A Cross-domain Data Sharing Scheme for VANETs Based on Blockchain

Wan-Yu Shang[1], Hai-Bing Mu[1], Jian-Xiong Liu[2]*

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, China
[2] Aerospace Science & Industry Network Information Development Co. LTD, China
22120111@bjtu.edu.cn, hbmu@bjtu.edu.cn, 13021097129@163.com

Abstract

With the continuous development of the Vehicular Ad Hoc Network (VANET), cross-domain sharing of vehicle data has become a significant concern. The Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm plays a key role in data sharing as it can achieve "one-to-many" transmission. In this paper, we propose a cross-domain data sharing scheme based on unpaired CP-ABE in VANETs, utilizing the blockchain and InterPlanetary File System (IPFS) system. The blockchain network is composed of trusted authorities (TAs) from different domains. Due to the high-speed movement characteristics of vehicles, we divide vehicle attributes into two categories: static and dynamic. we design a cross-domain data verification contract based on attribute bloom filter (ABF) for decryption testing. Vehicles that pass the test will receive dynamic attribute decryption keys generated by TAs in the data sharing domain to achieve cross-domain access. In addition, we design an outsourced decryption scheme to reduce the computational overhead during vehicle decryption and propose a direct permission revocation mechanism to ensure the flexibility and security of the system. The simulation experiment results show that our scheme optimizes the efficiency of cross-domain data access significantly compared with other approaches.

Keywords: VANET, CP-ABE, Data sharing, Blockchain

1 Introduction

Vehicular Ad Hoc Network (VANET) implements data transmission and communication among vehicles, roadside units, and Internet through wireless channel. The fast-moving vehicles with a highly dynamic network topology bring about the secure and efficient challenge in data sharing cross different domains. Blockchain, as a distributed, decentralized, and tamper-proof data storage technology, offers a solution for facilitating data interconnection among vehicles in different domains.

Traditional data sharing solutions often require pre-distribution of a large number of keys to support multiple identities and fine-grained access control. To solve these problems, Sahai and Waters [1] were the first to introduce the concept of attribute-based encryption (ABE) in 2005. ABE mainly consists of two categories: Key-Policy Attribute-Based Encryption (KP-ABE [2]) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE [3]). In CP-ABE schemes, only data visitors with attribute sets that match the access policy can decrypt the data. ABE schemes provide a flexible and efficient data protection mechanism for VANETs. By applying ABE schemes, vehicles can choose appropriate attributes and policies to encrypt or decrypt data according to different application scenarios and security requirements.

However, existing ABE schemes exhibit certain limitations in VANAET applications. Firstly, traditional CP-ABE schemes heavily rely on complex bilinear pairing operations, which are noted for the largest computational overhead in pairing-based cryptographic protocols. Secondly, in terms of permission revocation, most schemes use indirect revocation. This revocation method requires updating the ciphertext and key at the same time, which greatly increases the system overhead. Furthermore, most of the ABE schemes for VANETs struggle to distinguish between dynamic attributes and static attributes effectively, leading to a lack of model functionality and diminishing system performance.

In response to the aforementioned challenges, we propose a cross-domain data sharing scheme for VANETs based on blockchain architecture. The key contributions are outlined as follows:

1. Design a cross-domain trust center based on blockchain, which consists of trusted authorities (TAs) from different trust domains. The TA in the data sharing domain generates partial decryption keys for vehicles that meet the access policy and sends them to vehicles through the TA in the data access domain to achieve cross-domain data access.
2. Regarding the excessive computational overhead caused by bilinear pairing operations, we use the CP-ABE schemes based on Elliptic Curves Cryptography (ECC) and diminish the decryption burden on vehicles by outsourcing part of the decryption work to the roadside unit (RSU). The smart contract algorithm on the blockchain is used for decryption testing. It won't be necessary for vehicles to calculate whether the access policy is met, which improves the efficiency of accessing cross-domain ciphertexts. Aiming at the problem that the indirect revocation should update ciphertexts and keys frequently, we propose a direct permission revocation scheme to reduce communication overhead.
3. In this paper, attributes are categorized into static

*Corresponding Author: Hai-Bing Mu; E-mail: hbmu@bjtu.edu.cn
DOI: https://doi.org/10.70003/160792642024112506005

attributes and dynamic ones. Static attributes are managed by the blockchain, while dynamic attributes are submitted to the TA by vehicles. Additionally, to prevent the leakage of attributes in the access policy from exposing the user's private information, we adopt an access policy hiding algorithm based on the attribute bloom filter (ABF).

2 Related Works

CP-ABE is an attribute-based encryption scheme that can protect privacy and security effectively through fine-grained access control. Although CP-ABE schemes have shown great potential in practical applications, their efficiency and security issues remain the focus of research. Researchers have added many extensions to the ABE schemes, such as outsourced decryption, online/offline mechanism, multi-authority [4], traceability [5], revocability, and multi-keyword search.

In terms of improving efficiency, outsourced decryption schemes have been widely adopted in various ABE schemes. For example, Zhao et al. [6] implemented a lightweight CP-ABE scheme with key tracking and verification functions by combining outsourced decryption and online/offline mechanisms. Hu et al. [7] proposed a "test-decrypt-verify" scheme based on CP-ABE, which can return an intermediate value unrelated to the encrypted message during the outsourcing decryption process, thereby ensuring that the cloud server cannot obtain any valuable information. Nonetheless, outsourced decryption merely transfers the computational burden to a proxy server or a cloud server, without substantially reducing the computing overhead of the whole system. Therefore, it is crucial to identify an efficient arithmetic operation method that can replace complex bilinear pairing operations. Yao et al. [8] proposed a CP-ABE scheme based on ECC, which notably decreased the communication overhead and computational overhead of ABE, and discussed its limitations and avenues for enhancement. Qin et al. [9] improved the scheme in [8] and proposed an access control scheme suitable for the VANET, which combined ElGamal encryption to protect identity privacy and used outsourced decryption technology for lightweight decryption of vehicles. Das et al [4] proposed a fine-grained access control scheme for healthcare systems utilizing ECC and CP-ABE. This scheme distributes the key generation workload across multiple authorization authorities and overcomes key escrow problems effectively.

In terms of improving security, permission revocation plays a vital role, which can be divided into two types: direct revocation and indirect revocation depending on the executor. The direct permission revocation mechanism was first proposed by Ostrovsky et al. [10], but the scheme can only revoke users and causes an increase in the length of the ciphertext. Pirretti et al. [11] were the first to propose the indirect attribute revocation mechanism. In this scheme, during the system initialization process, the encryption party and the central attribute authorization center negotiate the lifecycle of each attribute. However, the system overhead increases significantly when the attribute lifecycle is short or the amount of user data is large, which constrains the scheme's practicability. Li et al. [12] proposed a user and attribute revocable CP-ABE scheme in the fog computing environment, which can revoke a user without updating the ciphertext by constructing a user group and updating the user group version key by utilizing fog nodes. Chen et al. [13] proposed a blockchain-based secure data sharing scheme, which combined outsourcing and attribute revocation, and built a consortium blockchain with RSUs as nodes to manage user attributes through KeK trees effectively. In recent years, as the focus on privacy protection concerns has grown, the privacy protection function of ABE schemes has garnered significant attention. In 2008, Nishide et al. [14] first proposed the concept of partial policy hiding, concealing attributes in the access policy by introducing wildcards to safeguard user privacy and security. Subsequently, various algorithms have been proposed to achieve access policy hiding. For example, Lai et al. [15] proposed an ABE scheme based on composite order bilinear groups, but the scheme exhibits high computational and space complexity. Yang et al. [16] proposed the concept of ABF, which achieves complete access policy hiding through ABF.

3 Proposed Scheme

The model of the cross-domain data sharing scheme for VANETs based on blockchain and ECC is shown in Figure 1. It comprises a total of 5 entities: InterPlanetary File System (IPFS), TA, consortium blockchain network (CBN), RSU, and vehicle.

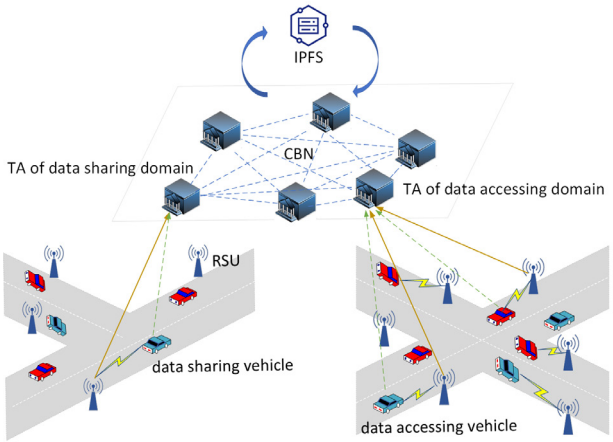

Figure 1. Scheme model

IPFS: IPFS is a distributed file system which stores data with multiple nodes to improve the reliability and availability of the VANET.

TA: TA is a trusted authority responsible for registering vehicles and generating keys.

CBN: The TAs from all domains form the CBN to help collaborate and share data. A cross-domain access verification contract is deployed on the blockchain for conducting decryption tests.

RSU: RSU is a communication device deployed on roadsides. Vehicles upload data to TA through RSU, and RSU is also responsible for outsourcing decryption.

Vehicles: vehicles act as data producers and sharers.

The symbols employed in this paper are described in Table 1.

Table 1. Description of symbols

Symbol	Description
PID	pseudonym
(M, ρ)	access policy
MSK_s	the master key of static attribute
MSK_d	the master key of dynamic attribute
PK_s	the public key of static attribute
PK_d	the public key of dynamic attribute
K_{ski_s}	decryption key of static attribute
K_{ski_d}	decryption key of dynamic attribute
K'_{ski}	conversion key
$Cred_i$	decryption credential

3.1 Scheme Initialization

3.1.1 Blockchain Configuration

It configures the node information of each domain TA, the consensus algorithm, block size, and so on. Generate the genesis block and deploy the following smart contracts:

Smart Contract 1: It manages the attribute master key corresponding to the static attributes of the vehicle and provides interfaces for storage and retrieval.

Smart Contract 2: It manages vehicle information, such as the vehicle's actual identity, general identity, static attribute set, and decryption keys of the static attributes, providing interfaces for storage and retrieval.

Smart Contract 3: It manages ciphertext storage addresses and access policies, providing storage and retrieval interfaces, as well as functions for decryption testing.

3.1.2 Public Parameters Initialization

The TAs on the blockchain negotiate public parameters. First, TAs choose a finite field GF(q) of order q and initialize the elliptic curve E. Then TAs initialize the ABF and its parameters, where L represents the size of the elements in ABF, L_a represents the maximum length of the attribute value att_n, and L_r represents the maximum length of the row number in the access control matrix, the hash function H_1, …, H_k are selected to map attributes into the ABF. TAs select the hash function H: $\{0,1\}\rightarrow Z_p$, H^*: G$\rightarrow Z_p$ and $H_0:\{0,1\}\rightarrow Z_p$, where H maps the PID into elements in Z_p, H^* is used in the vehicle registration phase and H_0 is applied to verify the decryption results. The scheme's public parameters are as follows:

$$\text{params} = \{\text{GF(q)}, \text{G}, \text{E}, L_a, L_r, L, H, H^*, H_0, H_1, \ldots, H_k\} \quad (1)$$

3.1.3 TAs and Vehicles Initialization

TA selects a random number η_i as the private key and calculates $T_{pub} = \eta_i$ G as the public key. The vehicle V_i selects a random number x_i as the private key and calculates $V_{pub} = x_i$G as the public key.

3.2 Vehicle Registration and Key Generation

The flowchart of the vehicle registration and key generation phase is displayed in Figure 2.

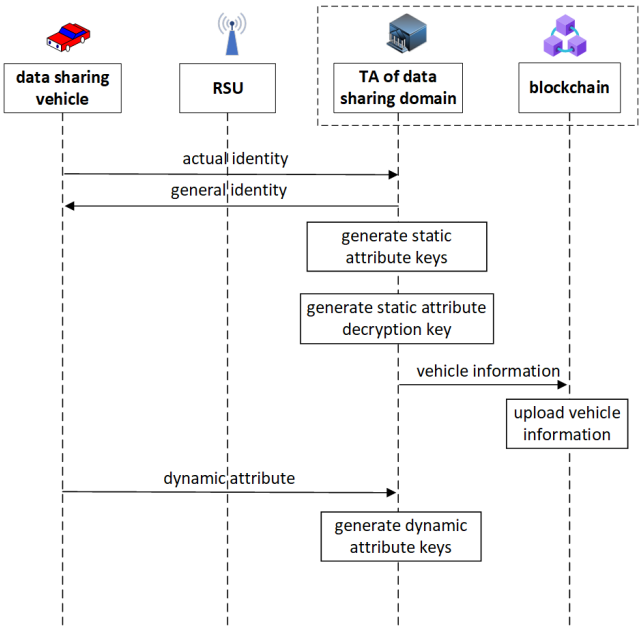

Figure 2. Vehicle registration and key generation flowchart

Vehicle V_i calculates $ID'_i = ID_i \oplus H^*(x_i \mathrm{T}_{\mathrm{pub}})$ and sends it to its domain TA through the public channel. After receiving ID'_i, TA gets the actual identity of the vehicle by calculating $ID_i = ID'_i \oplus H^*(\eta_i\ V_{pub})$. Then TA selects the random number α to calculate the vehicle V_i 's pseudonym $PID_i = ID_i + \alpha G$ and send it to the vehicle. TA obtains the static attribute set $U_i = \{att_i\}_{i\in[1,n]}$ of vehicle V_i from the traffic management department, and checks whether there is any corresponding attributes on smart contract 1. For attributes that are not on smart contract 1, TA selects random numbers y_{i_s} and $k_{i_s} \in Z_p$ and generates MSK_s as $\{y_{i_s}, k_{i_s}\}$ and PK_s as $\{y_{i_s}G, k_{i_s}G\}$, and store them in the smart contract1 so that each static attribute in the system corresponds to a unique public-private key pair. Then TA generates $K_{ski_s} = y_{i_s} + H(PID_i)k_{i_s}$ for the vehicle V_i and writes the actual identity, PID, static attribute set and static attribute decryption key to the smart contract 2, which constitutes a vehicle information registration form.

The data sharing vehicle sends the dynamic attribute set $\mathrm{U}' = \{att_{i_d}\}_{\mathrm{j}\in[1,\mathrm{d}]}$ to TA to be encrypted with the public key of the domain TA. TA selects random numbers y_{i_d} and $k_{i_d} \in Z_p$ for each attribute in the attribute set U′, and generates MSK_d as $\{y_{i_d}, k_{i_d}\}$ and PK_d as $\{y_{i_d}\mathrm{G}, k_{i_d}\mathrm{G}\}$ of the each dynamic attribute.

3.3 Policy Hiding and Data Encryption

The data sharing vehicle utilizes the ABF algorithm to conceal access policies and the ABE scheme to encrypt data. The flowchart of the policy hiding and data encryption phase is displayed in Figure 3.

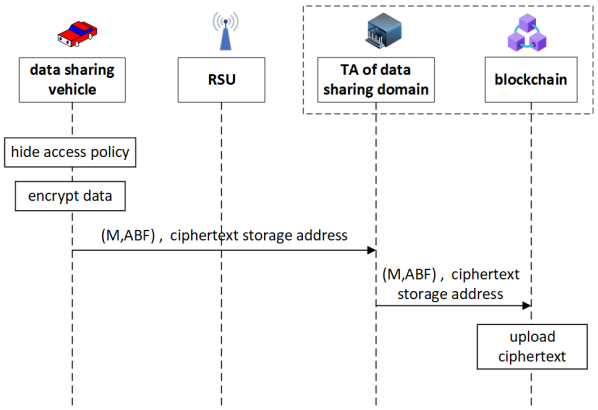

Figure 3. Policy hiding and data encryption flowchart

3.3.1 Access Policy Hiding

The data sharing vehicle defines an access control structure (M,ρ), where M is the access matrix. For example, (A ∨ B) ∧ (C ∨ (D ∧ E)) is an access policy specified by the data sharer. The Boolean function can be represented by an access tree as Figure 4. According to the LSSS matrix generation algorithm, M is expressed as equation (2), where ρ presents the mapping of row numbers to the attributes.

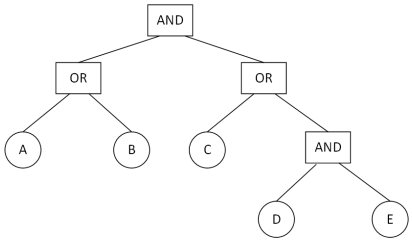

Figure 4. Access tree

$$\mathrm{M}=\begin{bmatrix}0 & -1 & 0 & \rho(1)=A\\ 0 & -1 & 0 & \rho(2)=B\\ 1 & 1 & 0 & \rho(3)=C\\ 0 & 0 & -1 & \rho(4)=D\\ 1 & 1 & 1 & \rho(5)=E\end{bmatrix} \tag{2}$$

First, the sharing vehicle binds the attribute value to the corresponding row number in M to obtain a mixed set $S_m = \{\mathrm{i}\|att_m\}_{\mathrm{i}\in[1,l]}$. Line numbers and attribute values that are smaller than the maximum length are padded with zero bits on the left, and two parts are added to the maximum length L_a and L_r. After that, k-1 strings of L-bit $r_{1,m}$, $r_{2,m}$, …, $r_{k-1,m}$ are chosen randomly, where $k > 1$, and the elements m in S_m are hidden through XOR operation, which is

$$r_{k,m} = r_{1,m} \oplus r_{2,m} \ldots \oplus r_{k-1,m} \oplus \mathrm{m} \tag{3}$$

Then, the attribute att_m associated with element m is hidden with k hash functions to obtain $H_1(att_m)$, $H_2(att_m)$, …, $H_k(att_m)$, where $H_i(att_m)$ represents the storage location of each random component $\mathrm{r_{i,m}}$ in this ABF, and the attributes corresponding to each row are computed to be hidden to obtain (M, ABF). If a conflict occurs when adding an element to the ABF, the original random component at this position is used.S

3.3.2 Data Encryption

The data sharing vehicle generates a symmetric key ck randomly and encrypts the data M to be shared with ck as $CT = Enc(M)_{ck}$, and calculates the hash value of the CT as $\mathrm{CT_H}=\mathrm{H_0(CT)}$. It choose random vectors $v = (\mathrm{s}, \mathrm{v_1}, \ldots, \mathrm{v_m})$ and $u = (0, \mathrm{u_1}, \ldots, \mathrm{u_m})$, where the random number s is the secret value to be shared in linear secret sharing. After that the data sharing vehicle calculates $\lambda_x = M_x \cdot v$, $\omega_x = M_x \cdot u$, where $x \in [1,l]$. The data sharing vehicle selects the random numbers $r_1 \in Z_p$, and C_0, $C_{1,x}$, $C_{2,x}$, $C_{3,x}$ are computed as

$$C_0 = ck + sG \tag{4}$$

$$C_{1,x} = \lambda_x G + r_1 y_{\rho(x)} G \tag{5}$$

$$C_{2,x} = r_1 G \tag{6}$$

$$C_{3,x} = r_1 k_{\rho(x)} G + \omega_x G \tag{7}$$

Ciphertext is constructed as $\mathrm{CT_{DO}} = \{(\mathrm{M,GBF}), C_0, (C_{1,x}, C_{2,x}, C_{3,x})_{\mathrm{x}\in[1,l]}, \mathrm{CT}, CT_H\}$.

3.3.3 Ciphertext Upload

The data sharing vehicle uploads the ciphertext $\{C_0, (C_{1,x}, C_{2,x}, C_{3,x})_{x\in[1,l]}, \mathrm{CT}, CT_H\}$ to IPFS, returns the storage address, and then uploads {(M, GBF), address} to TA, which add to the data into smart contract 3 on the chain.

3.4 Decryption Test and Data Decryption

The flowchart of the decryption test and data decryption phase is displayed in Figure 5.

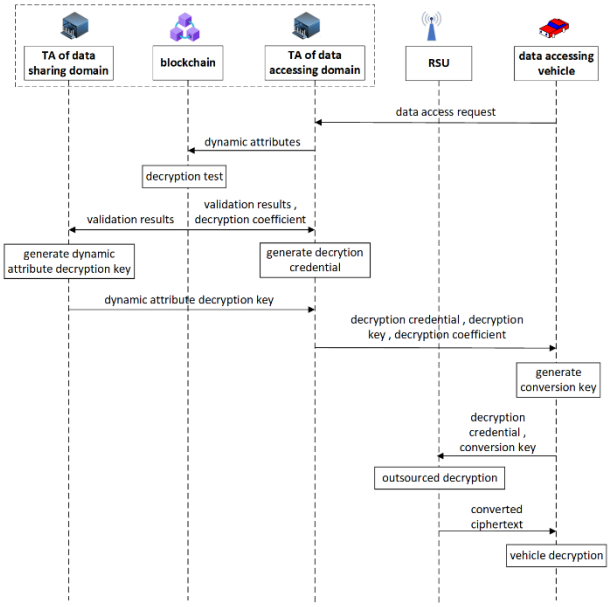

Figure 5. Decryption test and data decryption flowchart

3.4.1 Decryption Test

The data accessing vehicle encrypts PID_i, dynamic attribute set U^* and request domain number QTA with the public key of the domain TA and sends it to TA. TA decrypts it with his private key to get the dynamic attribute set U^*. Firstly, TA looks up the vehicle information registration form on the smart contract 2 to get the static attributes of the accessing vehicle, and then invokes the cross-domain access authentication algorithm in the smart contract 3 for decryption test.

K hash functions H_1, …, H_k are used to hash each attribute att_j to get $H_1(att_j)$, $H_2(att_j)$, …, $H_k(att_j)$, then get the corresponding strings $r_{1,m}$, $r_{2,m}$, …, $r_{k-1,m}$ at the position of $H_i(att_j)_{i\in[1,k]}$ in ABF. If there are strings at all k indexes, the reconstructed element m can be restored by using the k strings. m is computed as

$$m = r_{1,m} \oplus r_{2,m} \ldots \oplus r_{k-1,m} \oplus r_{k,m} \tag{8}$$

The element m consists of the attribute att_m and the row number x. The smart contract deletes all zero bits on the left side of the first L_r bit of the element to get the row number of the attribute in the access matrix, and deletes all zero bits on the left of the L_a bits counted from back to front to get the attribute att_m. If the att_m is the same as that in the set of attributes of the data accessing vehicle S, then the attribute is in the access policy, and restore the mapping $X = \{x \| \rho(x) \in S\}$. If the vehicle satisfies the access policy, then there is a set of coefficients $\{c_x \in Z_p\}_{x\in X}$ to meet $\Sigma_{x\in X} M_x \cdot c_x = (1,0,\ldots,0)$.

The TA queries the ciphertext storage address on the smart contract and generates a decryption credential $Cred_i$ signed with TA's private key, which includes the vehicle's general identity, ciphertext storage address, and timestamp.

3.4.2 Dynamic Attribute Decryption Key Generation

The TA of the data sharing domain calculates K_{skid} of the accessing vehicle that satisfies the access policy as follows:

$$K_{ski_d} = y_{i_d} + H(PID) k_{i_d} \tag{9}$$

Then, the TA sends $Cred_i$, K_{ski}, K_{skj} and decryption coefficient $\{c_x\}$ encrypted with the accessing vehicle's public key. The vehicle selects the random number $r_2 \in Z_p$ as the decryption private key and calculates the K'_{ski} for partial decryption of the RSU. K'_{ski} is computed as

$$K'_{ski} = y_i + H(PID) k_i + r_2 \tag{10}$$

3.4.3 Data Decryption

The vehicle sends the $Cred_i$ to RSU for identity authentication and RSU verifies the dynamic attributes of the vehicle and the validity period of the $Cred_i$. After passing the authentication, the RSU downloads the ciphertext from IPFS according to the ciphertext storage address, and the accessing vehicle sends the K'_{ski} and the decryption coefficient $\{c_x\}$ to the RSU with the public key. RSU performs the following calculations as

$$D_x = C_{1,x} - K'_{ski} C_{2,x} + H(PID) C_{3,x} \tag{11}$$

$$T_1 = \sum\nolimits_{x\in X} c_x \cdot D_x = sG - r_2 c_x r_1 G \tag{12}$$

$$T_2 = \sum\nolimits_{x\in X} c_x \cdot C_{2,x} = c_x r_1 G \tag{13}$$

where $\Sigma_{x\in X} c_x \cdot \lambda_x = s$ and $\Sigma_{x\in X} c_x \cdot \omega_x = 0$.

RSU sends the converted ciphertext $\{CT, C_0, CT_H, T_1, T_2\}$ to the data accessing vehicle. The vehicle calculates $ck' = C_0 - T_1 - r_2 T_2$ to obtain the symmetric key and decrypts C_0 with it to get the plaintext M. Vehicle verifies whether $CT_H = H_0(E_{ck}(M))$ holds. If it does, it means that the decrypted data has not been tampered with.

3.5 Permission Revocation

3.5.1 User Revocation

User revocation is achieved when the TA learns that the access permission of a certain vehicle is to be revoked, and the TA invokes smart contract 2 to remove the vehicle's record.

3.5.2 Static Attribute Revocation

When a vehicle's attributes are revoked, the TA invokes smart contract 2 to update the vehicle's records, removing the revoked static attributes and their corresponding decryption keys.

4 Security Analysis

4.1 Data Confidentiality

For attribute sets that do not satisfy the access policy, it is impossible to obtain decryption coefficients $\{c_x \in Z_p\}_{x\in X}$ in polynomial time to make equation $\Sigma_{x\in X} M_x c_x = (1,0,\ldots,0)$ hold, which results in the data accessing vehicle unable to obtain the decryption coefficients, decryption credentials, and decryption key. This effectively prevents malicious access and stealing behavior. In addition, the RSU lacks the vehicle's decryption private key r_2 and cannot completely decrypt the data to obtain the plaintext.

4.2 Forward Security

Forward security in user revocation: If the accessing vehicle attempts to initiate an access request after the permission has been revoked, the decryption test algorithm will not be triggered because the relevant vehicle information is no longer stored in the Smart Contract 2. Consequently, the vehicle will be unable to access crucial parameters like decryption coefficients, leading to ineffective data decryption.

Forward security in attribute revocation: If a vehicle with a revoked attribute initiates an access request, as the attribute is no longer stored in Smart Contract 2, it will no longer participate in the decryption test algorithm and the distribution of the attribute decryption key. Consequently, if the vehicle fails to meet the access policy, it won't decrypt the data.

In this research scheme, a vehicle will be failed to communicate with the RSU using its previous decryption credential because the embedded timestamps in the credential will restrict its validity to a short duration. Moreover, the decryption credential contains information about the ciphertext storage address. Therefore, the vehicle using an expired credential can't access the correct storage address of the current ciphertext and decrypt the valid plaintext.

4.3 Collusion Attack

The collusion attack occurs when multiple data users, who do not satisfy the access policy, collaborate to decrypt a ciphertext by sharing their attribute decryption keys. To defend against such attacks effectively, we ensure that the decryption key of each vehicle is bound to its pseudonym *PID* uniquely. For example, the access policy of a certain data sharing vehicle is ((taxi ∨ network car) ∧ street A ∧ driving westward). If Vehicle A owns the attributes: street A and driving westward while Vehicle B owns the attribute taxi, neither Vehicle A nor Vehicle B can decrypt the ciphertext independently. Their attribute set may satisfy the access policy only when they share the attribute decryption key. However, the ciphertext cannot be decrypted due to the fact that $H(PID_A) \neq H(PID_B)$, which cannot compute $\Sigma_{x\in X} c_x H(PID)\omega_x G \neq 0$ to obtain sG.

5 Performance Analysis

5.1 Theoretical Analysis

Our scheme along with the scheme in [17] and [18] is presented in Table 2. Compared with the other two schemes, the CP-ABE algorithm utilized in this scheme avoids the high computational cost of bilinear pairing operations, which significantly improves the efficiency of the system. Moreover, our scheme introduces policy hiding technology, which can effectively protect sensitive attribute information in access policies. In contrast to the scheme in [17], both our scheme and the scheme in [18] use outsourced decryption algorithms, which can significantly reduce the computational overhead of the vehicle in the decryption process. In addition, we design an efficient permission revocation mechanism. Therefore, our scheme exhibits notable performance advantages.

Table 2. Comparison of features

	[17]	[18]	Ours
Access policy	LSSS	LSSS +ABF	LSSS +ABF
Bilinear pairing	Yes	Yes	No
Decryption	vehicle	TA +vehicle	RSU +vehicle
Permission revocation	No	No	Yes
Ciphertext storage	edge vehicle +cloud	blockchain +IPFS	blockchain +IPFS

5.2 Experimental Analysis

Through simulation experiments, we compare and analyze the schemes in the following three aspects.

5.2.1 Vehicle Decryption

The vehicle decryption time is shown in Figure 6. From the figure, it is evident that both our scheme and the scheme in [18] employ outsourced decryption algorithms, and the user's computational overhead does not grow as the number of attributes increases. In contrast, the scheme in literature [17], where the decryption operations are performed by the vehicle, exhibits a linear increase in the decryption time for the vehicle user as the access policy complexity increases.

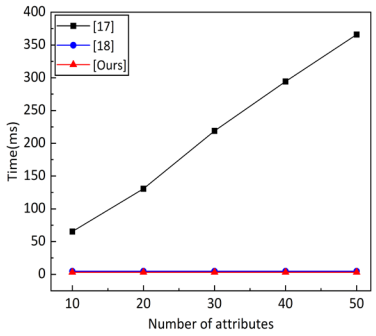

Figure 6. Vehicle decryption time comparison

5.2.2 Cross-domain Ciphertext Conversion

To achieve cross-domain sharing of data, the scheme in [17] obtains the target domain attribute list through cross-domain authentication, while the scheme in [18] adopts the method of generating conversion keys and conversion ciphertexts for target domain vehicles. The method proposed in our scheme is to generate dynamic attribute decryption keys for target domain vehicles to achieve cross-domain sharing of data. In this experiment, for our scheme, all the attributes in the access policy are assumed to be dynamic attributes. The time of cross-domain ciphertext conversion is defined as the time required from the generation of the dynamic attribute decryption key for the target domain TA to the outsourcing of the decryption by the RSU. The number of cross-domain requests at the same moment for both our scheme and the comparison scheme is 100, and all requests pass the decryption test.

As shown in Figure 7, the cross-domain ciphertext conversion time increases linearly with the number of attributes in the access policy. Our scheme, which avoids bilinear pairing operations, consumes much less time than other schemes.

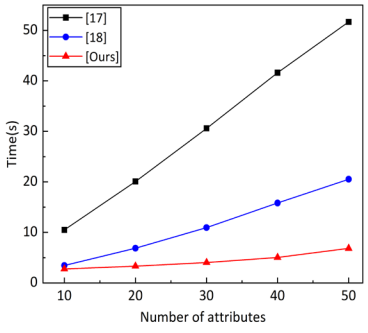

Figure 7. Cross-domain ciphertext conversion time comparison

5.2.3 Cross-domain Data Access

The time of cross-domain data access is shown in Figure 8, where the number of ciphertexts received by vehicles is 10. The performance of our scheme in cross-domain data access is much better than that of the scheme in [17-18], and as the number of attributes in the access structure increases, this gap will become more significant.

At the same time, both our scheme and the scheme in [18] adopt a cross-domain access verification method, which improves the efficiency of accessing a large amount of data.

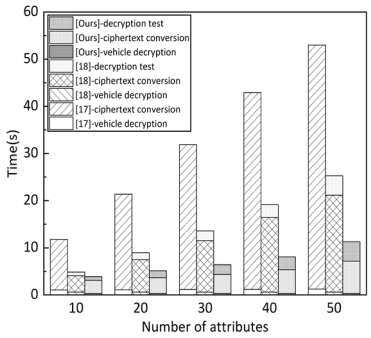

Figure 8. Cross-domain data access time comparison

6 Conclusion

In this paper, we propose a cross-domain sharing scheme based on pairing-free CP-ABE and blockchain for VANETs. Considering the high-speed movement characteristics of vehicles, we classify vehicle attributes as static and dynamic. To reduce the system's computational overhead, we employ simple scalar multiplication computation in elliptic curves. In addition, a safe and efficient outsourcing decryption algorithm is designed to reduce the user's computational overhead during decryption. Furthermore, we introduce a solution as direct revocation of user and static attributes. To address the cross-domain problem, we construct a blockchain network with TAs from different domains, in which the static attributes are managed by smart contracts on the chain, while the dynamic attributes are managed independently by TAs in each domain. By conducting the decryption test through smart contracts, the TA of the data sharing domain can generate dynamic attribute decryption keys for vehicles that pass the test to achieve cross-domain data access. Experimental results indicate that our scheme is more efficient than the existing schemes. However, the large amount of data generated by vehicles may have privacy implications, and in the future, we will focus on how to protect individual privacy in cross-domain data sharing.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2022JBZY002.

References

[1] A. Sahai, B. Waters, Fuzzy identity-based encryption, *Advances in Cryptology–EUROCRYPT 2005: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Aarhus, Denmark, 2005, pp. 457-473.

[2] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, *Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria, VA, USA, 2006, pp. 89-98.

[3] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy Attribute-based Encryption, *2007 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2007, pp. 321-334.

[4] S. Das, S. Namasudra, Multiauthority CP-ABE-based access control model for IoT-enabled healthcare infrastructure, *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 1, pp. 821-829, January, 2023.

[5] F. Cheng, S. Ji, C. F. Lai, Efficient CP-ABE scheme resistant to key leakage for secure cloud-fog computing, *Journal of Internet Technology*, Vol. 23, No. 7, pp. 1461-1471, December, 2022.

[6] Y. Zhao, H. Li, Z. Liu, G. Zhu, A lightweight CP-ABE scheme in the IEEEP1363 standard with key tracing and verification and its application on the Internet of Vehicles, *Transactions on Emerging Telecommunications Technologies*, Vol. 34, No. 7, pp. 1-16, July, 2023.

[7] G. Hu, L. Zhang, Y. Mu, X. Gao, An expressive "test-decrypt-verify" attribute-based encryption scheme with hidden policy for smart medical cloud, *IEEE Systems Journal*, Vol. 15, No. 1, pp. 365-376, March, 2021.

[8] X. Yao, Z. Chen, Y. Tian, A lightweight attribute-based encryption scheme for the Internet of Things, *Future Generation Computer Systems*, Vol. 49, pp. 104-112, August, 2015.

[9] X. Qin, Y. Huang, X. Li, An ECC-based access control scheme with lightweight decryption and conditional authentication for data sharing in vehicular networks, *Soft Computing*, Vol. 24, No. 24, pp. 18881-18891, December, 2020.

[10] R. Ostrovsky, A. Sahai, B. Waters, Attribute-based encryption with non-monotonic access structures, *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2007, pp. 195-203.

[11] M. Pirretti, P. Traynor, P. McDaniel, B. Waters, Secure attribute-based systems, *Proceedings of the 13th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2006, pp. 99-112.

[12] L. Li, Z. Wang. N. Li, Efficient attribute-based encryption outsourcing scheme with user and attribute revocation for fog-enabled IoT, *IEEE Access*, Vol. 8, pp. 176738-176749, September, 2020.

[13] X. Chen, Y. Chen, X. Wang, X. Zhu, K. Fang, DSVN: A Flexible and Secure Data-Sharing Model for VANET Based on Blockchain, *Applied Sciences*, Vol. 13, No. 1,

Article No. 217, January, 2023.
[14] T. Nishide, K. Yoneyama, K. Ohta, Attribute-based encryption with partially hidden encryptor-specified access structures, *International Conference on Applied Cryptography and Network Security*, New York, NY, USA, 2008, pp. 111-129.
[15] J. Lai, R. Deng, Y. Li, Fully secure ciphertext-policy hiding CP-ABE, *International Conference on Information Security Practice and Experience*, Guangzhou, China, 2011, pp. 24-39.
[16] K. Yang, Q. Han, H. Li, K. Zhang, Z. Su, X. Shen, An efficient and fine-grained big data access control scheme with privacy-preserving policy, *IEEE Internet of Things Journal*, Vol. 4, No. 2, pp. 563-571, April, 2017.
[17] J. Pan, J. Cui, L. Wei, Y. Xu, H. Zhong, Secure data sharing scheme for VANETs based on edge computing, *EURASIP Journal on Wireless Communications and Networking*, Vol. 2019, Article No. 169, June, 2019.
[18] X. Liu, T. Cao, Y. Xia, Research on efficient and secure cross-domain data sharing of IoV under blockchain architecture (in Chinese), *Journal on Communications*, Vol. 44, No. 3, pp. 186-197, March, 2023.

Biographies

Wan-Yu Shang is currently a graduate student at the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China. Her research intersets is network security.

Hai-Bing Mu received the Ph.D. degree from Beijing Jiaotong University in 2008. Now, she is an associate professor at BJTU. Her research interests include IoT and network security.

Jian-Xiong Liu obtained his master degree at the graduate school of China Aerospace Science and Engineering Second Institute. Currently, he works in Aerospace Science and Technology Network Information Development Co., Ltd. His research interests include cloud computing, and network security.