electronics

MDPI

Article

A Self-Sovereign Identity Privacy-Preserving Scheme for Logistics Transportation Based on One-Time-Use Tokens

Nigang Sun [1], Chenyang Zhu [2,*] and Yining Liu [3]

[1] School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213000, China; ngsun@cczu.edu.cn
[2] School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213000, China
[3] School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; ynliu@guet.edu.cn
* Correspondence: s22150812067@smail.cczu.edu.cn

Abstract: The advancement of the logistics industry has fostered the enhancement of operational efficacy within the socioeconomic domain. However, the current inevitable privacy leaks in the process of logistics transportation have seriously affected the development of the industry, which led to a crisis of consumer trust and even caused economic recession. This paper proposes a self-sovereign identity privacy protection scheme tailored specifically for logistics transportation contexts. First, the scheme entails furnishing users with one-time-use tokens while establishing decentralized identities capable of concealing identity information and ensuring the secure transmission of data. Furthermore, the scheme integrates fuzzy identity-based encryption to encrypt identity information, thereby guaranteeing the confidentiality and integrity of logistics user identities along with their associated data. Compared with other schemes, this scheme exhibits superior security in the realm of logistics transportation. Its targeted encryption technology and self-sovereignty strategies address the critical issue of privacy leakage, thereby safeguarding consumer privacy rights and interests while facilitating the sustainable development of the logistics industry.

Keywords: logistics; privacy protection; blockchain; digital identity; smart contract; identity encryption

check for updates

Citation: Sun, N.; Zhu, C.; Liu, Y. A Self-Sovereign Identity Privacy- Preserving Scheme for Logistics Transportation Based on One-Time- Use Tokens. *Electronics* **2024**, *13*, 2799. https://doi.org/10.3390/electronics13142799

Academic Editor: Dimitra I. Kaklamani

Received: 27 June 2024
Revised: 13 July 2024
Accepted: 15 July 2024
Published: 16 July 2024

1. Introduction

The rapid development of the logistics industry has promoted the optimized allocation of resources and provided strong support for the sustainable and healthy development of the socioeconomic domain [1]. However, with the continuous growth of logistics data [2], the problem of privacy leakage in logistics transportation has caused users to suffer economic losses and has had a significant negative impact on the development of the industry [3,4]. Data analysis can improve transportation efficiency [5], and digital transformation necessitates that enterprises focus on protecting user data privacy while benefiting from these efficiency gains [6]. At present, the negligence of logistics enterprises in data management cannot be ignored, especially the significant deficiencies in identity management protocols, which pose a serious threat to the personal information security of users. The significant increase in privacy leakage of logistics identity has prompted researchers to propose various solutions to cope with criminal activities, which can be summarized into three major kinds: pseudonym technology, decentralized storage, and digital identity. Pseudonym technology hides the user's true identity by assigning one or more untraceable identifiers to the user, thereby protecting their privacy. Previous studies [7–13] employ pseudonym technology, which facilitates users in retrieving information under pseudonymous identities, thereby obscuring their genuine identities. However, although pseudonym technology attempts to conceal the true identity of users, attackers can still track true identities through network traffic analysis and behavior pattern recognition [14]. Decentralized storage is not affected by a single node, avoiding the potential risk of single points of failure and data leakage in centralized storage. Previous studies [15–19] use data decentralized storage to enhance the security of distributed networks and ensure the availability and integrity of the data stored. Nonetheless, a significant drawback of decentralized storage is that it requires logistics companies to collect and store a large amount of detailed user identity information in databases, which significantly amplifies the potential risk of information leakage. Digital identity is based on blockchain technology, where a user's real identity can be replaced by public and private keys and addresses in the blockchain. Moreover, digital identity enables users to distinguish their identity without disclosing identity data, and selectively disclose identity data through authorization or restrictions, while avoiding the limitations and shortcomings of the previous two technologies. Additionally, digital identity is gaining wider and deeper recognition and application due to its outstanding performance in the field of privacy protection.

The latest stage of digital identity development is reflected in establishing self-sovereign identity (SSI), which aims to give users complete control over their identity data while ensuring the credibility and privacy of their identities [20,21]. Xiao et al. [22] proposed a new encryption and partial decryption mechanism by combining the blockchain-based SSI management (BbSSIM) scheme with thresholded ciphertext policy attribute-based encryption (CP-ABE). However, the information about verifiers is sent in the form of a list, which will cause the user's true identity to be leaked. Schanzenbach et al. [23] proposed an architecture that allows secure ID attribute sharing between holders and verifiers to enable the digital ID discovery process while avoiding dependence on a centralized identity provider (IdP). Soltani et al. [24] streamlined the data to facilitate the provision of services to holders by validators. However, whenever the holder requests data access, they must send text to the verifier, which will cause the digital identity to be associated with the true identity of the user. Stokkink et al. [25] realized a decentralized digital passport without permission during the deployment of SSI. Nevertheless, despite providing a high-performance SSI model, the same account is used in various transactions and services, which will lead to privacy breaches. Zebra [26] is the first on-chain verified anonymous credential based on zkSNARK. Yadav et al. [27] proposed a vehicle insurance blockchain framework to streamline accident reporting and insurance claim filing. Karmakar et al. [28] proposed an automated and tamper-proof framework based on the Ethereum blockchain with TOPSIS and smart contracts. Lux et al. [29] implemented a proof of concept decentralized by marrying OpenID Connect Provider with SSI. However, these schemes also use the same account in various transactions and services. As mentioned above, there are two problems with the extant schemes: the first pertains to the users' trade through the same account, and the second is that the digital identity directly connects the user's true identity. As a result, existing solutions cannot effectively address the dilemma of privacy leakage in logistics transportation.

This paper proposes a novel SSI scheme that specifically addresses the dilemma of privacy leakage caused by identical account transactions and digital identity associations by utilizing one-time-use tokens and fuzzy identity-based encryption (Fuzzy-IBE). The proposed scheme leverages decentralized identity and encryption technology to furnish data privacy safeguards for logistics users, which can avoid the problem of privacy leakage during logistics transportation. During the initial phase, the scheme generates a one-time-use token serving as the user's virtual identity, designed to expire after utilization, which improves the security of the SSI framework. This scheme adopts Fuzzy-IBE technology and significantly improves the privacy protection level of user information by implementing fuzzy identity, effectively preventing the stealing or peeping of the user's real identity. The scheme establishes a supervisory authority to supervise the activities of all participating entities, aiming to prevent contentious disputes and inappropriate utilization of accounts. Overall, the scheme demonstrates its capacity to withstand privacy breaches during logistics transportation while bolstering the supervisory mechanism to promptly detect and address potential violations.

2. Methodology

2.1. Blockchain

The pseudonymous person Satoshi Nakamoto proposed the concept of Bitcoin and successfully mined the first block of Bitcoin, the genesis block [30]. Blockchain is a chain structure composed of sequential links of blocks characterized by decentralization, immutability, and transparency. In terms of the underlying technology, researchers explore new consensus mechanisms, encryption algorithms, and storage technologies to improve the performance and security of the blockchain. For example, blockchain platforms like Ethereum adopt a more flexible smart contract mechanism, allowing the blockchain to support a wider range of application scenarios. In terms of privacy protection, researchers are committed to developing more efficient, secure, and anonymous privacy protection technologies to prevent the leakage of transaction information.

2.2. Smart Contracts

A smart contract is a computer program that automatically executes and implements specific conditions on the blockchain network [31]. Automatic execution means that the smart contract will automatically perform relevant operations without third-party intervention once the preset conditions are met. The emergence of smart contracts has dramatically enhanced the functionality and application scope of blockchain technology, allowing blockchain to expand from a single digital currency application to many fields, such as supply chain management, the Internet of Things, and digital copyright. The value of smart contracts is that they provide a new, decentralized trust mechanism. People can exchange value and collaborate through smart contracts without trusting third-party institutions. This trust mechanism not only reduces transaction costs but also improves the transparency and security of the system.

2.3. Decentralized Identity (DID)

Decentralized identity (DID) is a new type of identity identification mechanism that aims to solve problems in traditional identity management through decentralization, such as data privacy leaks, identity theft, and single points of failure in centralized institutions. The core idea of DID is to return the control and management of identity information to individuals or entities to achieve autonomous management and secure sharing of identity data. The key characteristics of DID include the following aspects.

Decentralization: DID does not rely on any centralized organization or server, but is based on distributed ledger technology to store and verify identity information. This makes DID more reliable and resistant to attacks. Verifiable: DID can generate verifiable credentials (VC) through interaction with other entities or services to prove the identity, attributes, or permissions of an individual or entity. These credentials can achieve privacy protection based on cryptographic techniques such as zero-knowledge proofs. Controllability: DID owners can independently manage their identity information, including choosing when and what information to share with whom. This helps protect personal privacy and prevent data misuse.

2.4. Self-Sovereign Identity (SSI)

Self-sovereign identity is an emerging identity management framework. The core idea of SSI is to give individuals complete control over their identity information to ensure that they can independently use their data. SSI emphasizes that no third party should monopolize or control personal identity data, but rather use decentralized, encrypted, and secure technologies to enable individuals to own and manage their digital identities.

In addition, SSI is a decentralized identity management system. It relies on verifiable data registers to verify decentralized identity (DID), which can be achieved through decentralized systems such as distributed ledger technology (DLT) and databases. SSI utilizes blockchain to store and manage identity information, eliminating reliance on centralized providers.

The core of SSI is verifiable credits (VC). W3C has released a formal recommendation for VC, defining it as a tamper-proof credential with a passwordable author identity. VC utilizes digital signature technology to ensure the authenticity and integrity of identity information. These credentials can be issued by trusted entities and verified through encryption to achieve secure and privacy-protected identity information sharing. VC has interoperability and supports selective disclosure of its user information. Each VC is issued on the decentralized identity (DID) of its holder and issuer and has the function of a public key.

In the SSI framework, an individual's identity data is stored in a decentralized, encrypted storage scheme, such as a distributed ledger or a secure multi-party computing environment. Individuals can control access to these data using private keys, ensuring that only authorized entities can access and verify their identity. In addition, SSI enables individuals to selectively share their identity information through VC mechanisms to meet the required authentication requirements in different situations.

Through the encryption mechanism of public and private keys, SSI allows individuals to securely share verified identity information with others without revealing unnecessary details. DID is a globally unique persistent identity composed of letters and numbers, directly associated with a pair of public and private keys. The private key allows users to access and manage their data. The user is the only person who knows the private key and does not share the private key with others. The DID protocol ensures that identity and credentials are encrypted and protected against unauthorized access or tampering with information.

2.5. Fuzzy-IBE

Fuzzy-IBE is a special public key encryption system that allows users to safely perform encryption and decryption operations when there is a certain degree of fuzziness or uncertainty in identity information [32]. Fuzzy-IBE has broad application prospects in identity management, access control, and privacy protection.

Fuzzy-IBE implementation usually relies on advanced cryptography techniques such as bilinear pairing. Specifically, the system establishes a public key space based on identity attributes, where each public key is associated with a set of attributes. Then, by defining a special encryption algorithm, when the sender encrypts using a public key that partially matches the receiver's attribute set, the receiver can still successfully decrypt using its private key. This partial matching property is achieved through a fuzzy extractor in cryptography, which can tolerate differences or noise in identity information to a certain extent.

2.6. One-Time Address

Monero is a digital currency that addresses user privacy breaches caused by transaction plaintext information recorded on public ledgers. Monero uses a one-time address solution. Users randomly generate private key k and public key $K = kG$, of which G is the base point of the elliptic curve.

The receiver generates a public key–private key pair k, K. Long-term public key K is published outside. When sending a transaction, the sender selects a random number r and sets a one-time public key $K' = rG + K$. While passing the random r to the receiver, the transaction is sent to K'. The hash function H maps the point on the ellipse curve into a scalar. In summary, the process selects a random number r, calculates $K' = H(rK)G + K, R = rG$, and sends R publicly to the receiver.

In the process of selecting r and calculating R, assuming that the value of r is randomly selected as d, which is a large constant, the operation on the elliptic curve will be calculated up to d-1 times. After receiving R, the receiver calculates $K' = H(kR)G + K$. The recipient compares the received R with K'. If they are consistent, the recipient can restore the one-time private key associated with the transaction using the formula $k' = H(kR)G + k$.

3. Scheme Design

This section describes the scheme's implementation details. Section 3.1 is the overall architecture. Section 3.2 is the process of the scheme, mainly including the generation of one-time-use tokens. Section 3.3 describes the deployment of smart contracts in this scheme. Section 3.4 contains relevant algorithms for Fuzzy-IBE. Section 3.5 provides a detailed introduction to the experiments on Fuzzy-IBE and data storage.

3.1. Scheme Architecture

The scheme includes issuers, logistics users, regulators, SSI BCN, and verifiers. The plan also includes smart contracts and identity encryption. The smart contract is deployed on the blockchain network and called in the transaction process. Identity encryption protects an individual's identity from leakage, ensuring personal privacy and data security. In addition, there are logistics-related entities, including trucks and employees performing transportation tasks. The architecture of the scheme is shown in Figure 1.

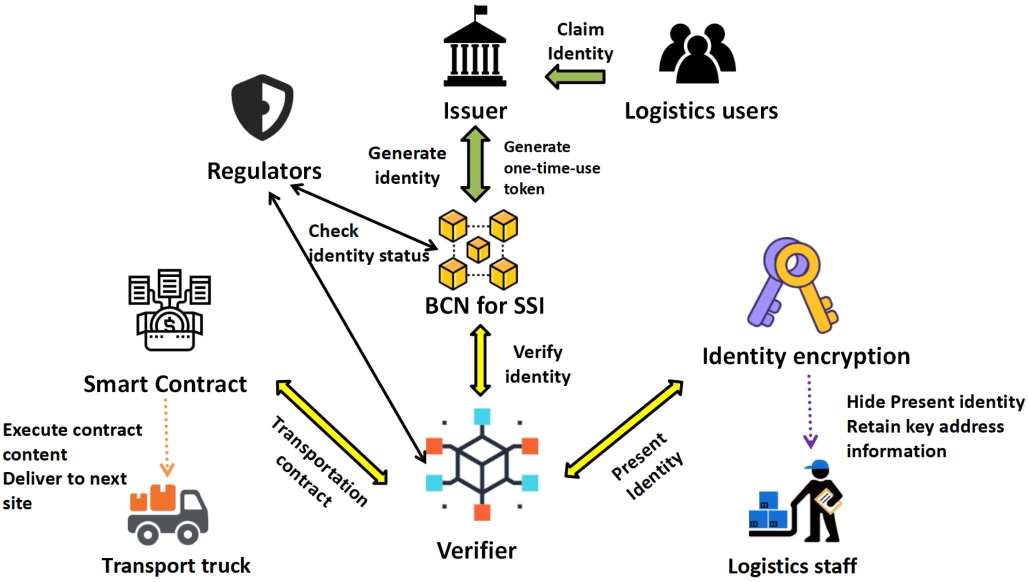


Figure 1. Scheme architecture.

Components of the Architecture

(1) Issuer: A trusted third-party organization is responsible for issuing and managing digital certificates. These digital certificates are used to verify network entities' identity and ensure communication's confidentiality and integrity. (2) Logistics users: Users have unique identity credentials and can autonomously manage them and selectively share them for secure and controlled interaction with other entities. (3) Regulators: The entity establishes the network's legal framework and guidelines for identity management. It defines the rules and standards for identity issuance, verification, and interaction. (4) SSI BCN: This blockchain infrastructure forms the cornerstone of the SSI scheme and is responsible for securely recording and managing identity-related transactions, credentials, and interaction information to ensure data security and integrity. (5) Verifier: The verifier requests an authentication operation or establish a secure communication channel.

3.2. Scheme Process

3.2.1. Secret Exchange Protocol

A secret exchange protocol is used to transfer r to the recipient in a private manner. Elliptic-curve Diffie–Hellman (ECDH) is an anonymous key agreement protocol. In the formula that the protocol depends on, $(a \cdot G) \cdot b = (b \cdot G) \cdot a$, where a and b are constants. The sender and receiver determine the parameters and fundamental point G of the elliptic curve, as shown in Table 1. These parameters are all publicly available.

Table 1. Symbol comment.

Notation	Descriptive
a, b	Constants
k_s	Shared key
p, g, g^a, g^b	Positive integers
X, Y	Formula variable
S, S_1, S_2	Secret key
A, OA	User addresses and one-time addresses
x, y	Elliptical curve formula variables
m	Modulo
G	Parameters on elliptic curves
C	Address generation function
r, r_r, r_s	Random numbers on elliptic curves
R, R_r, R_s	The random number after multiplication with the common parameter G
$Hash$	Hash operation
K_a, K_b, K_r, K_s	Receiver and sender public keys
k_a, k_b, k_r, k_s	Receiver and sender's private keys
K_{r_1}, K_{s_1}	One-time public key from hash calculation
sk, sk_r, sk_s	Private key from the hash calculation
pk	Public key

The sender generates a random ECC key pair (public key: $K_a = k_a G$, private key: k_a). The receiver generates a random ECC key pair (public key: $K_b = k_b G$, private key: k_b). The sender multiplies the recipient's public key by their private key to obtain the shared key $k_s = (k_b G) k_a$. Similarly, the recipient multiplies the sender's public key by their private key to obtain the shared key $k_s = (k_a G) k_b$. According to the formula mentioned earlier, $(a \cdot G) \cdot b = (b \cdot G) \cdot a$, the shared key obtained is the same for both the sender and receiver. Through this method, both parties completed the key exchange through the protocol.

Even if some clients do not support elliptic curves, the key exchange process can still be effectively completed using alternative methods. In key exchange, ECDH can be used as the preferred method, and If ECDH is not supported, the system can fall back to using traditional Diffie–Hellman or other alternative methods. The sender and receiver agree to use two positive integers p and g. The sender selects a secret integer a, calculates $X = g^a$ mod p, and sends it to the receiver. The receiver selects a secret integer b, calculates $Y = g^b$ mod p, and sends it to the sender. The sender calculates $S_1 = Y^a \mod p$. The receiver calculates $S_2 = X^b \mod p$. Given that $(Y^a \mod p) = (g^{ab} \mod p) = (X^b \mod p)$, therefore $S_1 = S_2 = S$. In this way, both parties negotiated the key S.

3.2.2. One-Time-Use Tokens

The scheme adopts a one-time address-generation mechanism to enhance transaction security and privacy protection and create one-time-use tokens. The core of this mechanism uses encryption algorithms to generate unique and unpredictable addresses for individual transactions.

This mechanism aims to generate a unique one-time-use token for each request or operation to ensure that each interaction is independent and cannot be reused.

To ensure data integrity and identity verification during the communication process, the Ed25519 algorithm was adopted in the scheme. The Ed25519 algorithm is currently the fastest elliptic curve encryption algorithm with extremely high security. The curve equation is $y^2 = x^3 + 486662x^2 + x$, $-x^2 + y^2 = 1 - \frac{121665}{121666}x^2y^2$, modulo $m = 2^{255} - 19$; G is the base point on the curve, also known as the generator. The key pairs generated using these specific parameters include a 32-byte private key and a corresponding 32-byte public key. In addition, the signature length generated by the Ed25519 algorithm is 64 bytes, providing high efficiency and security protection.

Step 1: Address generation function. Let C be the address generation function, which takes a random number r and a base address A as inputs, where r is generated by a secure random number generator (RNG), as shown in Equation (1). A new random number r is generated during each transaction. Each generated r is unique and unrelated to the previous r value. The constant r is generated locally by the data sender, so the data sender knows r, but the data receiver and other third parties usually do not know the specific value of r.

$$OA = C(r, A) \tag{1}$$

Step 2: Generate an essential public–private key pair. Use the ECC algorithm to generate essential public keys pk and sk.

Step 3: The receiver and sender each randomly generate private keys, denoted as k_r and k_s, respectively. Then, they disclose their corresponding public keys to each other, denoted as $K_r = k_r G$ and $K_s = k_s G$.

Step 4: The receiver generates a random value r_r, to compute R_r and transmits R_r to the sender, as shown in Equation (2).

$$R_r = r_r G \tag{2}$$

Step 5: Upon receiving R_r, the sender generates another random number r_s, and computes the receiver's one-time public key, denoted as K_{r1}, as shown in Equation (3).

$$K_{r_1} = Hash(r_s K_r)G + K_r \tag{3}$$

Step 6: The sender computes the one-time public key, denoted as K_{s1}, following the expression shown in Equation (4). It is important to note that only the sender possesses the private key sk_s, as shown in Equation (5). In the equation, the data are processed using the Keccak256 hash algorithm. The structure of this algorithm gives it excellent sensitivity and collision resistance. It converts input data into fixed-length hash values, which are then used in subsequent calculations.

$$K_{s_1} = Hash(k_s R_r)G + K_s \tag{4}$$

$$sk_s = Hash(k_s R_r) + k_s \tag{5}$$

Step 7: The sender employs K_{s1} to authenticate with the logistics company and uses the private key to digitally sign the intended shipping address. Subsequently, the sender designates the receiver as K_{r1} and proceeds to compute R_s, transmitting it to the receiver, as shown in Equation (6).

$$R_s = r_s G \tag{6}$$

Step 8: The receiver acquires R_s and computes K'_{r1} according to Equation (7). The receiver sets their address to K_{r1}, possessing their private key sk_r as indicated in Equation (8). Consequently, the receiver utilizes K_{r1} to log in to the account and employs their private key to digitally sign the recipient's address.

$$K'_{r_1} = Hash(k_r R_s)G + K_r = K_{r_1} \tag{7}$$

$$sk_r = Hash(k_r R_s) + k_r \tag{8}$$

Step 9: A one-time public key is calculated using a hash function to obtain a fixed-length hash value for the one-time address OA, as shown in Equation (9). Similarly, in Equation (9), the one-time address OA is determined using the Keccak256 hash algorithm, as described in Step 6.

$$OA = Hash(K_s, K_r) \tag{9}$$

Once the address OA is generated, a one-time-use token bound to it will be created by inputting OA, transaction amount, recipient address, and other information as part of the token function.

At the same time, one-time-use tokens provide secure access to logistics data. During logistics and transportation, the distributed storage system ensures that data are protected during storage and transmission through decentralized and immutable characteristics. Certificate authority (CA) plays a crucial role in managing digital certificates. Digital certificates provide identity verification and authorization mechanisms for each participant in the scheme, ensuring that only legitimate entities can participate in data processing and management. In this scheme, logistics data management is implemented through smart contracts to ensure the authenticity of the data, as shown in Figure 2. Finally, data tourists can access the data through a one-time-use token login system and upload their data to the data system.

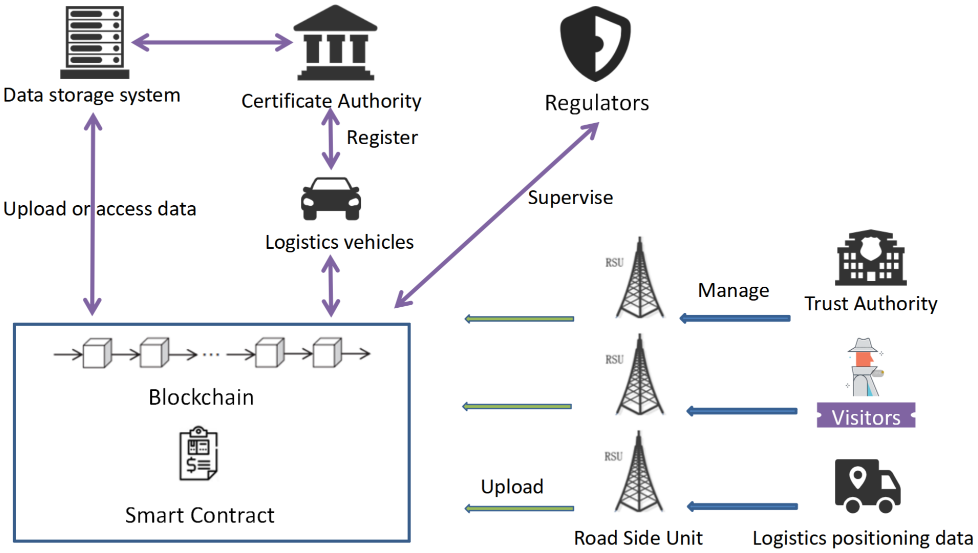


Figure 2. Upload or access logistics data.

3.3. Smart Contracts

Smart contracts' characteristics include automation, transparency, and immutability, which make them highly reliable and efficient in executing contract terms.

In logistics transportation, smart contracts have the profound ability to monitor the condition of goods in real time, ensuring strict compliance with contract provisions and further optimizing the transportation process, while significantly reducing human errors and fraudulent activities. In addition, smart contracts can automatically perform various operations, including contract signing, updating the status of goods, payment processing, dispute resolution, etc.

Algorithm 1 defines the logistics transportation contract. First, it creates a structure that includes the unique identity of the contract, sender and receiver addresses, goods description, shipping status, etc.

Algorithm 1 contract LogisticsContract

```
Input: uint256 contractId,
       address sender,
       address receiver,
       string goodsDescription,
       string transportStatus
```

Algorithm 2 creates a new logistics transportation contract. First, it assigns a new contract ID, creates a new transport structure, and stores it on the transport map. It then triggers the ContractCreated step, after which the newly created contract ID is returned.

Algorithm 2 createContract() public

```
Input: address receiver,
       string memory goodsDescription,
       uint256 fee
  uint256 contractId = nextContractId++;
  transports[contractId] = Transport();
  emit ContractCreated();
  return contractId;
```

Algorithm 3 is a function that updates the status of goods. It verifies that only the authorized sender or receiver possesses the capability to update the status, thereby ensuring the validation of any status changes and facilitating the subsequent update of the shipping status. If the status is updated to “Delivered”, it checks whether it has been paid and marks the contract as complete.

Algorithm 3 updateStatus() public

```
Input: uint256 contractId,
       string memory newStatus
  require(transports.sender == msg.sender);
  require(newStatus!=transportStatus);
  transports.transportStatus = = newStatus;
  if(newStatus== (“Delivered”));
  transportStatus = “Completed”;
```

Algorithm 4 is the pay shipping function. The amount requested to be paid must be consistent with the freight amount specified in the contract, and the address must be consistent with the recipient’s address in the contract. After the freight payment is successful, the freight for the contract will be marked as paid.

Algorithm 4 payFee() public

```
Input: uint256 contractId
  require(msg.value == transports.fee);
  require(transports.receiver == msg.sender);
  transports.isPaid = true;
```

Algorithm 5 checks whether the contract is completed. It calculates the hash of the shipping status in the contract and compares it with the hash of the “Completed” string. If equal, it returns true, indicating that the contract is completed.

Algorithm 5 isContractCompleted() public view

```
Input: uint256 contractId
  return keccak256(bytes(transportStatus)) == keccak256(bytes(“Completed”));
```

Algorithm 6 checks whether the contract is breached. If the block time exceeds the contract’s delivery deadline and has not been completed, it sets the contract’s shipping status to “Breached”. It triggers contract breach and delivers relevant information.

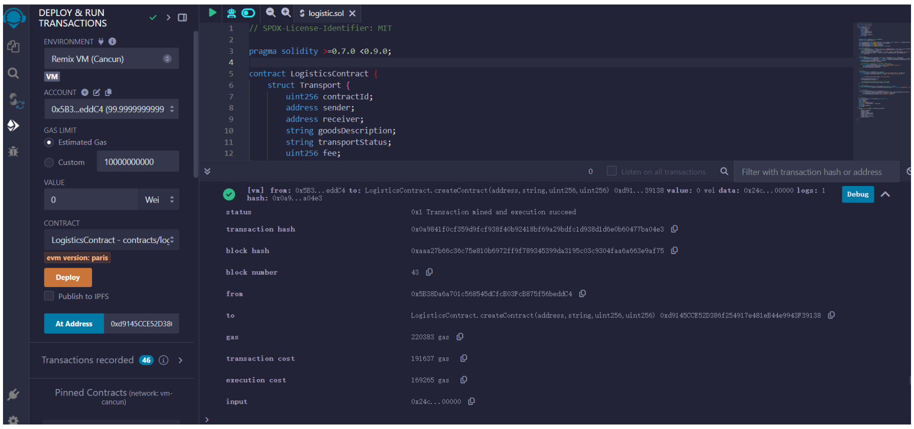
Algorithm 6 checkContractBreach() public view

```
Input: uint256 contractId
  1.if (timestamp > deliveryDeadline !isContractCompleted);
  2.transportStatus = "Breached";
```

In logistics, the decentralized platform Ethereum can be used to manage the transportation of goods, payment settlement, and handover of goods. When the goods arrive at a particular node, the smart contract automatically triggers the payment operation to ensure the security of the transaction. It uses the interfaces provided by libraries such as Web3.js and Ethers.js to call deployed smart contracts, and the functions of the contracts will perform logistics and transportation-related operations. Account owners with the required permissions can register in the logistics system. This means that users must provide identification to associate themselves with their packages. During registration, the sender registers their package and uploads both the package and recipient addresses. In the logistics framework, users and packages are assigned an address after registration.

A transport structure is defined in the contract as storing the primary information about the package and the address of its owner. It uses a mapping data structure to store all transportation information, where the key is the contract ID. The sender can create a new logistics contract through the 'createContract' function. This function requires the recipient's address, goods description, cost, and delivery deadline as parameters. The recipient can pay the shipping fee through the payFee function, which requires the contract ID as a parameter. The function checks whether the paid amount matches the contract amount and whether the fees have not been paid. When the package is delivered to the recipient, ownership changes, and information is updated. The contract also has an updateStatus function that allows the sender or recipient to update the transportation status of the contract. Only the shipper or recipient can call the updateStatus function to change the transportation status of the contract. The isContractCompleted function allows anyone to query whether a contract has been completed. The checkContractBreach function can check whether a given contract is breached due to exceeding the delivery deadline. Sender and recipient can conduct secure logistics transactions on a decentralized platform through these features.

Remix is an online integrated development environment (IDE) for Ethereum. We tested the deployed smart contracts in the Remix environment to ensure their functionality meets expectations, as shown in Figure 3. In Ethereum, Gas is a pricing unit that measures the execution of smart contract operations. The Gas value determines the workload of smart contract operations and quantifies the cost of using smart contracts within the system. After executing the function, Remix will display a transaction details window, which includes the transaction's gas usage. The gas consumption of each function is shown in Figure 4, with the highest cost being the createContract function. The remaining functions have a gas consumption of less than 50,000, which aligns with actual expectations.



DEPLOY & RUN TRANSACTIONS

ENVIRONMENT: Remix VM (Cancun)

ACCOUNT: 0x5B3...eddC4 (99.9999999999

GAS LIMIT: ● Estimated Gas ○ Custom 10000000000

VALUE: 0 Wei

CONTRACT: LogisticsContract - contracts/log

evm version: paris

Deploy

□ Publish to IPFS

At Address 0xd9145CCE52D38

Transactions recorded 46

Pinned Contracts (network: vm-cancun)

logistic.sol

```
// SPDX-License-Identifier: MIT

pragma solidity >=0.7.0 <0.9.0;

contract LogisticsContract {
    struct Transport {
        uint256 contractId;
        address sender;
        address receiver;
        string goodsDescription;
        string transportStatus;
        uint256 fee;
```

```
[vm] from: 0x5B3...eddC4 to: LogisticsContract.createContract(address,string,uint256,uint256) 0xd91...39138 value: 0 wei data: 0x24c...00000 logs: 1
hash: 0x0a9...a04e3
status              0x1 Transaction mined and execution succeed
transaction hash    0x0a9841f0cf359d9fcf938f40b92418bf69a29bdfc1d938d1d6e0b60477ba04e3
block hash          0xaaa27b66c36c75e810b6972ff9f789345399da3195c03c9304faa6a663e9af75
block number        43
from                0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                  LogisticsContract.createContract(address,string,uint256,uint256) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas                 220383 gas
transaction cost    191637 gas
execution cost      169265 gas
input               0x24c...00000
```

Filter with transaction hash or address
Listen on all transactions
Debug

Figure 3. Remix smart contract deployment.

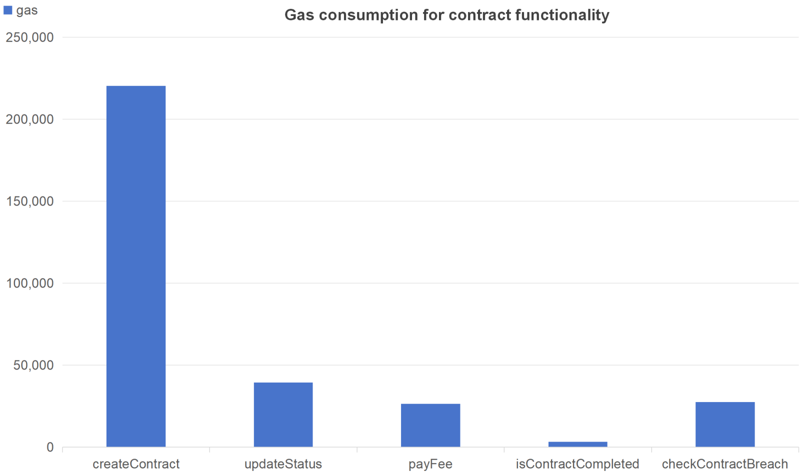


Figure 4. Gas consumption for contract functionality.

3.4. FUZZY-IBE Process

The proposed scheme harnesses Fuzzy-IBE to obfuscate addresses and veil the identity information of logistics users. Such an approach fortifies resilience against data leakage from malicious attacks during logistics transportation, thereby enhancing identity privacy protection.

In Fuzzy-IBE, the user's accurate identity information is not directly used as the public key. A fuzzy identity identifier is used, including a part of the user ID and attributes related to the user's identity, such as the user's location and type. As a public key, this blurred identity can effectively prevent attackers from directly associating intercepted information with specific individuals. When sending logistics information, it uses an identity identifier as the public key to encrypt the data. In this way, even if information is intercepted during transmission, attackers cannot directly associate it with specific individuals, as the public key does not directly expose the user's true identity.

Fuzzy-IBE can protect the security of logistics information. In logistics and transportation, a large amount of sensitive data, such as product information, transportation routes, and consignee information, are involved. Only users who meet specific attribute conditions can decrypt and access relevant data, thus ensuring the confidentiality and integrity of logistics information, as shown in Algorithm 7.

Algorithm 7 Encryption algorithm

```
Input: messageAttList, plaintext, userAttList
Output: delivery result
  Initialization                                                    ▷ generate parameters
  Determine the integer set $U$;Determine the system threshold $d$
  Generate swarm elements $g$
  Generate random number $y$ and $t_i$
  Generate $g^{t_i}$
  Keygen                                                            ▷ calculate the private key
  mskProp←$(t_1, t_2, \ldots t_u, y)$
  pkProp←$(T_1, T_2, \ldots T_u, y)$
  Generate random elements $d$-1, make $q(o) = y$
  //Calculate the private key $g^{q(i)/t_i}$, where $q$ is the value of the polynomial at that attribute
  position and $t$ is the master key corresponding to the attribute
  for userAttList do
      Target each attribute in the user's property collection
      if $D_i = g^{q(i)/t_i}$ then
          skProp←$(D_i)$
      end if
  end for
  Encrypt ← messageAttList                                          ▷ encrypting plaintext
  Generate random number $s$
  Calculate the ciphertext component ct
  Plaintext $M \in G_T$
  Calculate $E' = MY^s = Me(g,g)^{ys}$
  for messageAttList do
      Calculate $E_i = T_i^s$
      ciphertext $ct = (E', E_i)_{i \in W}$
  end for
  Decrypt ← messageAttList                                          ▷ decrypting plaintext
  Load ciphertext, private key
  //Check if the number of overlapping attributes in two lists is less than $d$
  for messageAttList,userAttList do
      if Overlapping attributes >= $d$ then
          calculate $P_i = e(E_i, D_i)^{\delta_i(0)} = e(g,g)^{s^{q(i)}\delta_i(0)}$
          Plaintext $M = E'/\Pi_{i \in I} P_i$
      end if
  end for
```

3.4.1. Initialization

Step 1: Generate pairing-related public parameters(e, g, G_1, G_T, Z_r), as shown in Table 2.

Step 2: Determine the complete set of attributes U as a set of integers$(1, 2, \ldots, |U|)$, and simultaneously determine the system threshold d.

Step 3: Select a random number $t_i \in Z_r$ as the master key component for attribute i, and calculate T_i as the corresponding public key component, as shown in Equation (10).

$$T_i = g^{t_i} \tag{10}$$

Step 4: Select a random number $y \in Z_r$, and calculate Y, as shown in Equation (11).

$$Y = e(g,g)^y \tag{11}$$

Step 5: Calculate the system master key msk and public key pk, as shown in Equations (12) and (13).

$$msk = (t_1, t_2, \ldots, t_{|U|}, y) \tag{12}$$

$$pk = (T_1, T_2, \ldots, T_{|U|}, Y) \tag{13}$$

Table 2. Symbol comment.

Notation	Descriptive
e, g	The public parameters
G_1, G_T	Random group element
Z_r	Set of integers
U	Complete set of properties
D_i	Random number calculation private key
δ_i	Lagrange factor
Y	Random number calculation public key
pk	The public key
sk	The private key
t	Random number
M	Plaintext message
H	Hash operation
ct	Standardized ciphertext
S	User attribute set
W	Plaintext attribute set
T	Public key component
I	Attribute collection
i	Attributes node

3.4.2. Keygen

Step 1: Randomly select a polynomial $q(x)$ of degree d-1, where $q(0) = y$.

Step 2: For attribute i in user attribute set S, calculate $q(i)$. Then, further calculate D_i, as shown in Equation (14).

$$D_i = g^{q(i)/t_i} \tag{14}$$

Step 3: Get the user private key sk, as shown in Equation (15).

$$sk = (D_i)_{i \in S} \tag{15}$$

3.4.3. Encryption

Step 1: Select a random number $s \in Z_r$ and calculate E' for the plaintext message $M \in G_T$, as shown in Equation (16).

$$E' = MY^s = Me(g, g)^{ys} \tag{16}$$

Step 2: For attribute i in the plaintext attribute set W, calculate E_i, as shown in Equation (17).

$$E_i = T_i^s \tag{17}$$

Step 3: The calculated ciphertext is ct, as shown in Equation (18).

$$ct = (E', E_i)_{i \in W} \tag{18}$$

3.4.4. Decryption

Step 1: If the number of overlapping attributes between the user attribute set S and the plaintext attribute set W is not less than d, then d is selected from the overlapping attributes to form the attribute set I.

Step 2: For each attribute i in I, calculate P_i, where $\delta_i(0)$ is the Lagrange factor, as shown in Equations (19) and (20).

$$P_i = e(E_i, D_i)^{\delta_i(0)} = e(g, g)^{sq(i)}\delta_i(0) \tag{19}$$

$$\Pi_{i\in I}P_i = e(g,g)^{s\Sigma_{i\in I}q(i)\delta_i(0)} = e(g,g)s^y \tag{20}$$

Step 3: Finally, the plaintext M is obtained through decryption, as shown in Equation (21).

$$M = E'/\Pi_{i\in I}P_i \tag{21}$$

The encryption algorithm based on Fuzzy-IBE is based on dual-linear mapping and Lagrangian coefficients, providing flexible and safe access control for users with specific attributes to effectively avoid privacy leakage in logistics transportation.

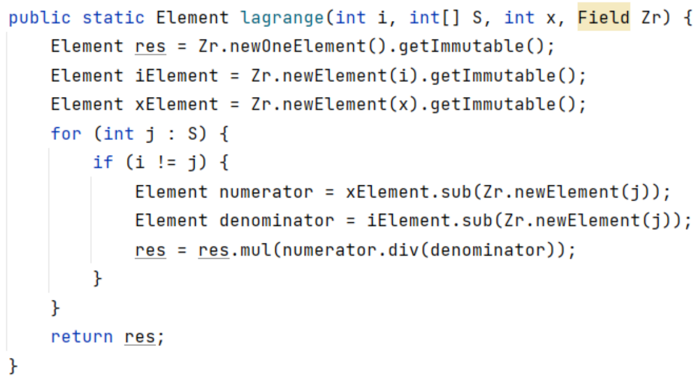
The sender encrypts the package information based on the receiver's identity attribute to ensure that only users with corresponding attributes can decrypt it. The package information is sent to the logistics company and uploaded to the terminal. In the preliminary stage, the logistics company generates the encryption key *pk* and the master key *msk* through Fuzzy-IBE to generate a regular replacement after a certain period. Logistics companies encrypt logistics information based on their identity attribute strategies and logistics information planning. The administrator and courier of the transportation company uploaded the attributes to the credible key generation agency to register to obtain the property's private key. The administrator allocates identity rights to employees based on their work content, which constitutes the account attributes. This mechanism effectively protects the security of logistics information in transmission and storage procedures and prevents access to unauthorized users.

After the package reaches the courier, the following operations are performed: a scan to obtain the package ID and an upload of the package ID with its corresponding identity attributes. Subsequently, the terminal searches the logistics information wrapped and associated with the encrypted encryption to determine whether the weight blending between the identity attribute and the attribute set was greater than the system door limit value. If the access strategy is met, the ciphertext of the logistics information is sent to the courier. The courier uses a private key *sk* to decrypt the logistics information. The courier decrypts the ciphertext to access the logistics information, including the shipping address of the sender and the receiver. After obtaining accurate logistics distribution information, the courier ensures delivery to the designated receiving place specified by the recipient. Similarly, the user scans the package ID, and the overlapping convergence of the completion of the identity attribute and the attributes in the terminal is greater than the system door limit value. After the two parties successfully complete their identity verification, the recipient can successfully receive the package.

3.5. Simulation Experiments

This experiment consists of two parts: implementing the Fuzzy-IBE algorithm and connecting the user wallet to the decentralized platform Arweave and storing data. Fuzzy-IBE uses the Java Pairing-Based Cryptography (JPBC) library, which is a pair-based encryption library. It provides a Java package for the Pairing-Based Cryptography (PBC) library and is often used to simulate pairing-based cryptography algorithms, such as identity-based encryption.

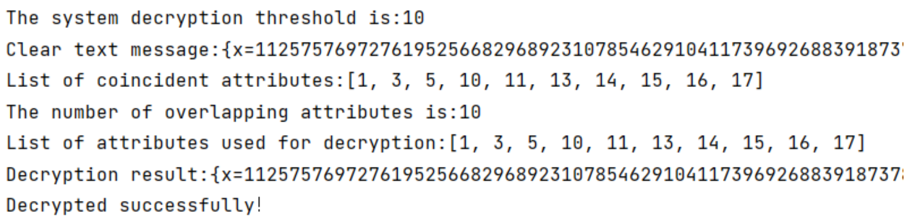
Lagrangian interpolation is one of the core steps of Fuzzy-IBE. It is a polynomial interpolation method. Nodal basis functions are provided at given nodes, and then these basis functions are linearly combined, where the combination coefficient is the node function value. The resulting polynomial interpolation function passes precisely through all the given data points. Lagrange factor calculation i is an element in the set S, and x is the target point value, as shown in Figure 5.



```java
public static Element lagrange(int i, int[] S, int x, Field Zr) {
    Element res = Zr.newOneElement().getImmutable();
    Element iElement = Zr.newElement(i).getImmutable();
    Element xElement = Zr.newElement(x).getImmutable();
    for (int j : S) {
        if (i != j) {
            Element numerator = xElement.sub(Zr.newElement(j));
            Element denominator = iElement.sub(Zr.newElement(j));
            res = res.mul(numerator.div(denominator));
        }
    }
    return res;
}
```

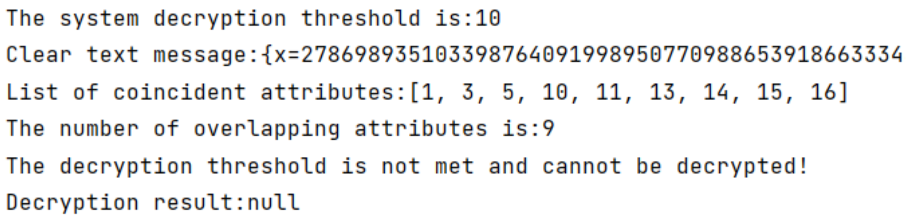
Figure 5. Computation part of Lagrange interpolation.

When the user decrypts the identity information and meets the set attribute list, the encrypted ciphertext will be decrypted, as shown in Figure 6. People who do not meet the conditions have not reached the decryption threshold and cannot decrypt the accurate identity information, thereby enhancing privacy protection, as shown in Figure 7.



```
The system decryption threshold is:10
Clear text message:{x=112575769727619525668296892310785462910411739692688391873
List of coincident attributes:[1, 3, 5, 10, 11, 13, 14, 15, 16, 17]
The number of overlapping attributes is:10
List of attributes used for decryption:[1, 3, 5, 10, 11, 13, 14, 15, 16, 17]
Decryption result:{x=1125757697276195256682968923107854629104117396926883918737
Decrypted successfully!
```

Figure 6. The ciphertext decryption results when the system threshold is met.



```
The system decryption threshold is:10
Clear text message:{x=278698935103398764091998950770988653918663334
List of coincident attributes:[1, 3, 5, 10, 11, 13, 14, 15, 16]
The number of overlapping attributes is:9
The decryption threshold is not met and cannot be decrypted!
Decryption result:null
```

Figure 7. The ciphertext decryption results when the system threshold is not met.

The time costs associated with testing Fuzzy-IBE setup, key generation, encryption, and decryption functions using different attribute nodes are illustrated in Figures 8 and 9. A detailed configuration is provided in Table 3. The simulation experiment is implemented in Java, with encrypted data size set to 128 bytes. During the attribute test, the total time of Fuzzy-IBE is about 1 second. Specifically, the logistics company management server requires less than 0.7 seconds for each Fuzzy-IBE encryption and decryption operation. Additionally, it takes less than 0.2 seconds for the logistics company to manage the server. In a general logistics scheme, less than ten attributes are used, which can meet the needs of the standard logistics and transportation process.

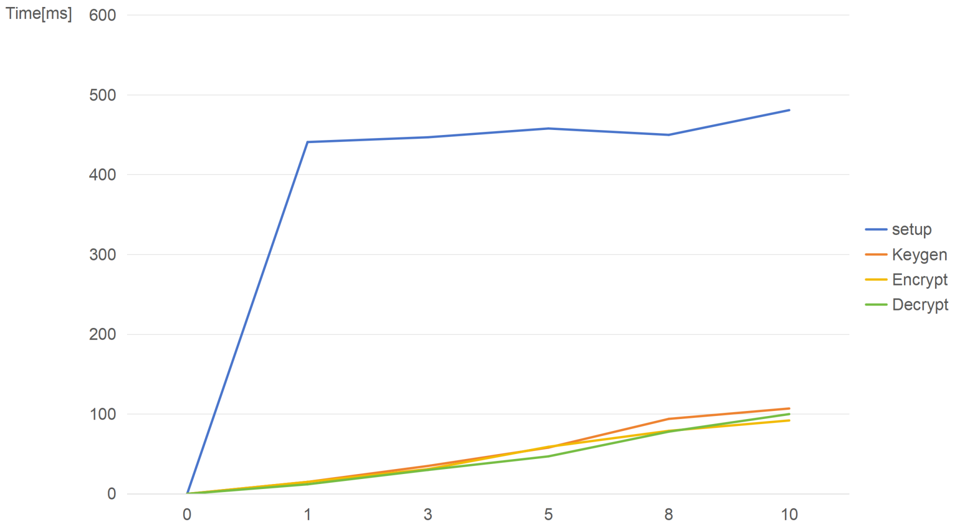


Figure 8. Fuzzy-IBE encryption algorithm overhead.

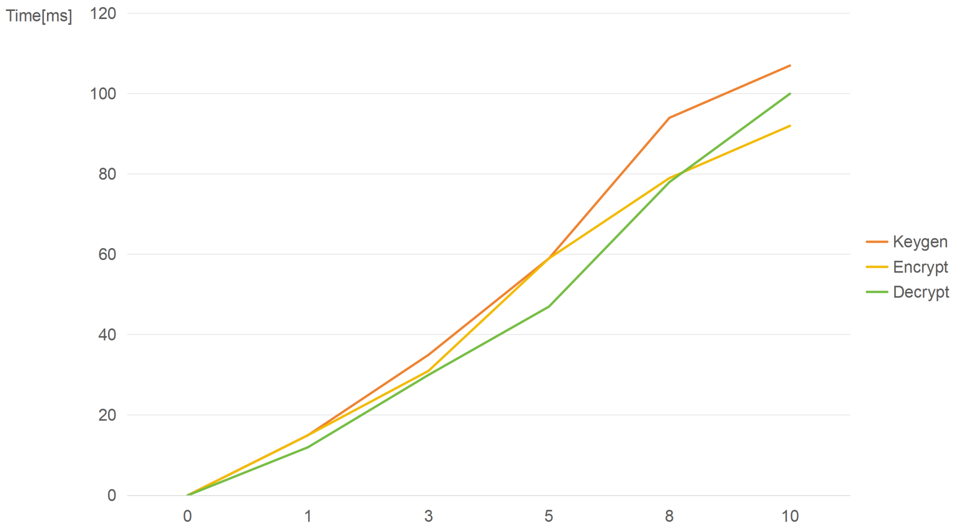


Figure 9. Fuzzy-IBE encryption algorithm overhead.

Compared with Fan et al.'s TraceChain study [33] and Hu et al.'s Test Decrypt Verify Attribute-Based Encryption study [34], the decryption time of this scheme is similar to Fan et al.'s, as shown in Figures 10 and 11. This means that when processing large amounts of logistics information, we can provide fast data decryption capabilities to ensure the real-time accuracy of logistics information. Compared to Hu et al.'s scheme, the advantage of decryption time in this scheme is crucial for scenarios that require rapid acquisition of logistics information. Moreover, the encryption time of this scheme is shorter than the other two schemes, which means that, in high-frequency scenarios of logistics information updates, we can complete data encryption more quickly and reduce the risk of information exposure during transmission and storage. As the number of attributes increases, the advantage of this scheme in decryption calculation becomes more apparent.

This scheme uses Arweave to implement decentralized storage. Arweave is a permanent storage network built on blockchain technology. It aims to address issues in traditional network storage methods, such as data loss, censorship, and centralized control. The economic model designed by Arweave allows storage costs to decrease over time, which means that data can be permanently stored on the network at a low one-time fee, effectively saving costs. To simulate the transaction, a local test chain was deployed to obtain a certain number of test AR coins, as shown in Figure 12.

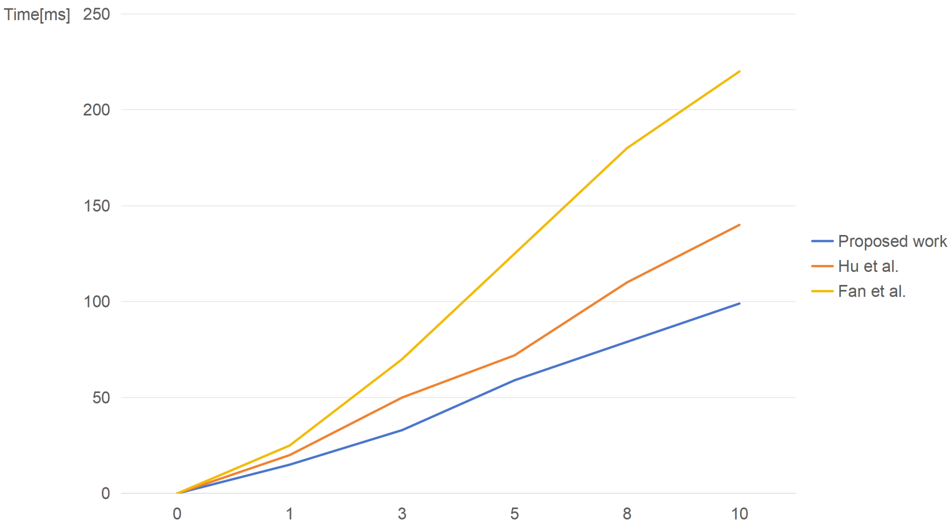


Figure 10. Comparison with other schemes in encryption.

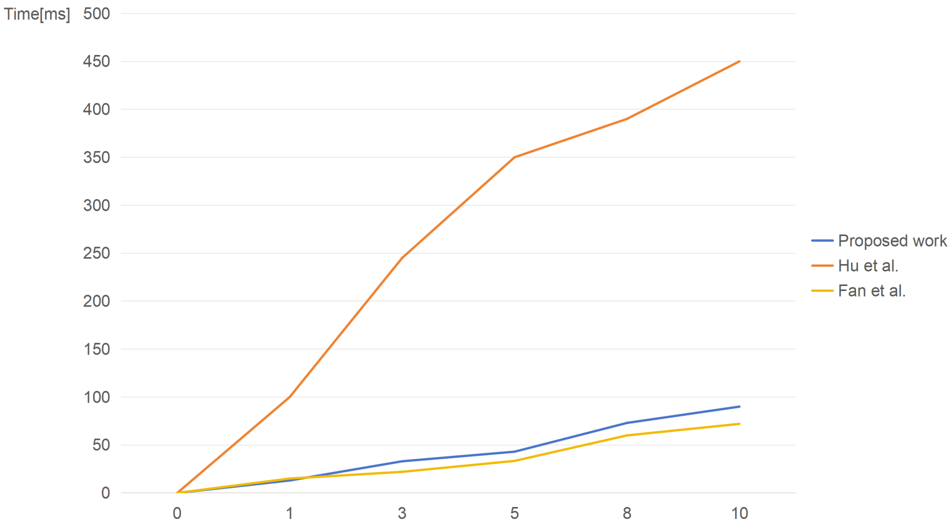
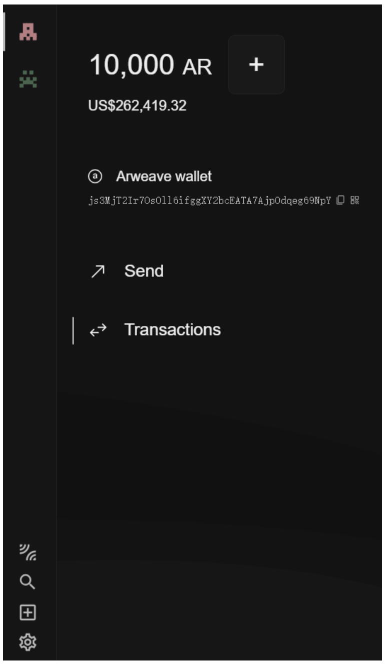


Figure 11. Comparison with other schemes in decryption.

Table 3. Software and hardware environment configuration.

Software and Hardware Environment	Configuration
CPU	2.90 GHz Intel Core i5-10400
RAM	16 GB DDR4-3200
System	Windows 10



10,000 AR

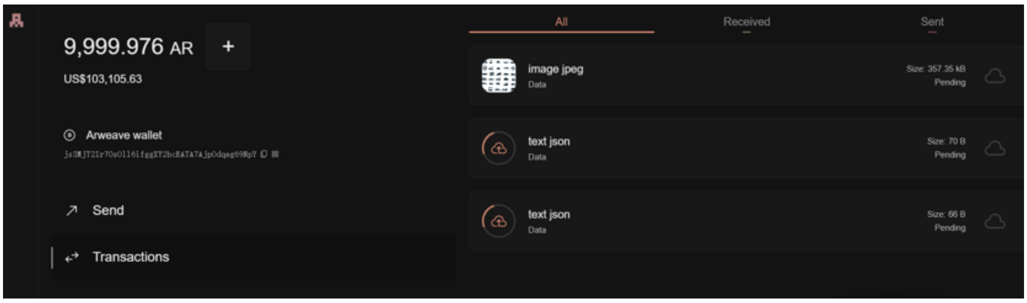
US$262,419.32

Arweave wallet

js3MjT2Ir7OsOll6ifggXY2bcEATA7AjpOdqeg69NpY

Figure 12. Deployment of the test chain locally to obtain test coins.

Smart contracts can use Arweave to store evidence or data related to contract execution, ensuring that the data will not be lost or tampered with, thereby enhancing the reliability and transparency of the contract. The scheme is integrated with Arweave, allowing users to upload data to the network and obtain a unique permanent link, as shown in Figure 13.



9,999.976 AR

US$103,105.63

Arweave wallet

js3MjT2Ir7OsOll6ifggXY2bcEATA7AjpOdqeg69NpY

image jpeg — Data — Size: 357.35 kB — Pending

text json — Data — Size: 70 B — Pending

text json — Data — Size: 66 B — Pending

Figure 13. Resources consumed after successful deployment.

4. Discussion

This paper explores the application of blockchain technology, data encryption, and access control to secure logistics transportation data. Blockchain's decentralized and immutable nature ensures data security and minimizes the risk of intrusion. The digital identity established on blockchain enables users to control their private data. Data encryption employs algorithms to render data unreadable, safeguarding them during transmission and storage. Access control guarantees that only authorized individuals can access sensitive logistics data.

In the context of the rapid development of information and digitization, the logistics and transportation industry has become an essential part of the modern economy. However, as the logistics and transportation industry progresses towards intelligence and networking, the concern regarding identity privacy protection is escalating, posing an urgent challenge that the industry must confront and resolve. Logistics transportation involves multiple parties, including cargo owners, carriers, logistics companies, etc. If there are deficiencies in the identity authentication mechanism, it may trigger risks such as identity forgery and information tampering, thereby posing a potential threat to the rights and interests of relevant parties. Furthermore, logistics companies are unaware of identity privacy protection and may arbitrarily disclose or abuse information from others. This violates the privacy rights of others and increases information security risks. The following paragraphs analyze this from two perspectives: privacy and security.

From a privacy perspective, Fuzzy-IBE protects personal privacy by encrypting and blurring identity information. The principle is desensitizing identity information so it cannot be directly identified. At the same time, Fuzzy-IBE can also add noise data, increase the uncertainty of identity information, and improve the degree of protection of individual privacy.

From a security perspective, one-time-use token technology is based on password-based security technology and uses a one-time password and token generation method for authentication. The principle is that each time the authentication is verified, the scheme will generate a new token and send it to the user. After the user uses the token to verify the identity, the token is invalid and cannot be used again. The generation process of a one-time-use token is random and unpredictable, so it has high security.

Some potential attacks and vulnerabilities remain against the Fuzzy-IBE encryption scheme, including the following: Collusion attacks, in which multiple malicious users collaborate, attempting to combine their attributes to decrypt information that they individually cannot decrypt. To mitigate this, the scheme can include a crucial management mechanism designed to reduce the risk of collusion. Identity forgery attacks, where attackers try to forge user identities for unauthorized communication. If attackers obtain sufficient identity information and construct similar identities, they may decrypt sensitive information. Improving identity verification accuracy through multi-factor authentication, such as biometric technology, can address this issue. Side channel attacks, in which attackers infer keys or sensitive data by analyzing side channel information, like power consumption changes during encryption operations. Techniques like shielding and filtering can be adopted to minimize side-channel information leakage. Simultaneously, an effective revocation mechanism is essential to ensure that only legitimate users can access the system.

In addition, in the design of logistics solutions, the issue of sustainability in the logistics transportation process should also be recognized. As Pečman et al. pointed out in their study [35], packaging waste management issues are caused by the surge in logistics transportation volume. Therefore, while pursuing logistics efficiency and privacy protection, it is necessary to establish an efficient recycling system. Considering the dynamic nature of transportation systems, the supply–demand ratio model proposed by Bartuska et al. provides an effective indicator for evaluating the state of transportation systems [36]. The plan can similarly define the privacy protection supply–demand ratio between the supply of privacy protection measures and the demand for privacy leakage risk. This indicator will help evaluate the sustainability of privacy protection schemes.

In summary, the privacy leakage problem caused by technical deficiencies and management loopholes in data security protection in the logistics and transportation industry is becoming increasingly prominent. Strengthening data encryption and storage security, improving identity authentication and access control mechanisms, and improving privacy protection and legal awareness can effectively solve these problems and ensure the safety and efficiency of logistics transportation. At the same time, enterprises and individuals should work together to form a synergy to promote the healthy and sustainable development of the logistics and transportation industry.

This scheme is compared with other privacy protection solutions based on several evaluation metrics, as shown in Table 4.

Table 4. Comparison of solutions.

	Zero-Knowledge Proof	**Off Chain Storage**	**Homomorphic Encryption**	**Differential Privacy**	**The Scheme**
Implementation principle	Miners prove legality	Off chain storage for transactions	Hide transaction amount with ciphertext	Adds noise to query results	Key exchange and identity policy
Management Strategies	Verification without leakage, zero knowledge disclosure	Secure decentralized storage	Manage parameters for homomorphic operations	Selection of noise parameters	Security fuzzy matching
Security Guarantees	Both parties do not need to disclose information to each other	Distributed encryption and verification	Guarantees privacy during computation	Balancing privacy and data availability	Selective disclosure
Cut off contact between both parties in the transaction	Yes	No	No	No	Yes
Is the account anonymous to the identity provider?	No	No	No	No	Yes

5. Applications and Analysis

In practical applications, customers must contact logistics and transportation companies in real time to understand the progress. After placing an order, customers can submit a query request to the company at any time and will receive a data packet containing a one-time address and encryption key. Logistics companies or e-commerce platforms encrypt critical information during shipment and transportation using Fuzzy-IBE and send it to the customer's one-time token account. Customers can use their private key to decrypt this information and stay informed of the progress of real-time shipping.

Logistics companies can provide a secure online customer service platform to establish temporary communication identifiers between customers and transportation companies, such as one-time chat IDs or session keys. Customers can contact transportation companies through this service without directly disclosing their contact information.

The scheme does not require the client to understand the operation of elliptic curves. These token accounts can be automatically generated and managed by the system, and clients only need to use these accounts without knowing the elliptic curve technology behind them. In practical applications, the operations related to elliptic curves are usually encapsulated in libraries or APIs. Users only need to interact through the interface or API provided by the system while the system internally processes encryption and decryption operations related to elliptic curves.

The plan aims to serve a wide range of users, including those without a cryptographic background. Similar to the elliptic curve mentioned above, in the design, complex cryptographic operations are encapsulated in the background and transparent to users. Users only need to complete secure logistics transactions through a simple interface operation without understanding the encryption mechanism behind it.

Although one-time-use tokens provide a high degree of anonymity, this anonymity poses challenges for law enforcement agencies in identifying illegal fund flows during the global regulatory efforts against money laundering (AML) and counter-terrorism financing (CFT). To address this, the plan proposes to regulatory agencies the establishment of a regulatory framework tailored to the characteristics of anonymous coins. This framework aims to ensure that regulatory requirements are met while protecting user privacy.

Additionally, generating one-time-use tokens involves complex computational processes, including hash functions, random number generation, and key pair management. This complexity may be daunting for ordinary users, increasing the barrier to entry. Therefore, as emphasized earlier, it is essential to encapsulate these intricate operations within the design to reduce the difficulty of user operations and enhance usability.

In practical applications, the distributed system architecture disperses data and computing tasks to multiple nodes, enhancing processing capabilities and response speeds. Distributed computing frameworks like MapReduce or Apache Hadoop facilitate the processing of numerous datasets across various nodes or clusters. This scheme effectively handles complex and large-scale data processing tasks, leveraging distributed computing technology to ensure scalability and high-performance data analysis. Additionally, load-balancing technology is employed to distribute the workload evenly among nodes, preventing single-point overloads. At the same time, implementing a cache mechanism often entails storing data or calculation results, thereby reducing the response time for subsequent requests. These scalability technologies empower the system to manage numerous users and data records, ensuring optimal performance, responsiveness, and resource utilization, even as workloads increase.

6. Conclusions

This paper proposes a privacy protection scheme for an SSI framework in logistics transportation and solves the problem of data leakage during logistics transportation. One-time-use tokens verify identity and enhance data security and user authentication within the logistics scheme. To ensure users' personal privacy and logistics route information, the integrated Fuzzy-IBE effectively hides the true identity, ensures authorized data access, and prevents unauthorized data leakage. By implementing enhanced data integrity and confidentiality measures, this scheme can ensure the security and protection of critical logistics data in various stages such as storage, transmission, and processing. Regulators are introduced to supervise the conduct of all relevant entities, thereby safeguarding the rights and interests of all stakeholders. This scheme establishes an environment of trust between users and entities such as logistics staff, thus promoting the development of the logistics field. Future work will focus on developing a sample project for demonstration purposes and aim to optimize the efficiency and other aspects of the scheme.

Author Contributions: N.S. was the advisor. C.Z. designed the scheme. C.Z. carried out the implementation. C.Z. wrote the manuscript. N.S. and Y.L. revised the final version of the text. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no grant or funding from any funding agency in the public, commercial, or not-for-profit sectors.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Chen, Y.; Zhang, H.; Wang, F.Y. Society-centered and DAO-powered sustainability in transportation 5.0: An intelligent vehicles perspective. *IEEE Trans. Intell. Veh.* **2023**, *8*, 2635–2638. [CrossRef]
2. Chung, S.H. Applications of smart technologies in logistics and transport: A review. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *153*, 102455. [CrossRef]
3. Xiaoguo, L.I.N.; Wang, X. PTLchain: Privacy and Traceability Enhanced Scheme for Logistics by using Consortium Blockchain. *J. Networking. Netw. Appl.* **2022**, *1*, 160–169.
4. Hunter, T. 2023 Data Breach Risk Annual Report; Shenzhen, China 2024. Available online: https://www.threathunter.cn/report (accessed on 30 June 2024)
5. Bartuska, L.; Hanzl, J.; Lizbetin, J. Urban traffic detectors data mining for determination of variations in traffic volumes. *Arch. Moto.* **2020**, *90*, 15–31. [CrossRef]
6. Nasution, A.A.; Erwin, K.; Bartuska, L. Determinant study of conventional transportation and online transportation. *Transp. Res. Procedia* **2020**, *44*, 276–282. [CrossRef]

7. Qian, W.E.I.; Xing-Yi, L.I. Express information protection application based on K-anonymity. *Appl. Res. Comput.* **2014**, *31*, 555.
8. Han, J.; Chen, L.; Schneider, S.; Treharne, H.; Wesemeyer, S.; Wilson, N. Anonymous single sign-on with proxy re-verification *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 223–236. [CrossRef]
9. Liu, Y.N.; Lv, S.Z.; Xie, M.; Chen, Z.B.; Wang, P. Dynamic anonymous identity authentication (DAIA) scheme for VANET. *Int. J. Commun. Syst.* **2019**, *32*, e3892. [CrossRef]
10. Bao, S.; Lei, A.; Cruickshank, H.; Sun, Z.; Asuquo, P.; Hathal, W. A Pseudonym Certificate Management Scheme Based on Blockchain for Internet of Vehicles. In Proceedings of the DASC/PiCom/CBDCom/CyberSciTech, Fukuoka, Japan, 5–8 August 2019; IEEE: New York, NY, USA, 2019; pp.28–35.
11. Zheng, D.; Jing, C.; Guo, R.; Gao, S.; Wang, L. A traceable blockchain-based access authentication system with privacy preservation in VANETs. *IEEE Access* **2019**, *7*, 117716–117726. [CrossRef]
12. Benarous, L.; Kadri, B.; Bouridane, A. Blockchain-based privacy-aware pseudonym management framework for vehicular networks. *Arabian J. Sci. Eng.* **2020**, *45*, 6033–6049. [CrossRef]
13. Moussaoui, D.; Kadri, B.; Feham, M.; Bensaber, B.A. A Distributed Blockchain Based PKI (BCPKI) architecture to enhance privacy in VANET. In Proceedings of the IHSH, Boumerdes, Algeria, 9–10 February 2021; IEEE: New York, NY, USA, 2021; pp. 75–79.
14. Garrido, G.M.; Sedlmeir, J.; Uludağ, Ö.; Alaoui, I.S.; Luckow, A.; Matthes, F. Revealing the landscape of privacy-enhancing technologies in the context of data markets for the IoT: A systematic literature review. *J. Network Comput. Appl.* **2022**, *207*, 103465. [CrossRef]
15. Da Silva, J.O.D.; dos Santos, D.R. Study of Blockchain Application in the Logistics Industry. *Theor. Econ. Lett.* **2022**, *12*, 321–342. [CrossRef]
16. Dorri, A.; Steger, M.; Kanhere, S.S.; Jurdak, R. Blockchain: A distributed solution to automotive security and privacy. *IEEE Commun. Mag.* **2017**, *12*, 119–125. [CrossRef]
17. Singh, M.; Kim, S. Branch based blockchain technology in intelligent vehicle. *Comput. Netw.* **2018**, *145*, 219–231. [CrossRef]
18. Yang, Y.T.; Chou, L.D.; Tseng, C.W.; Tseng, C.W.; Liu, C.C. Blockchain-based traffic event validation and trust verification for VANETs. *IEEE Access* **2019**, *7*, 30868–30877. [CrossRef]
19. Liu, X.; Huang, H.; Xiao, F.; Ma, Z. A blockchain-based trust management with conditional privacy-preserving announcement scheme for VANETs. *IEEE Internet Things J.* **2019**, *7*, 4101–4112. [CrossRef]
20. Mišić, J.; Mišić, V.B.; Chang, X. Scalable self-sovereign identity architecture. *IEEE Netw.* **2022**, *36*, 114–121. [CrossRef]
21. Mühle, A.; Grüner, A.; Gayvoronskaya, T.; Meinel, C. A survey on essential components of a self-sovereign identity. *Comput. Sci. Rev.* **2022**, *30*, 80–86. [CrossRef]
22. Xiao, M.; Ma, Z.; Li, T. Privacy-preserving and scalable data access control based on self-sovereign identity management in large-scale cloud storage. In Proceedings of the SpaCCS 2020, Nanjing, China, 18–20 December 2020; Springer International: Berlin/Heidelberg, Germany, 2020; pp. 1–18.
23. Schanzenbach, M.; Bramm, G.; Schütte, J. reclaimID: Secure, self-sovereign identities using name systems and attribute-based encryption. In Proceedings of the TrustCom/BigDataSE 2018, New York, NY, USA, 1–3 August 2018; IEEE: New York, NY, USA, 2018; pp. 946–957.
24. Soltani, R.; Nguyen, U.T.; An, A. Data capsule: A self-contained data model as an access policy enforcement strategy. In Proceedings of the BRAINS, Paris, France, 27–30 September 2021; IEEE: New York, NY, USA; pp. 93–96.
25. Stokkink, Q.; Pouwelse, J. Deployment of a blockchain-based self-sovereign identity. In Proceedings of the iThings.GreenCom.CPSCom. SmartData, Halifax, NS, Canada, 30 July–3 August 2018; IEEE: New York, NY, USA; pp. 1336–1342.
26. Rathee, D.; Policharla, G.V.; Xie, T.; Cottone, R.; Song, D. ZEBRA: Anonymous Credentials with Practical On-chain Verification and Applications to KYC in DeFi. *IACR Cryptol. ePrint Arch.* **2022**, *2022*, 1286.
27. Yadav, A.S.; Charles, V.; Pandey, D.K.; Gupta, S.; Gherman, T.; Kushwaha, D.S. Blockchain-based secure privacy-preserving vehicle accident and insurance registration. *Expert Syst. Appl.* **2023**, *230*, 120651. [CrossRef]
28. Karmakar, A.; Ghosh, P.; Banerjee, P.S.; De, D. ChainSure: Agent free insurance system using blockchain for healthcare 4.0. *Intell. Syst. Appl.* **2023**, *17*, 200177. [CrossRef]
29. Lux, Z.A.; Thatmann, D.; Zickau, S.; Beierle, F. Distributed-ledger-based authentication with decentralized identifiers and verifiable credentials. In Proceedings of the 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS), Paris, France, 28–30 September 2020; IEEE: New York, NY, USA; pp. 7–78.
30. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**, *21260*, 1–9. [CrossRef]
31. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997** 2, 1. Available online: https://firstmonday.org/ojs/index.php/fm/article/download/548/469 (accessed on 20 June 2024). [CrossRef]
32. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
33. Fan, Y.; Lin, X.; Liang, W.; Wang, J.; Tan, G.; Lei, X.; Jing, L. TraceChain: A blockchain-based scheme to protect data confidentiality and traceability. *Softw. Pract. Exper.* **2022**, *52*, 115–129. [CrossRef]
34. Hu, G.; Zhang, L.; Mu, Y.; Gao, X. An expressive "test-decrypt-verify" attribute-based encryption scheme with hidden policy for smart medical cloud. *IEEE Syst. J.* **2020**, *15*, 365–376. [CrossRef]

35. Pečman, J.; Vrábel, J.; Mašek, J.; Šedivý, J.; Stopková, M.; Bartuška, L. Packaging Waste Research Responding to the Rise of Transport and Logistics Volumes during the Covid-19 Pandemic. *Arch. Motoryz.* **2023**, *101*, 3. [CrossRef]
36. Bartuska, L.; Stopka, O.; Luptak, V.; Masek, J. Approach Draft to Evaluate the Transport System State—A Case Study Regarding the Estimation Ratio Model of Transport Supply and Demand. *Appl. Sci.* **2023**, *13*, 4638. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.