \Lambda \rightarrow \Lambda>$, where the encoder is

$$f(\psi_i, x_i) \rightarrow m_i \tag{1}$$

and decoder is

$$g(\psi_i, m_i) \rightarrow x_i \tag{2}$$

in which x_i and m_i are plaintext and ciphertext, respectively.

We declare that an effective CBCF scheme shall possess the following characteristics.

1. The SA, $\psi_i \in \Psi$, is a slice of common information shared by the source and the sink.
2. There is a pair of efficient encoding and decoding functions, $<f(), g()>$, so that $g(\psi_i, f(h_i, x_i)) = x_i$ holds, for any given $x_i \in \Lambda$.
3. Any third party cannot or has to pay an unacceptable price to decode x_i, even if it has the opportunity to intercept quiet a few or all of the ciphertexts.
4. Any third party cannot or has to pay an unacceptable price to know the content of ψ_i.

Note that we assume that $<f(), g()>$ can be publicly known by a third party, but it should be impossible or extremely arduous to know the contents of SA. Feature 1 defines that the essence of the anchor is information. Feature 2 requires the SA be used for efficiently encoding and decoding. Feature 3 emphasizes that the third party should be unable to decipher the message. Feature 4 requires a third party to be unable to crack the SA to support for Feature 3.

In short, the two parties of the communication maintain a context based on some kind of shared information between them to control the coding process, as

shown in Fig. 2. Due to the constant movement of vehicles and continuous interaction between them, the context is always changing, so it's extremely arduous for the eavesdropper to obtain the complete context, consequently, the current message is prevented from being cracked by introducing information integrity problems.

3 Two Typical Implementations of Context-Bound Cybersecurity Framework

Although all kinds of information, such as moving trajectory, topology, reciprocal channel characteristics and historical communication messages, that are privately shared between the source and the sink can be utilized for constructing the SA, we focus on historical communication messages here for brevity.

Here, we propose two typical implementations of context-bound cybersecurity framework: AES(Advanced Encryption Standard) enhanced Sink-Anchoring Coding (ASAC) and Pseudo-random generator based Sink-Anchoring Coding (PSAC).

3.1 Iterative Function

For x_i, we define its SA ψ_i as the result of an iterative function $\psi : \Psi \times \Lambda \rightarrow \Lambda$ based on some historical messages. Equation (3) and Eq. (4) below represent the iterative formulas of ASAC mode and PSAC mode, respectively:

- ASAC mode:

$$\begin{aligned} \psi_i &= \psi(\psi_{i-1}, x_{i-1}) \\ &= \begin{cases} AES_{key}(\psi_{i-1} \oplus x_{i-1}) & R_{key}(*) \geq \overline{R} \\ \psi_{i-1} & R_{key}(*) < \overline{R} \end{cases} \end{aligned} \tag{3}$$

- PSAC mode:

$$\begin{aligned} \psi_i &= \psi(\psi_{i-1}, x_{i-1}) \\ &= \begin{cases} (\psi_{i-1} + x_{i-1} \oplus (R_{seed}(*) \times 2^z)) \; MOD \; 2^z & R_{seed}(*) \geq \overline{R} \\ \psi_{i-1} & R_{seed}(*) < \overline{R} \end{cases} \end{aligned} \tag{4}$$

where $R_\alpha(*)$ is a pseudo-random sequence generator function, defined as $R_\alpha : \Lambda \rightarrow [0, 1)$, and it is assumed that the generator function implementation can be public, with only the random number seed to be private to the eavesdropper. For the sake of uniformity, we refer to α as the initial parameter of the SA. $\overline{R}$ represents the average sampling rate and can be set to a pseudo-random sequence or constant. As an example, $\overline{R} = 0.1$ indicates that about 90% of the historical messages (historical SAC units) are used in the calculation of ψ_i.

The ASAC mode is implemented based on the one-way function $AES_{key}()$, which is the encoding function of AES. $AES_{key}(x)$ means using AES to encrypt x based on key (Note: only encryption operation is needed, no decryption operation), and $R_{key}(*)$ indicates that the AES key is used as the random number seed

of pseudo-random sequence generator function (if the length does not match, fill or truncate it). In addition, the PSAC mode is based on the pseudorandom sequence generator function, while MOD is a modulo operation to ensure the length of ψ_i is equal to that of SAC unit, and the role of $x_{i-1} \oplus (R_{seed}(*) \times 2^z)$ is to randomly flip x_{i-1} by using $R_{seed}(*)$.

3.2 Communicating Process

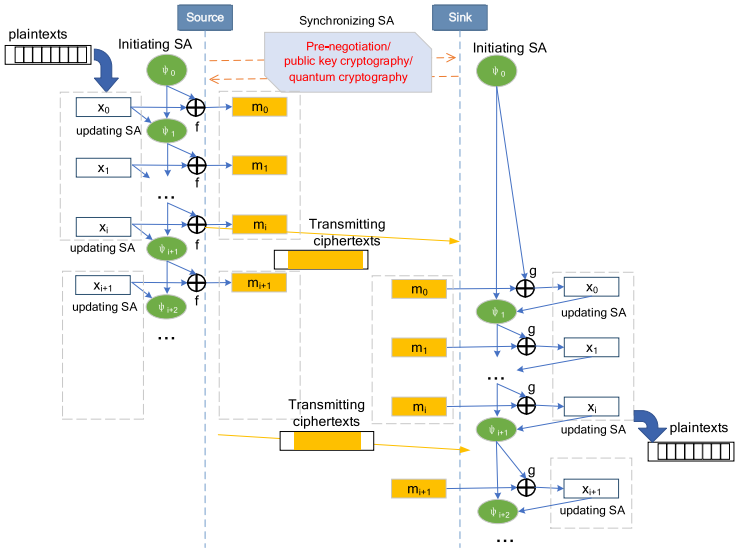


Fig. 3. Sequence diagram of context-bound cybersecurity framework

Figure 3 gives the sequence diagram of both ASAC mode and PSAC mode. At first, a pair of source and sink synchronizes the initial SA and such a synchronization process should be protected by a key exchange mechanism, such as public key cryptography or quantum cryptography, so that the third party cannot know the SA. After starting the communication, the source and sink update the SA according to the same iterative function (i.e., ASAC or PSAC). Since the SA is privately shared between the source and the sink, $f()$ and $g()$ can utilize the SA to construct a key for encryption and decryption of messages, respectively. Specifically, the source encrypt SAC unit one by one with the constantly updated SA through encryption algorithm $f()$, while the sink utilizes the corresponding SA through decryption algorithm $g()$ to decrypt received ciphertexts. For simplicity, we adopt XOR operation for implementing $f()$ and $g()$ () as follows:

$$f(\psi_i, x_i) = \psi_i \oplus x_i \rightarrow m_i \tag{5}$$

$$g(\psi_i, m_i) = \psi_i \oplus m_i \rightarrow x_i \tag{6}$$

where $\oplus$ is a XOR operator, satisfying $0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$, i.e., same as 0, different as 1.

A simple way to understand cybersecurity is to think of the key cracking process as solving equations. For a third party who is assumed to have eavesdropped k ciphertext units, $<m_{i-k}, m_{m-k+1}, ..., m_{i-1}>$, a brute force deciphering is equivalent to solving the following system of equations.

$$\begin{cases} \psi_{i-k+1} \oplus x_{i-k+1} = m_{i-k+1} \\ \psi(\psi_{i-k+1}, x_{i-k+1}) \oplus x_{i-k+2} = m_{i-k+2} \\ \vdots \\ \psi(\psi...(\psi_{i-k+1}, x_{i-k+1})...), x_{i-1}) \oplus x_i = m_i \end{cases} \tag{7}$$

Note that the equation system (7) consists of k equations, but involves a total of $k+1$ unknown variables, i.e. $\{\psi_{i-k+1}, x_{i-k+1}, x_{i-k+2}, ..., x_i\}$, so it cannot be solved directly in general, unless there is a special correlation among these equations that can be explored. When k is equal to $i+1$, it is corresponding to the case that the third party has eavesdropped all historical ciphertexts. Under the assumption that the initial SA is unknown, it is also impossible to directly solve the equation system, as the number of unknown variables is still more than the number of equations. For more rigorous theoretical analysis on the cybersecurity, please refer to Sect. 3.4.

3.3 The Necessity of Pseudo-random Generator Function

If a true random number generator is used, the same seed cannot lead to the same sequence, then the source and the sink will not be able to construct feasible encoding and decoding functions, which violates the feature 2 of CBCF scheme.

From Eq. (3) and Eq. (4), it can be seen that the role of pseudo-random generator function in PSAC mode is mainly in three aspects: initializing SA, selecting historical messages, and randomly flipping x_{i-1} by bit. In ASAC mode, there are only two effects of pseudo-random function: initializing SA and selecting historical messages. Using a pseudo-random generator function $R_\alpha(*)$ to select historical messages brings the following advantages:

1. From the perspective of staying security from eavesdroppers. First of all, when the initial parameter α of $R_\alpha(*)$ is unknown to the third party, all historical messages have to be collected since it is impossible to know what historical messages are used for constructing ψ_i, which may trigger difficulties in collection or storage. Secondly, even if the eavesdropper collects all the historical messages, it will not be able to crack the current ciphertext m_i without α, because it cannot know which messages are used to construct ψ_i. Finally, if the eavesdropper does not collect all the messages from the beginning, it will fail to crack the ciphertext because it cannot master all the messages satisfying $R_\alpha(*) \geq \overline{R}$, even if the eavesdropper learns α by cracking $R_\alpha(*)$ after the communication has started for a period of time.

2. From the perspective of the performance of encryption and decryption at the source and the sink. The source and the sink invariably generate the same pseudo-random number sequence, and only need to update the SA for messages satisfying $R_\alpha(*) \geq \overline{R}$. When $\overline{R}$ approaches 0, the sink anchor will be updated for almost every message, thus achieving a similar "one-time pad" effect [13]. When $\overline{R} > 0$, the calculation frequency will be reduced and thus the computing resources are saved.

3.4 Analysis of Cybersecurity

In order to simplify the analysis, we only consider the situation of passive eavesdropping, i.e., eavesdroppers will not interfere with the transmission of messages, so that no message-authentication code (MAC) is needed.

Min-entropy is widely used in modern cryptography for evaluating the guessing probability of a key in the worst case, so it provides another perspective for understanding the cybersecrity. For a random variable X, its min-entropy of is defined as [1]

$$H_\infty(X) \stackrel{def}{=} -\log_2(\max_{x \leftarrow X} Pr[X = x]) \tag{8}$$

where $x \leftarrow X$ means the operation of sampling a random x according to X. Given two independent random variables, X and Y, it can be proven that

$$H_\infty(X \oplus Y) \geq max(H_\infty(X), H_\infty(Y)) \tag{9}$$

and

$$H_\infty((X + Y)\ MOD\ 2^z) \geq \\ max(H_\infty(X\ MOD\ 2^z), H_\infty(Y\ MOD\ 2^z)) \tag{10}$$

where $max()$ is a function that returns the larger value from the two inputs. For simplicity, we assume that the one way function $AES_{key}()$ is entropy-preserving in an ideal case, i.e., $H_\infty(AES_{key}(X)) \equiv H_\infty(X)$. Then, for ASAC mode, we have,

$$H_\infty(\psi_i) \geq max(H_\infty(\psi_{i-1}), H_\infty(x_{i-1}), H_\infty(key)) \tag{11}$$

Thus, during the update process of the SA, the min-entropy is always increasing until its value reaches the absolute maximum limit for z-bit keys. That is to say, the update process of the SA makes it more difficult for eavesdroppers to guess the SA. Therefore, by taking the SA as keys for encryption and decryption, the cybersecurity is indeed enhanced by the ASAC mode.

Likewise, we can draw the same conclusion for the PSAC mode.

3.5 Theoretical Analysis of Performance

In the PSAC mode, each update process of SA includes at most one addition, XOR, multiplication, and modulo operation, so the computational time complexity is O(1). In the ASAC mode, each update process of SA only adds an

XOR operation to the standard AES encryption operation, therefore its computational time complexity can still be considered as O(1). In terms of space complexity, both the source and sink need to maintain a record of SA, and determine whether to update it every time a message is sent or received. The input of computation process is ψ_{i-1} and x_{i-1} while the output is ψ_i. Since there is no need to save earlier historical messages, the storage space overhead is O(1). After encoding, the ciphertext and the plaintext are of equal length, so the additional communication overhead of the protocol is O(0). In contrast, for the eavesdropper, the theoretical computational time complexity of cracking plaintext from the ciphertext is infinite if the initial parameter α is unknown, because the equation system (7) cannot be solved directly unless all historical messages are intercepted and stored.

4 Experiments and Analyses

In order to evaluate the proposed framework, we implemented the scenario, as shown in Fig. 1, and then analyzed the statistical characteristics of plaintext and ciphertext.

4.1 Ciphering and Deciphering in Vehicle Networks

We simulated a vehicle network based on multicast and TCP connections for internal interaction, where each vehicle establishes communication with other vehicles centered on itself. Here, we focus on the communication process of one vehicle V_s in a vehicle network consisting of n vehicles, and we use $V_i (i = 1, ..., s-1, s+1, ..., n)$ to represent the remaining vehicles, as shown in Fig. 4.

The vehicles firstly synchronize the initial SA parameter α by using public key encryption, then establish context-bound secure channels with initialized SA. The whole process can be divided into four steps:

- Step A: V_s generates a pair of RSA keys locally and then sends the public key to the remaining vehicles as a multicast source. After receiving the public key, vehicle V_i will establish a TCP channel with V_s.
- Step B: V_i generates an initial SA parameter α locally, which will be encrypted with the received RSA public key and then synchronized to V_s latter. After V_s decrypting the encrypted parameter α with local RSA private key, both parties will initialize SA according to α.
- Step C: After SA is initialized, V_i will encrypt messages with SA and send them to V_s through the TCP connection, while V_s decrypts using the synchronized SA.
- Step D: Each message transmission will lead to the update of SA according to the iteration function, and both parties will iteratively update SA with the same algorithm (ASAC is used as an example here) in the subsequent communication process.

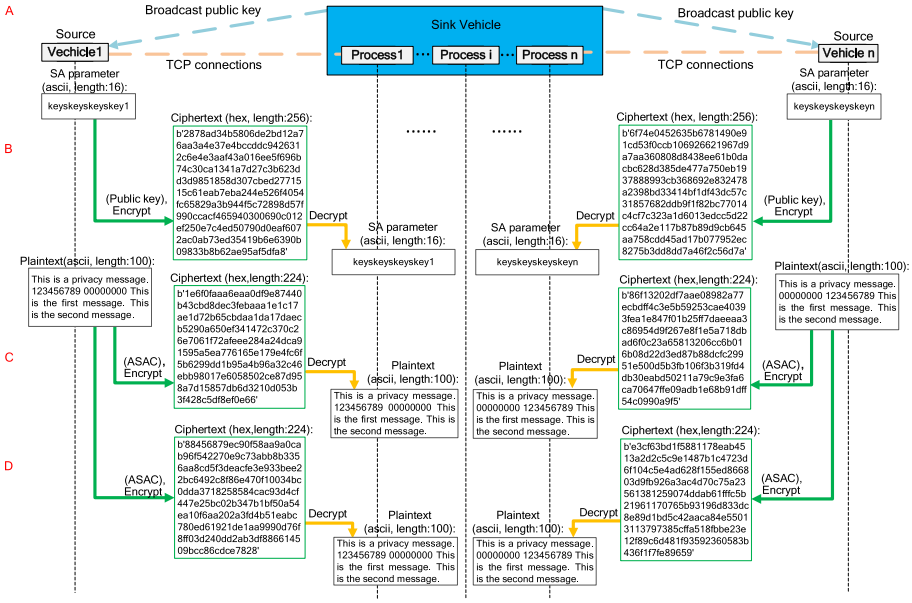


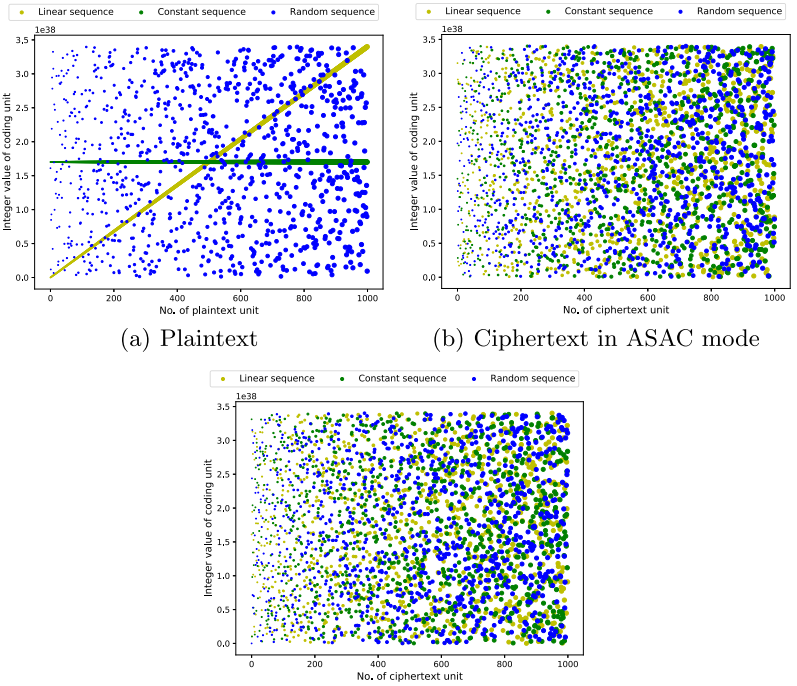
Fig. 4. An example of ciphering and deciphering process (z = 128)

For the iteration and communication with different vehicles, V_s will handle them in a multi-threaded manner. Besides, as it can be seen in Fig. 4, even if the same message is sent continuously, the ciphertext obtained is completely different due to the update of SA, which is benificial for the security of user data.

4.2 Distribution and Randomness of Ciphertext

In order to study the statistical characteristics of the encrypted messages under the proposed cybersecurity framework, we use linear sequence, constant sequence and random sequence as plaintext, which are encrypted in ASAC and PSAC modes, respectively, so as to explore the distribution of the ciphertext. For comparison, we uniformly set the random number seed to 1, the binary length of the coding unit to 128, the average sampling rate $\overline{R}$ to 0.1, and each sequence contains 1000 coding units.

The results are shown in Fig. 5, where Fig. 5(a) represents the distribution of plaintext, Fig. 5(b) represents the distribution of ciphertext in ASAC mode, and Fig. 5(c) represents the distribution of ciphertext in PSAC mode. It can be seen that regardless of the type and serial number of the plaintext, the corresponding ciphertext is evenly distributed in the range of $[0, 2^z - 1]$ whether in ASAC or PSAC mode, reflecting outstanding statistical distribution characteristics. Even at the worst case, assuming that the third party has the opportunity to know the initial parameter α of SA, the only case it can crack the ciphertext is that



(a) Plaintext (b) Ciphertext in ASAC mode

(c) Ciphertext in PSAC mode

Fig. 5. Distribution of plaintext and ciphertext

it collects and stores all the ciphertexts before getting α, which is usually not possible for third parties with relatively limited capabilities.

Further, the randomness of ciphertext is also tested. Since the ciphertext is used to ensure the security of information, and any non-random features will reduce the difficulty for third parties to crack the message, the randomness of ciphertext is crucial [22]. For testing, we use the linear sequence generated above as plaintext (1000 coding units), then encrypt it in ASAC mode and PSAC mode, respectively. For the obtained ciphertext, we use a few different algorithms from the NIST public-domain test suite to calculate the $p-value$ of it [23,31], which is an index used to evaluate the randomness of bit sequence, and the sequence is considered to be random when $p-value > 0.01$ (For a detailed representation of the tests and definitions of $p-value$, interested readers can refer to [31]). The results are shown in Table 1, in which we can see that the ciphertext we obtained has a good performance in terms of randomness, which further ensures the effectiveness of our framework.

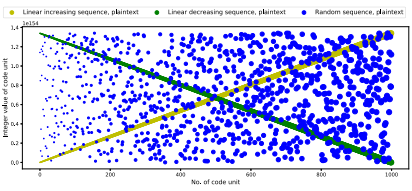
Table 1. Results from randomness tests on ciphertext

Test	P-value (ASAC)	P-value (PSAC)
DFT test	0.7953	0.3316
Longest run	0.2724	0.7328
Maurer's test	0.3722	0.9793
Non-overlapping	1.0000	1.0000

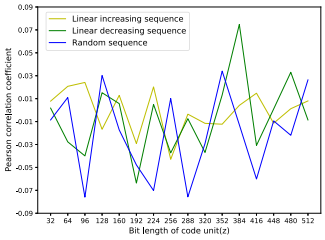
4.3 Correlation Between Plaintext and Ciphertext

To study the correlation between plaintext and ciphertext, we use random sequence, linearly increasing sequence and linearly decreasing sequence as plaintext respectively to get ciphertext, and then calculate the corresponding Pearson correlation coefficient [5]. The results are shown in Fig. 6.

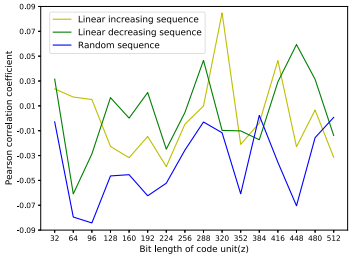
We use the same seed 1 for pseudo-random generator to produce randomly distributed plaintext sequences, the linear increasing sequence and the linear decreasing sequence. These linear sequences take values uniformly on $[0, 2^z - 1]$, and the range of the coding unit length z is varying in $[32, 512]$, incremented by 32 each time. Figure 6(a) demonstrates the plaintext distribution of three sequences when $z = 512$, where each sequence contains 1000 coding units. Figure 6(b) demonstrates the correlation between plaintext and ciphertext under different coding unit lengths in ASAC mode, and Fig. 6(c) demonstrates the correlation



(a) Distribution of plaintext



(b) Correlation in ASAC mode



(c) Correlation in PSAC mode

Fig. 6. Correlation between plaintext and ciphertext

in PSAC mode. The Pearson correlation coefficient in Fig. 6(b) and Fig. 6(c) represent the linear correlation between plaintext and ciphertext, where 1 is indicative of a positive correlation and -1 is indicative of a negative correlation. It can be seen that the correlation coefficients between plaintext and ciphertext of the three different sequences are all within the range of $[-0.09, +0.09]$, which manifests that the encryption result is not sensitive to different values of z, and reflects outstanding statistical correlation characteristics of the proposed framework.

5 Discussions

5.1 Resistance to Attacks

From the perspective of attack games common to cryptographic systems, the quintessential types of attack fall into four categories, including ciphertext-only attacks, known ciphertext attacks, chosen plaintext attacks, and chosen ciphertext attacks [6]. These attacks assume that the ciphertext relies only on the key, consequently, the focus is on attacking the key. However, the encoding and decoding process of our context-bound cybersecurity framework rely on the dynamic context, which means it naturally has the advantage of resisting such attacks. Furthermore, the proposed framework can resist third-party's retrospective attacks on communication fragments. In a retroactive attack, if the third-party cannot crack the message in a timely manner, the interested communication segment can be stored and decrypted when the conditions are met. As an example, the eavesdropper can wait for the future quantum computer to decrypt messages that was captured 10 years ago. Encryption methods such as DES, AES, and RSA all face this type of risk. In contrast, our framework makes the decryption process not only require the initial parameter of SA, but also historical messages, which will force the third-party to have the ability to continuously monitor the channel, collect data and store complete historical messages in addition to powerful computing capabilities, so the cost is extremely great.

5.2 Extensions

As mentioned above, the method of constructing context is not limited to historical messages. As an example, some reciprocal channel characteristics in the vehicle network also provide context information shared between the source and sink, thus can be utilized for our framework. Currently, the fifth generation (5G) telecommunications techniques [34], such as massive multiple-input-multiple-output (MIMO), millimeter wave (mmWave), and non-orthogonal multiple access (NOMA), have been widely explored for physical-layer key generation, but each key is used only once and discarded, resulting in a waste of a large number of precious keys. By weaving these keys into the context, our scheme provides an elegant way to resist key exposure for these methods. One approach

is replacing x_{i-1} in Eq. (3) and Eq. (4) with reciprocal channel parameters, then the framework no longer depends on historical messages, but on the channel parameters.

In addition, in Sect. 2.2, we assumed that the channel is error-free, reliable, and not out of order, but the real networking environment may be unreliable or does not need to be absolutely reliable. For instance, audio and video transmission have certain tolerance to channel errors.

To this end, other variations of ASAC and PSAC can be considered, as shown in Fig. 7. We can use Automatic Repeat-reQuest (ARQ) to get an error-free, reliable, and not out of order channel. If ARQ is not available, by replacing x_{i-1} in Eq. (3) and Eq. (4) with fixed values, two weakened modes are obtained: Weak-ASAC mode and Weak-PSAC mode. These weakened modes can be applied to unreliable transmission environment, though some protocol enhancements, e.g., embedding message-authentication codes (MAC), may be needed to handle packet loss and out of order. Therefore, combining with various modes, it is possible to apply our framework to all protocol layers of the vehicle network, including physical layer, link layer, network (IP) layer, transport layer, application layer, etc.

5.3 Relationship with Blockchain

A blockchain is typically an ordered and growing list of blocks that are linked using cryptography. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data. Essentially, it is a decentralized distributed ledger database where each member independently stores a copy of the blockchain and updates it synchronously [26]. Our framework uses a similar idea of chain, but the context is only shared between legitimate communication peers, i.e., the source and the sink. We take messages and SA as the content of block and carefully designed iterative function as the chain-relationship between blocks to establish a context that is stored in a distributed manner and updated synchronously between the two parties in communication. In natrue, we constructed a dynamic key that changes with the context for message encryption and decryption, so that the risk of key exposure can be overcomed.

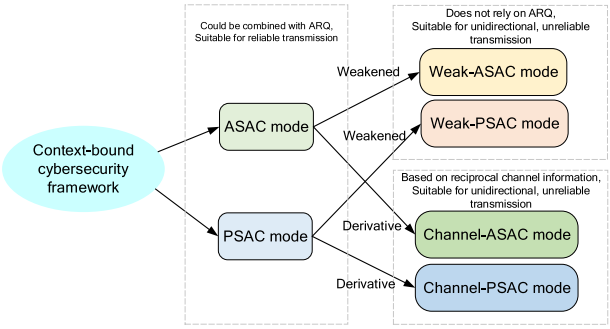


Fig. 7. Six typical modes based on context-bound cybersecurity framework

6 Conclusion

In this paper, we propose a context-bound cybersecurity framework to solve the data security problem in vehicle networking scenarios. By binding the message encryption to the context information in the communication process, it is impossible or extremely hard for third parties to crack the message from the channel or relay. On the basis of the proposed framework, this paper offers two typical implementations in combination with AES and pseudo-random generator: ASAC and PSAC. By experimenting on communication process in a vehicle network consisting of several cars, the simulation results demonstrate that the proposed framework has outstanding statistical correlation and distribution characteristics. The context-bound cybersecurity framework greatly reduces the risk of key exposure by utilizing the mobility of vehicles, thus offers a solution to simplify the technology implementation of mobile user privacy protection and data security sharing. Our approach also provides a promising way to resist the upcoming quantum computers, because it will become more and more difficult for third parties to collect the complete context as the context continues to update.

Besides, we only considered the situation of passive eavesdropping and assumed that the channel is error-free, reliable, and not out of order. The real networking environment may be unreliable or does not need to be absolutely reliable. Moreover, the eavesdropper may actively interfere with the transmission process of the communication by modifying, deleting, inserting, or replaying the message transmission. Thus, some protocol enhancements, e.g., embedding message-authentication codes (MAC), may be needed. These issues need further efforts invested.

References

1. Aggarwal, D., Dodis, Y., Jafargholi, Z., Miles, E., Reyzin, L.: Amplifying privacy in privacy amplification. In: Garay, J.A., Gennaro, R. (eds.) CRYPTO 2014. LNCS, vol. 8617, pp. 183–198. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44381-1_11
2. Al-Sultan, S., Al-Doori, M.M., Al-Bayatti, A.H., Zedan, H.: A comprehensive survey on vehicular ad hoc network. J. Netw. Comput. Appl. **37**, 380–392 (2014)
3. Alaba, F.A., Othman, M., Hashem, I.A.T., Alotaibi, F.: Internet of things security: a survey. J. Netw. Comput. Appl. **88**, 10–28 (2017). https://doi.org/10.1016/j.jnca.2017.04.002
4. Arute, F., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**(7779), 505–510 (2019)
5. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Cohen, I., Huang, Y., Chen, J., Benesty, J. (eds.) Noise Reduction in Speech Processing, pp. 1–4. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00296-0_5
6. Biryukov, A., Wagner, D.: Advanced slide attacks. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 589–606. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_41

7. Brumley, D., Boneh, D.: Remote timing attacks are practical. Comput. Netw. **48**(5), 701–716 (2005). https://doi.org/10.1016/j.comnet.2005.01.010
8. Cao, J., Yu, P., Xiang, X., Ma, M., Li, H.: Anti-quantum fast authentication and data transmission scheme for massive devices in 5G NB-IOT system. IEEE Internet Things J. **6**(6), 9794–9805 (2019)
9. Cao, J., et al.: A survey on security aspects for 3GPP 5G networks. IEEE Commun. Surv. Tutor. **22**(1), 170–195 (2019)
10. Chaudhary, R., Aujla, G.S., Kumar, N., Zeadally, S.: Lattice-based public key cryptosystem for internet of things environment: challenges and solutions. IEEE Internet Things J. **6**(3), 4897–4909 (2019). https://doi.org/10.1109/JIOT.2018.2878707
11. Conti, M., Dragoni, N., Lesyk, V.: A survey of man in the middle attacks. IEEE Commun. Surv. Tutor. **18**(3), 2027–2051 (2016). https://doi.org/10.1109/COMST.2016.2548426
12. Cui, C., et al.: Twin-field quantum key distribution without phase postselection. Phys. Rev. Appl. **11**(3), 034053 (2019)
13. Deng, F.G., Long, G.L.: Secure direct communication with a quantum one-time pad. Phys. Rev. A **69**(5), 052319 (2004)
14. Ebrahimi, S., Bayat-Sarmadi, S., Mosanaei-Boorani, H.: Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT. IEEE Internet Things J. **6**(3), 5500–5507 (2019)
15. Guan, L., Lin, J., Ma, Z., Luo, B., Xia, L., Jing, J.: Copker: a cryptographic engine against cold-boot attacks. IEEE Trans. Dependable Secure Comput. **15**(5), 742–754 (2016)
16. Handler, I.: Data sharing defined - really! IEEE Comput. **51**(2), 36–42 (2018). https://doi.org/10.1109/MC.2018.1451659
17. Huh, S., Cho, S., Kim, S.: Managing IoT devices using blockchain platform. In: 2017 19th International Conference on Advanced Communication Technology (ICACT), pp. 464–467. IEEE (2017)
18. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48405-1_25
19. Kocher, P.C.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 104–113. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68697-5_9
20. Liu, Y., et al.: Experimental twin-field quantum key distribution through sending or not sending. Phys. Rev. Lett. **123**(10), 100505 (2019)
21. Luzzi, L., Vehkalahti, R., Ling, C.: Almost universal codes for mimo wiretap channels. IEEE Trans. Inf. Theory **64**(11), 7218–7241 (2018). https://doi.org/10.1109/TIT.2018.2857487
22. Mathur, S., Trappe, W., Mandayam, N., Ye, C., Reznik, A.: Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In: Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, pp. 128–139 (2008)
23. Maurer, U.M.: A universal statistical test for random bit generators. J. Cryptol. **5**(2), 89–105 (1992). https://doi.org/10.1007/BF00193563
24. Meng, W., Li, W., Yang, L.T., Li, P.: Enhancing challenge-based collaborative intrusion detection networks against insider attacks using blockchain. Int. J. Inf. Secur. **19**(3), 279–290 (2019). https://doi.org/10.1007/s10207-019-00462-x

25. Mollah, M.B., Azad, M.A.K., Vasilakos, A.V.: Secure data sharing and searching at the edge of cloud-assisted internet of things. IEEE Cloud Comput. **4**(1), 34–42 (2017). https://doi.org/10.1109/MCC.2017.9
26. Ortega, V., Bouchmal, F., Monserrat, J.F.: Trusted 5G vehicular networks: blockchains and content-centric networking. IEEE Veh. Technol. Mag. **13**(2), 121–127 (2018)
27. Pirandola, S., et al.: High-rate measurement-device-independent quantum cryptography. Nat. Photonics **9**(6), 397–402 (2015)
28. Preneel, B.: Cryptography and information security in the post-snowden era. In: TELERISE@ ICSE, p. 1 (2015)
29. Renauld, M., Standaert, F.-X.: Algebraic side-channel attacks. In: Bao, F., Yung, M., Lin, D., Jing, J. (eds.) Inscrypt 2009. LNCS, vol. 6151, pp. 393–410. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16342-5_29
30. Riyadi, M.A., Khafid, M.R.A., Pandapotan, N., Prakoso, T.: A secure voice channel using chaotic cryptography algorithm. In: Proceedings of International Conference on Electrical Engineering and Computer Science (ICECOS), pp. 141–146 (2018)
31. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Technical report, Booz-allen and hamilton inc mclean va (2001)
32. Singh, J., Pasquier, T.F.J., Bacon, J., Ko, H., Eyers, D.M.: Twenty security considerations for cloud-supported internet of things. IEEE Internet Things J. **3**(3), 269–284 (2016). https://doi.org/10.1109/JIOT.2015.2460333
33. Wang, Z., Han, Y., Liu, W., Chen, L.: Anti-quantum generalized signcryption scheme based on multivariate and coding. In: Proceedings of Chinese Control and Decision Conference (CCDC), pp. 3587–3594 (2019)
34. Wu, Y., Khisti, A., Xiao, C., Caire, G., Wong, K.K., Gao, X.: A survey of physical layer security techniques for 5G wireless networks and challenges ahead. IEEE J. Sel. Areas Commun. **36**(4), 679–695 (2018)
35. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutor. **15**(4), 2046–2069 (2013). https://doi.org/10.1109/SURV.2013.031413.00127
36. Zhang, J., Zheng, K., Zhang, D., Yan, B.: AATMS: An anti-attack trust management scheme in VANET. IEEE Access **8**, 21077–21090 (2020)