BAGKD: A Batch Authentication and Group Key Distribution Protocol for VANETs

Guangquan Xu, Xiaotong Li, Litao Jiao, Weizhe Wang, Ao Liu, Chunhua Su, Xi Zheng, Shaoying Liu, and Xiaochun Cheng

Abstract

As an important application of mobile ad hoc networks, VANETs play an important role in intelligent transportation. However, with the development of mobile communication technology, as well as the needs of intelligent transportation, both security and efficiency are required for real-time authentication and communication. Traditional authentication and key distribution schemes suffer from network failure and high response latency. To solve the above problems, this article proposes a BAGKD protocol to achieve robust and efficient networking for the security and efficiency of VANETs. In our protocol, bilinear mapping is used to realize batch authentication, which can improve authentication efficiency and reduce message errors caused by factors such as high speed of vehicles. The group key distribution mechanism can update the group key dynamically, which reduces the risk of group key leakage effectively. For the sake of privacy protection, vehicles utilize pseudonyms issued by a trusted authority to communicate with RSUs. The security of BAGKD protocol is verified by simulation in our experiments on AVISPA. In addition, when compared to three existing protocols based on bilinear mapping, our proposed BAGKD outperforms them in terms of efficiency and communication overhead while maintaining security. The simulation results further confirm that BAGKD is suitable for short-range communication scenarios such as VANETs.

Introduction

Nowadays, more and more countries have clearly defined the goal of building smart cities. In the construction of smart cities, intelligent transportation is a vital part. It is designed to foresee road congestion, and prevent traffic accidents and any other traffic problems. As one of the main ways to achieve intelligent transportation, vehicular ad hoc networks (VANETs) have been adopted in practice.

Through VANETs, various traffic information can be shared between vehicles and traffic information management centers, vehicles and roadside units (RSUs), and vehicle-to-vehicle. Therefore, road conditions and traffic hazards can be predicted in advance, thereby greatly reducing traffic accidents and making timely emergency responses. However, the dramatic increase of vehicles has raised more and more strict demands on the communication efficiency of VANETs, and the messages in VANETs may be easily intercepted, modified, and/or replayed in an unauthorized way due to the fact that VANETs adopt wireless communication technology [1]. To solve this problem, we propose a batch authentication and group key distribution (BAGKD) protocol for VANETs. It provides efficient authentication and real-time communications for all of the vehicles in VANETs. Simultaneously, it utilizes group key distribution to ensure the secure communication of these vehicles in VANETs.

Our BAGKD protocol essentially consists of the following two components.

Batch Authentication of Messages: Because the quantity of vehicles is large and the vehicles move at high speeds, the topology of VANETs also changes frequently. The traditional one-by-one authentication is inefficient and does not meet the requirements of the VANET architecture. Therefore, more and more batch authentication protocols have been proposed. There are three mainstream authentication protocols [2–4] utilizing bilinear pairing to realize batch authentication. Due to the calculation of bilinear mapping being complex, however, the improvement of efficiency in batch authentication is still challenging. To address this challenge, we design another batch authentication method based on bilinear mapping. Compared to the existing authentication protocols, our protocol has superior performance in terms of computation cost and resource consumption.

Group Key Distribution Mechanism: The sharing of a group key by all group members is a precondition for secure communication in a group, but the dynamic change of group members increases the risk of a group key's leakage. Therefore, when a member joins or quits, the group needs to update the group key in time. We adopt distributed group key management, in which all the group members participate in the establishment of the group key. It eliminates the dependence on the safety of the group key manager. Once the manager of group keys is not safe anymore, the vehicles can still communicate securely. In addition, the distributed group key management can adapt well to a frequently changing dynamic group such as a VANET.

As an important application of mobile ad hoc networks, VANETs play an important role in intelligent transportation. However, with the development of mobile communication technology, as well as the needs of intelligent transportation, both security and efficiency are required for real-time authentication and communication.

Guangquan Xu is with Qingdao Huanghai University and Tianjin University; Xiaotong Li and Weizhe Wang are with Tianjin University; Litao Jiao is with Qingdao Huanghai University; Ao Liu (corresponding author) is with Tianjin University of Technology; Chunhua Su is with the University of Aizu and Peng Cheng Laboratory; Xi Zheng is with Macquarie University; Shaoying Liu is with Hiroshima University; Xiaochun Cheng (corresponding author) is with Middlesex University.

Digital Object Identifier: 10.1109/MCOM.001.2000118

0163-6804/20/$25.00 © 2020 IEEE

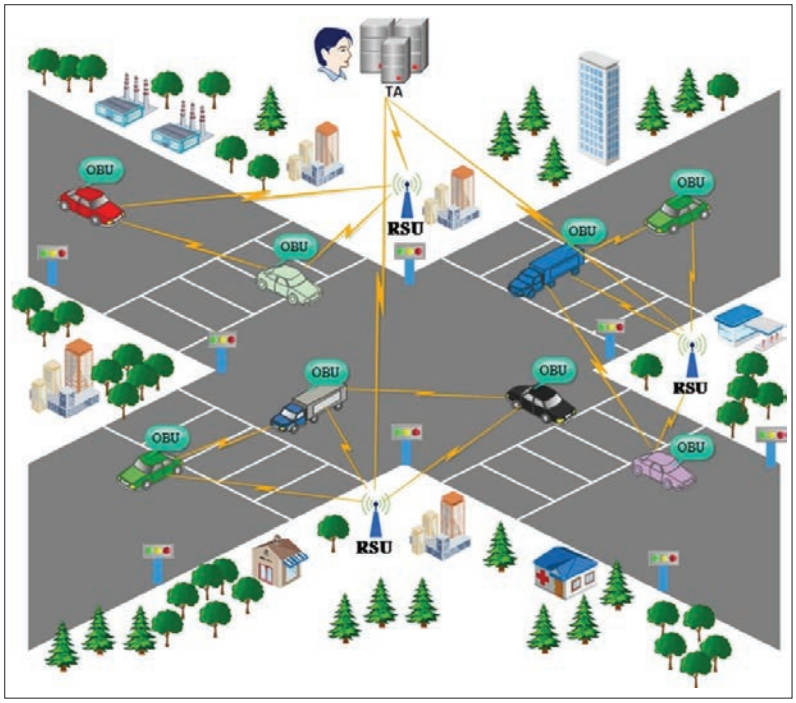

Figure 1. The network model of VANETs.

We combine the two components to form our BAGKD protocol, which ensures that each authenticated vehicle can take part in the establishment of the group key and get the same group key in a timely fashion. In this way, it breaks from the dependence on the group key manager. The group key can still generate and distribute normally even if there is no trusted third party. Meanwhile, it can respond to the group key requirements from all the legal vehicles by one-time establishment and synchronous distribution, which meets the demands of big data networking. In addition, we also utilize anonymous communication in our protocol for privacy.

The VANET Scenario

In this section, we mainly investigate the application scenario: VANETs, including the network model and security requirements.

Network Model

As an important component of intelligent transportation, VANETs are already applied in the traffic of traditional mobile ad hoc networks (MANETs). There are two types of communications in VANETs, vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). Both V2V and V2I normally follow dedicated short-range communications (DSRC) protocol, which provides the ability to support secure V2V and V2I communications [5]. According to DSRC in the wireless communication environment, vehicles broadcast messages about traffic conditions (e.g., weather, road congestion), as well as the vehicle state information (e.g., location, direction, speed, driving status) every 100–300 ms. The network structure model of VANETs is discussed in [6] and illustrated in Fig. 1. There are three elements: *trusted authority* (TA), *RSU*, and *onboard unit* (OBU). Below, we introduce the key elements in the structure model one by one.

TA: The TA is the central agency with the greatest computing and communication capabilities. It is responsible for verifying all vehicles and roadside units. For authentication, the vehicles and roadside units register their unique certificates from the TA over the secure channel. Our protocol assumes that the TA is completely trusted and cannot be compromised in any way.

RSU: The RSU is usually a fixed device that is installed at roads or dedicated locations, such as a parking lot or road intersection. The RSU is equipped with transceivers, antennas, processors, and sensors, as well as storage capabilities for storing the information of vehicles' OBUs and the TA. Each RSU uses an IEEE 802.11p-based DSRC radio access radio channel, a directional antenna, and an omnidirectional antenna.

OBU: Each vehicle is equipped with an OBU, which is used to exchange information with the RSU, other OBUs, and other computation devices from other vehicles. The OBU is constituted by a resource command processor, a read/write memory, a user interface, and an IEEE 802.11p-based DSRC radio to access the wireless channel.

Security Requirements

Both security and privacy are significant requirements of VANETs. The security requirements of VANETs are usually divided into four categories, including *confidentiality*, *integrity*, *availability*, and *authentication* [7].

Confidentiality: It is to ensure that information is only obtained by authorized users. The attacker could destroy confidentiality through traffic analysis [8]. An eavesdropping attack is also a threat to confidentiality. During V2V or V2I communication, a man-in-the-middle attack could be set to listen and modify messages, but it will not be known by the communicators.

Integrity: It is one of the key aspects of information security. It focuses on the accuracy and reliability of information throughout the transmission process. In the process of information transmission, both the replay attack and message modification attack may threaten data integrity [9]. Apart from these malicious activities, an attacker can steal passwords to enter the VANET's system as a legitimate user by a masquerading attack.

Availability: The availability of information is an essential requirement for ensuring the efficiency of VANETs. The availability may be destroyed by a denial of service (DOS) attack, which will prevent primary communication between vehicles [10]. A jamming attack is used to damage the radio communication channel of base stations in a VANET system [11]. Moreover, an attacker could also install malware to infiltrate a VANET system.

Authentication: Authentication is a mechanism to protect VANETs from malicious attackers and can be regarded as the first line of defense against attackers. The authentication process in VANETs protects the legitimate nodes from internal or external attackers. Sybil attack and impersonation attack have commonly been adopted to disrupt authentication. The former indicates that an entity acts as several identities to send messages. The latter means that the attacker pretends to be a legitimate user by guessing the true ID of this user.

The BAGKD Protocol

Our BAGKD protocol includes two components: the batch authentication and the group key distribution. The batch authentication is divided into four stages, and the group key distribution consists of three parts.

Batch Authentication

In this subsection, we introduce the batch authentication in the BAGKD protocol. It includes four stages: *system initialization*, *vehicle registration*, *RSU registration*, and *batch certification*.

System Initialization: At this stage, the TA involved initializes all system parameters according to the following process:

- The TA selects two large prime numbers and determines an elliptic curve defined on the finite field.
- The TA selects an additive cyclic group that contains all the points and the infinity point, and one of the generators of the group. Then it gets a bilinear pairing.
- The TA selects two random numbers in the additive cyclic group and determines a random number as the private key. After getting the private key, the TA calculates the corresponding public key with the private key and the generator.
- The TA first selects two hash functions and then makes the system parameters public.

Vehicle Registration: Annual vehicle inspection is the basic system of transportation management. We assume that the vehicle can get enough pseudonyms from the TA during the annual inspection and store it for daily communication. A pseudonym is a hash value generated by the TA using the real information of the vehicle, which is beneficial to protect user information and achieve anonymous authentication. Then the pseudonym and private key of a vehicle in a certain time period can be generated according to the following rules:

- The vehicle selects a unique identity and a corresponding password, and sends them to the TA through a secure channel.
- The TA generates a random number and calculates the pseudonym by hashing with function. Finally, a private key is generated.
- The TA generates another random number, then uses random number, generator, pseudonym, and the private key of the TA to get the private key of the vehicle at the period by hash and connect calculation.
- The TA embeds the pseudonym and private key in the OBU of a vehicle.

RSU Registration: For the RSU in the area, the TA distributes the certificate according to the following rules:

- The TA generates a random number as the private key of the RSU and calculates the public key of the RSU with the private key and the generator.
- After getting the public key of the RSU, the TA uses its own private key to generate the signature of the public key and area information of the RSU.
- Finally, the TA sends the certificate to the RSU securely. The certificate consists of the public key of the RSU, area information, and signature.

Batch Authentication: After the above process is completed, the vehicle and the RSU can mutually authenticate through the pre-stored parameters and the common parameters. For the sake of understanding, we start the description from the single authentication. The main steps are as follows:

- The RSU in an area broadcasts its own certificate periodically (e.g., every 5 s).
- When the vehicle receives the certificate and recognizes that the area is a new area, the vehicle verifies the signature of the RSU by the public key of the TA. During the period, if the certificate is valid, the vehicle generates a random number, and then computes the product of the random number and the generator. The shared key between the vehicle and the RSU is obtained by hashing with the hash function. The shared key is used for the RSU to communicate with the vehicle, such as encrypting the group key.
- If the vehicle needs to send the message with a timestamp, the vehicle ought to generate a random number for the computation of the group key later. Finally, the vehicle sends the authentication parameters to the RSU.
- After receiving the message sent by the vehicle, the RSU can obtain the current time period according to the parameters in the message and compute the verify parameter. Then it verifies the identity of the vehicle, and if the verification passes, the vehicle is successfully authenticated. Finally, the RSU computes the shared key with hash function.
- Actually, our protocol adopts batch authentication based on the above single authentication. When n vehicles enter a certain area at the same time, they will receive the certificate broadcast by the RSU simultaneously and check it. If the verification is successful, each vehicle will send a parameter message to the RSU. The RSU verifies the message in a batch way. If vehicles are verified successfully, the RSU will compute the shared key with these vehicles using the same method as single authentication individually.
- Finally, the RSU calculates the group key and distributes it to vehicles for safe communication later.

Group Key Distribution

In order to ensure secure communication of V2V and V2I, each vehicle must receive the same group key. When a vehicle leaves or enters an area, the group key of the area needs to be updated. It must be ensured that the new member of the group cannot calculate the previous group key, and the leaving vehicle cannot calculate the subsequent group key. In this subsection, we introduce the generation and update of group keys.

Group Key Establishment: Our protocol adopts the improved distributed group key management in which all group members participate in the generation of group keys. The specific process is as follows:

- After batch authentication, the RSU receives authentication messages from the vehicles in the area and calculate the shared key.

In order to ensure secure communication of V2V and V2I, each vehicle must receive the same group key. When a vehicle leaves or enters an area, the group key of the area needs to be updated. It must be ensured that the new member of the group cannot calculate the previous group key, and the leaving vehicle cannot calculate the subsequent group key.

The RSU generates a random number and utilizes it to calculate the group key of the remaining vehicles. It encrypts the group key with the shared key for transmission. Calculating the verify parameter follows the same method as mentioned above. The RSU sends a message to each vehicle.

- The RSU generates random numbers. Then the TA calculates the group key and the remaining product via the random number. It next encrypts the group key with the shared key to get a transmission key for transmission. In this process, verify parameter is calculated by hashing pseudonym, transmission key, and shared key with the function. Finally, the RSU sends the message including pseudonym, transmission key, and verify parameter to each vehicle.
- The vehicle calculates the verify parameter in the same way as calculating the verify parameter according to the message received, and compares it with the original verify parameter. If they are equal, decrypting the transmission key with the shared key gets the true group key. In this way, each vehicle gets a group key with which they can communicate securely. If not, it indicates that the message may be tampered with or replayed by a malicious attacker during transmission.

Group Key Update When the Vehicle Enters: Assume that there are n vehicles in an area. If the new vehicle applies to enter this area, the original group key ought to update to ensure that the new vehicle cannot obtain the previous group key. There is the main updating process:

- First, the new vehicle ends an authentication message to the RSU, where elements included in the message denote the parameters for authentication of the new vehicle. The RSU verifies the new vehicle according to the message. Once the verification is passed, the RSU will calculate the shared key in the same method as the calculation of the shared key between the RSU and the vehicle.
- The RSU generates a random number and calculates the new group key in the period. It encrypts a new group key with the shared key to obtain the transmission key. Then it calculates the hashing pseudonym, transmission key, and shared key with the hash function to compute the verify parameter. The RSU sends a message including pseudonym, transmission key, and verify parameter to the vehicle.
- The RSU encrypts the new group key with the original group key to generate the verify group key. As in the calculation mentioned above, the RSU uses hash function to get the verify parameter. Finally, the RSU sends an update message to the original vehicles to update the group key
- After the new vehicle receives the verify message from the RSU, it calculates the verify parameter in the same way and compares it with the verify parameter in the message. If they are not equal, the message has been tampered with and replayed by a malicious attacker during transmission; otherwise, the shared key will be applied for decrypting the encrypted group key to get the true group key. Therefore, the new vehicle gets the new group key.
- After the original vehicle receives the update message, similarly, it will calculate its own verify parameter and make a comparison with the verify parameter in the message. The judgment of comparison is the same as the new vehicle. When they are equal, the group key is used as the shared key to decrypt the verify group key to get the true group key. Lastly, the original vehicles get the new group key.

Group Key Update When the Vehicle Leaves: When a vehicle requests to leave the area, the original group key should be updated as follows to make sure that the departing vehicle cannot obtain the new group key.

- When the vehicle leaves the area, make a hash operation of the pseudonym, leaving message, and shared key by hash function. The result is defined as h_l. At the same time, the vehicle sends the message hashed before to the RSU. Under these messages, the RSU calculates h_l* in the same way as h_l. The leaving request of the vehicle will be allowed only if h_l and h_l* are equal. After the vehicle leaves, the group key is updated.
- The RSU generates a random number and utilizes it to calculate the group key of the remaining vehicles. It encrypts the group key with the shared key for transmission. Calculating the verify parameter follows the same method as mentioned above. The RSU sends a message to each vehicle.
- After receiving the message, the vehicle calculates the verify parameter and compares it to the verify parameter in the message. If they are equal, decrypting the encrypted group key with the shared key will get the true group key. In this way, the remaining vehicles update the group key. Furthermore, the vehicle that has left cannot get the new group key.

Performance Evaluation

First, we conduct a simulation experiment to verify the security of our protocol. Second, our protocol is compared to the other three representative protocols in terms of authentication efficiency. Finally, we analyze the communication overhead of the protocols.

Security Analysis

We simulate our protocol and verify its security by adopting a tool named Automated Validation of Internet Security Protocols and Applications (AVISPA). We model the protocol in the High Level Protocol Specification Language (HLPSL) [12].

We implement two basic roles, *A* and *B*, where *A* denotes the RSU and *B* denotes the vehicle. The auxiliary role *session* describes the conversation of *A* and *B*. Another auxiliary role, *environment*, represents the operating environment of the protocol, including the enemy's ability to attack. In our model, we assume that the enemy is aware of the identities of the RSU and vehicle. Besides, the enemy is disguised as a non-malicious vehicle to take part in the communication.

This article takes advantage of a terminal in AVISPA named OFMC to operate the verification of the protocol. OFMC is a powerful verification terminal that can support algebraic rules and perform replay attack checking and Dolev-Yao model checking. In the experiment, it visits 21 nodes in depth 5 within 0.03 s. As shown in Fig. 2, the analysis result is SAFE.

In summary, our protocol guarantees secure authentication and communication. First, our protocol provides anonymous authentication. Vehicles use the pseudonym generated by the trust center TA, protecting the user's identity information. Second, the mutual authentication between the vehicle and the RSU can ensure the confidentiality of the data. After the mutual identity authentication of the vehicle and the RSU is realized, the RSU distributes group keys to the vehicles in the group. All vehicles in the group store the group key and use the group key to encrypt the message sent by the vehicle. If it can be decrypted correctly, the message is authenticated. Data confidentiality is guaranteed. The decentralized group key distribution mechanism guarantees forward and backward security. In addition, the protocol uses timestamps to resist replay attacks. Therefore, the protocol meets the security requirements of a VANET.

Comparison of Authentication Efficiency

In this subsection, we analyze the computational overhead during authentication. It is compared to the authentication algorithms presented in Zhang *et al.* [2], Majid *et al.* [3], and Wang *et al.* [4]. The method for time calculation is similar to that in [13], in which the execution time of authentication operations is calculated using MIRACL (Multi-precision Integer and Rational Arithmetic c/c++ Library). MIRACL is a famous cryptographic library for cryptographic operations in various environments [14]. We use a personal computer (HP with an Intel I7-4770 processor, 4G memory, Windows7 operating system) to calculate the running time of various operations in our protocol. The main operations in our protocol are the bilinear pairing operation, the bilinear-pairing-based point addition operation, and the bilinear-pairing-based scale multiplication operation. Furthermore, there are some other operations including the point addition operation on elliptic curve cryptography (ECC) and the scale multiplication operation based on ECC. There are regular hash operations and irregular hash operations; it indicates the mapping to the additive cyclic group.

In the analysis of the calculation cost, this article counts the vehicle computation cost (VCC), single authentication cost (SAC), and batch authentication cost (BAC) separately. VCC refers to the cost of vehicle anonymous identity generation and message signature. As shown in Fig. 3, it represents the cost of VCC and SAC. Figure 4 indicates the cost of BAC.

It can be seen from Fig. 3 that our protocol has superior performance to other protocols in terms of the vehicle computation cost. Specifically, it is 65 percent faster than that of Zhang *et al.* [2], 60 percent faster than that of Majid *et al.* [3], and 70 percent faster than that of Wang *et al.* [4]. As for the single authentication cost, our protocol takes less time than Majid *et al.* [3] and Wang *et al.* [4], but a little more than Zhang *et al.* [2] since the authentication methods of different protocols are composed of different time-consuming algorithms.

In order to analyze the influences on computation cost with the increase of vehicles in batch authentication, we have taken 30, 50, 70, and 90 vehicles for the cost comparison. As shown in Fig. 4, the slope of Majid *et al.* [3] and Wang *et al.* [4] is large. In other words, the increase of computation cost becomes faster and faster as the number of vehicles increases, which is likely to cause system collapse. How-

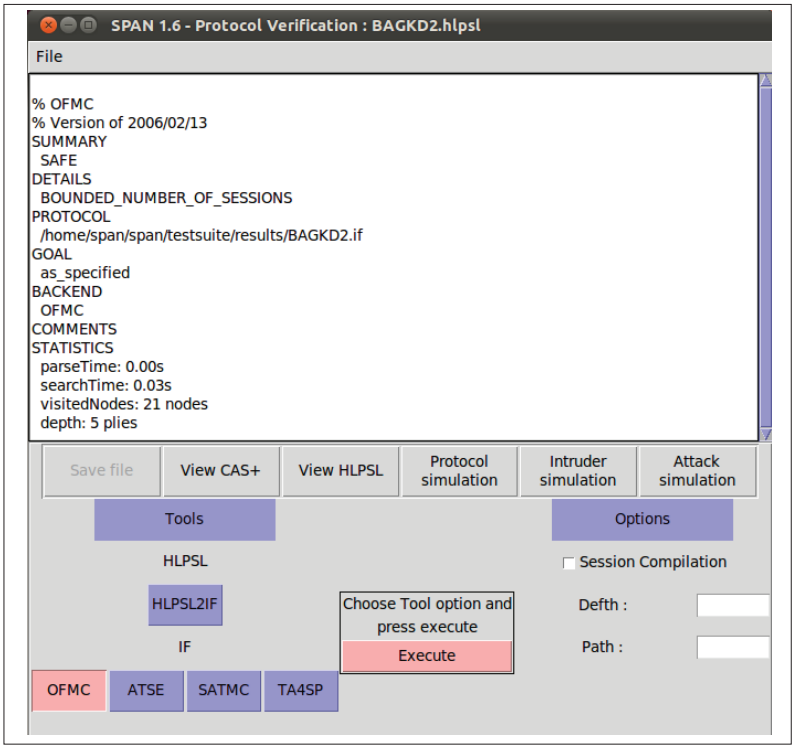

Figure 2. Security analysis result.

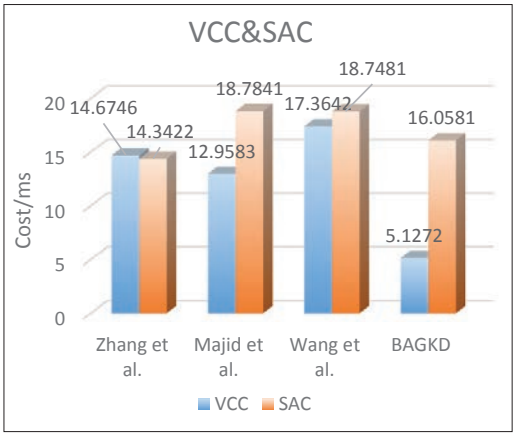

Figure 3. Vehicle calculation cost and single authentication cost.

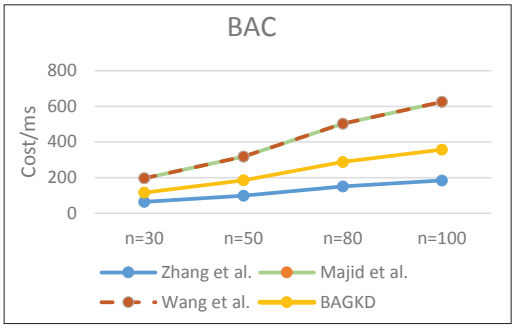

Figure 4. Batch authentication cost. n represents the number of vehicles in batch authentication.

Protocol	Year of publication	Single authentication	Batch authentication
Zhang *et al.* [2]	2014	388 bytes	388n bytes
Majid *et al.* [3]	2015	388 bytes	388n bytes
Wang *et al.* [4]	2018	512 bytes	512n bytes
BAGKD	2018	408 bytes	408n bytes

Table 1. Comparison of protocol communication burdens. n represents the numbers of vehicles in batch authentication.

ever, the cost increase of our protocol is slow with vehicle increase. As we can see, the protocol of Zhang *et al.* [2] takes less time than ours. Nevertheless, the authentication method of Zhang *et al.* [2] is worse as it is unable to resist a replay attack. This implies that our protocol is faster than the existing protocols while maintaining safety.

Comparison of Communication Cost

In this section, we compute the communication overhead of BAGKD and compare it to the above three mainstream protocols. We assume that the output of the regular hash is 10 bytes, the timestamp is 4 bytes, and the output of symmetric encryption is 10 bytes. Since the traffic information sent in the protocol is the same, only the signature and authentication information will be compared.

The communication overhead of the four protocols is shown in Table 1. By comparison, we can find that our protocol has a communication load 20 bytes higher than that of Zhang *et al.* [2] and Majid *et al.* [3]. However, it is 104 bytes lower than the communication burden of Wang *et al.* [4]. Therefore, under the premise of ensuring low computation cost and safety, our protocol has a relatively low communication burden.

Overall, the vehicle calculation cost of our agreement is at least 60 percent less than other methods in the experiment. Single authentication cost is 14 percent less than the scheme of Majid and Wang, and batch authentication cost is 43 percent less when they have more vehicles. The authentication cost is higher than Zhang's work, which cannot resist replay attacks. In order to maintain the integrity of the data, this part of the overhead is necessary. Therefore, our protocol can provide higher security performance under the premise of lower calculation cost and communication cost.

Discussion

In this article, we utilize batch authentication, which greatly improves the efficiency of authentication. Additionally, our protocol uses decentralized group key management, in which group members participate in the generation of group keys. It can adapt to the high-speed changing topology flexibly.

The extensive simulation results show that the protocol is safe. In particular, in the experiment of authentication efficiency, the cost of our protocol is confirmed to be slightly higher in the authentication phase than Zhang *et al.* [2], but from the perspective of security, our protocol is safer than that of [2].

As can be seen from the experimental results of the communication burden, our protocol is slightly higher than both Zhang *et al.* [2] and Majid *et al.* [3]. However, BAGKD is superior to these three existing mainstream protocols in terms of overall efficiency, safety, and communication burden. We can conclude that our protocol is generally superior to the three mainstream protocols for a large number of vehicles.

Conclusion and Future Work

In this article, we propose a BAGKD protocol for the communication among numerous vehicles in VANETs with a solid security proof. In comparison to the mainstream protocols in regard to efficiency and communication burden, we have found that our protocol takes less computing cost and resources but is safer and more efficient.

To further improve our protocol, we plan to work on the following topics in the near future:

- Apply biometric technology in VANETs to improve the convenience and security of the authentication process.
- Enhance the ability of the protocol to resist other malicious attacks.
- Improve the efficiency of the protocol while proving security.

Acknowledgments

This work is partially sponsored by the State Key Development Program of China (no. 2018YFB0804402, 2019YFB2101700) and the National Science Foundation of China (U1736115).

References

[1] M. Muhammad and G. A. Safdar, "Survey on Existing Authentication Issues for Cellular-Assisted V2X Communication," *Vehic. Commun.*, vol. 12, Apr. 2018, pp. 50–65.
[2] J. H. Zhang, M. Xu, and L.Y. Liu, "On the Security of a Secure Batch Verification with Group Testing for VANET," *Int'l. J. Network Security*, vol. 16, no. 5, Sept. 2014, pp. 355–62.
[3] B. Majid and R. Majid, "A Secure Authentication Scheme for VANETs With Batch Verification," *Wireless Networks*, vol. 21, no. 5, July 2015, pp. 1733–43.
[4] S. B. Wang and N.M. Yao, "LIAP: A Local Identity-Based Anonymous Message Authentication Protocol in VANETs," *Computer Commun.*, vol. 112, Nov. 2017, pp. 154–64.
[5] T. Alexander and W. Mazurczyk, "Mobile Communications and Networks," *IEEE Commun. Mag.*, vol. 57, no. 9, Sept. 2019, pp. 42–42.
[6] S. S. Manvi and S. Tangade, "A Survey on Authentication Schemes in VANETs for Secured Communication," *Vehic. Commun.*, vol. 9, July 2017, pp. 19–30.
[7] W. Mazurczyk *et al.*, "Traffic Measurements for Cyber Security," *IEEE Commun. Mag.*, vol. 55, no. 7, July 2017, pp. 12–13.
[8] R. Khanduzi and A. K. Sangaiah, "Tabu Search Based on Exact Approach for Protecting Hubs Against Jamming Attacks," *Computers & Electrical Engineering*, vol. 79, Oct. 2019, pp. 463–71.
[9] W. Z. Meng *et al.*, "Enhancing the Security of FinTech Applications With Map-Based Graphical Password Authentication," *Future Generation Computer Systems*, vol. 101, Dec. 2019, pp. 1018–27.
[10] Z. G. Zheng, A. K. Sangaiah, and T. Wang, "Adaptive Communication Protocols in Flying Ad Hoc Network," *IEEE Commun. Mag.*, vol. 56, no. 1, Jan. 2018, pp. 136–42.
[11] W. S. Yap, S. H. Heng, and B. M. Goi, "Security Analysis of M-DES and Key-Based Coded Permutation Ciphers in Wireless Channels," *IET Commun.*, vol. 12, no. 10, June. 2018, pp. 1230–35.
[12] M. Wazid *et al.*, "Design of Secure Key Management and User Authentication Scheme for Fog Computing Services," *Future Generation Computer Systems*, vol. 91, Feb. 2019, pp. 475–92.
[13] D. B. He, S. Zeadally, and B. W. Xu, "An Efficient Identity-Based Conditional Privacy-Preserving Authentication Scheme for Vehicular Ad Hoc Networks, " *IEEE Trans. Info. Forensics and Security*, vol. 10, no. 12, Dec. 2015, pp. 2681–91.

[14] G. Srivastava *et al.*, "An Efficient Public Key Secure Scheme for Cloud and IoT Security," *Computer Commun.*, vol. 150, Jan. 2020, pp. 634–43.

Biographies

Guangquan Xu [M'18] (losin@tju.edu.cn) is a full professor in the College of Intelligence and Computing, Tianjin University, China. He is a joint professor at Qingdao Huanghai University, China. His research interests include information security.

Xiaotong Li (lixiaotong@tju.edu.cn) is a Master's student in the College of Intelligence and Computing, Tianjin University. She received her B.S. degree from Shandong University of China in 2018. Her research interests include system security.

Litao Jiao (jiaolitao_11@163.com) received his M.B.A. degree in 2016 from Shandong University of Science and Technology. He is now an associate professor at Qingdao Huanghai University. His research interests include information security.

Weizhe Wang (will@tju.edu.cn) is a Master's student in the College of Intelligence and Computing, Tianjin University. He received his Bachelor's degree from Changchun University of Science and Technology in 2018. His research interests include cyber security.

Ao Liu (giususois@gmail.com) received his B.S. degree from Hebei University of Engineering in 2017. He is currently pursuing an M.E. degree at Tianjin University of Technology. His research interests include information security.

Chunhua Su (suchunhua@gmail.com) received his Ph.D. degree in computer science from Kyushu University in 2009. His research interests include information security.

Xi Zheng [M'16] (james.zheng@mq.edu.au) has a Ph.D. in software engineering from the University of Texas Austin, a Master's in computer and information science from the University of New South Wales, and a Bachelor's in computer information systems from FuDan. His research interests include information security and IoT.

Shaoying Liu [F'18] (shaoyingliu5@gmail.com) is a professor at Hiroshima University, Japan. He received his Ph.D. in formal methods from the University of Manchester, United Kingdom. His research interests include formal methods.

Xiaochun Cheng [SM'04] (xiaochun.cheng@gmail.com) received his Ph.D. in computer science in 1996 from Jilin University. He has been Computer Science EU Project Coordinator at Middlesex University since 2012. His research interests include information security.