future internet

MDPI

Article

An Identity Privacy-Preserving Scheme against Insider Logistics Data Leakage Based on One-Time-Use Accounts

Nigang Sun [1,*], Chenyang Zhu [2,*], Yuanyi Zhang [3] and Yining Liu [4]

[1] School of Microelectronics and Control Engineering, Changzhou University, Changzhou 213000, China
[2] School of Computer Science and Artificial Intelligence, Changzhou University, Changzhou 213000, China
[3] Shanghai Shentie Information Engineering Co., Ltd. No.12, Huangchengdong Road, Shangcheng District, Hangzhou 310009, China; revanton@icloud.com
[4] School of Computer and Information Security, Guilin University of Electronic Technology, Guilin 541004, China; ynliu@guet.edu.cn
[*] Correspondence: ngsun@cczu.edu.cn (N.S.); s22150812067@smail.cczu.edu.cn (C.Z.)

Abstract: Digital transformation of the logistics industry triggered by the widespread use of Internet of Things (IoT) technology has prompted a significant revolution in logistics companies, further bringing huge dividends to society. However, the concurrent accelerated growth of logistics companies also significantly hinders the safeguarding of individual privacy. Digital identity has ascended to having the status of a prevalent privacy-protection solution, principally due to its efficacy in mitigating privacy compromises. However, the extant schemes fall short of addressing the issue of privacy breaches engendered by insider maleficence. This paper proposes an innovative identity privacy-preserving scheme aimed at addressing the quandary of internal data breaches. In this scheme, the identity provider furnishes one-time-use accounts for logistics users, thereby obviating the protracted retention of logistics data within the internal database. The scheme also employs ciphertext policy attribute-based encryption (CP-ABE) to encrypt address nodes, wherein the access privileges accorded to logistics companies are circumscribed. Therefore, internal logistics staff have to secure unequivocal authorization from users prior to accessing identity-specific data and privacy protection of user information is also concomitantly strengthened. Crucially, this scheme ameliorates internal privacy concerns, rendering it infeasible for internal interlopers to correlate the users' authentic identities with their digital wallets. Finally, the effectiveness and reliability of the scheme are demonstrated through simulation experiments and discussions of security.

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

check for updates

Citation: Sun, N.; Zhu, C.; Zhang, Y.; Liu, Y. An Identity Privacy-Preserving Scheme against Insider Logistics Data Leakage Based on One-Time-Use Accounts. *Future Internet* **2023**, *15*, 361. https://doi.org/10.3390/fi15110361

Academic Editor: Claude Chaudet

Received: 28 September 2023
Revised: 30 October 2023
Accepted: 3 November 2023
Published: 5 November 2023

1. Introduction

The digital transformation of the logistics industry, catalyzed by the pervasive application of IoT technology [1], has prompted swift evolutions within logistics companies, further conferring substantial benefits upon society [2]. However, concomitant with the development of the logistics industry, particularly in the realms of information privacy and security [3], escalating challenges pertaining to identity disclosure are encountered. For example, personnel within a logistics company exploited vulnerabilities in the organization's internal infrastructure to illicitly acquire and disseminate approximately 1.2 million units of civilian identity data, encompassing usernames, mobile numbers, domiciliary addresses, and other confidential particulars [4]. The reality is that logistics enterprises demonstrate inadequate data stewardship coupled with a glaring deficiency in identity management protocols and employees can easily obtain user data, which further exacerbates the risk of logistics privacy leaks. The proliferation of privacy infractions related to logistics identities has precipitated a myriad of proposed solutions to counter criminal activities engendered by these violations, which can be categorized into three predominant kinds: hiding critical information, decentralized storage, and digital identity. Hiding critical information is fundamentally a data desensitization technique, where sensitive data is substituted with arbitrary characters or symbols, thereby diminishing the risk of privacy breaches. Schemes [5–8] utilize hiding critical information, and the customer's personal and logistics information is encapsulated in the QR code. However, the main limitation of hiding critical information is that the data are still stored in plaintext in the company database, which intensifies the susceptibility to breaches of user identity privacy. Decentralized storage employs distributed computing technology to mitigate the propensity for dataset vulnerabilities. Conventional logistics architectures, characterized by centralized data repositories, are inherently susceptible to data manipulation and vulnerable to single-point incursions, as delineated in [9]. Decentralized storage paradigms attenuate the perils concomitant with data centralization inherent in conventional logistics frameworks. Some schemes [10–14] advocate for the decentralization of data storage to bolster the security of distributed networks. Nonetheless, the salient limitation of decentralized storage is that it entails a logistics firm superfluously accumulating identity particulars, which are subsequently warehoused in the database, thereby amplifying both the likelihood and detrimental impact of informational breaches. Digital identity directly instantiates an identity management system on the blockchain, which enables users to exercise personal control over and maintain the logistics account particulars. Blockchain and privacy protection solutions are applied in more fields. According to a recent study on blockchain technology for smart grid applications, blockchain can be used to improve the security and efficiency of smart systems, and it has the potential to optimize energy management systems [15]. The adoption of IoT technology can improve the efficiency of energy distribution and enable real-time monitoring [16]. In a report investigating technical approaches to address privacy concerns associated with connected vehicle systems, a novel measurement scheme was proposed for collecting aggregate OD flow data without compromising the privacy of the motorists' identity [17]. Moreover, digital identity obviates the requirement for users to divulge extraneous data particulars, thereby attenuating the risk of identity information leakage and circumventing the limitations inherent in the preceding two kinds of solutions. Consequently, digital identity is incrementally gaining widespread adoption within the domain of privacy safeguarding.

In the realm of digital identity, the INCOGNITO project endeavors to establish identity verification through privacy-preserving credentials, enabling anonymous access to online services [18]. Additionally, another report introduces a comprehensive architectural framework for a privacy-preserving biometrically secured electronic document system. This system could substantially enhance privacy protection by incorporating an additional layer of authentication based on an individual's distinctive physical characteristics [19]. Decentralized identity (DID) [20–23] is prevalently employed in privacy-preservation schemes owing to its conferment of autonomous data governance upon individuals. CanDID [24] represents the introduction of an identity system dedicated to the issuance and management of credentials, complemented by a key recovery mechanism. Kang et al. proposed a system [25] that integrated decentralized identity (DID) identifiers, DID communication, and verifiable credentials (VCS) for comprehensive identity management. Furthermore, the system incorporated the Cheon–Kim–Kim–Song (CKKS) fully homomorphic encryption (FHE) scheme for control calculations. With NYM credentials [26], anonymous authentication credentials are implemented, establishing an identity system that prioritizes user privacy. Luecking et al. [27] presented a distributed ledger technology-based framework for IoT device identity and trust, incorporating a Web of Trust (WOT) scheme to facilitate automatic trust ratings for diverse identities. Mohammadinejad et al. [28] proposed a decentralized personal data management system, guaranteeing that users retain control over their private data. Feng et al. proposed a protocol [29] that encrypts communication between parties, aiming to mitigate transactional privacy breaches on public blockchain networks. Zether [30] employed encrypted smart contracts to safeguard public privacy files using public and private key mechanisms. BE-RAN [31] offered user-centric identity management for user equipment (UE) and RAN units, ensuring mutual authentication across all entities. However, it can be concluded from the above analysis that there are two problems with the extant schemes: the first pertains to users' proclivity for executing recurrent transactions through a single account and the second involves an absence of stringent control over user data access. As a result, the extant schemes are ineffective in countering the dilemma of internal privacy breaches. Internal privacy issues mainly consist of insider maleficence, which refers to the threat posed by individuals within an organization who misuse their authorized access to the company's systems or information for malicious purposes. This threat involves logistics staff, such as warehouse employees or administrators, misusing their access to sensitive customer data or shipment details. Employees with varying levels of privilege, including IT administrators or network operators, might abuse their access rights to gain unauthorized entry into critical systems, compromising the confidentiality and integrity of sensitive logistics data.

This paper proposes a novel privacy-preserving scheme designed to obviate internal logistics data breaches through the utilization of one-time-use accounts and attribute-based encryption. Expanding upon the foundation of decentralized identity, this scheme particularly redresses the quandary of privacy breaches engendered by single accounts and access control deficiencies. In the inception phase of the scheme, the identity provider engenders stochastic, transient, one-time-use accounts that serve as proxies for the user's actual address. A one-time-use account constitutes a provisional virtual identifier that lapses into obsolescence post single utilization, thereby safeguarding against the prolonged storage or manipulation of logistical information. The scheme incorporates CP-ABE to safeguard account information by typifying access permissions predicated on specific attribute conditions. Consequently, this scheme ensures that internal logistics staff are confined to accessing only the data for which they have been authorized, thereby precluding them from the unauthorized acquisition of other users' sensitive particulars. This scheme guarantees that the authentic identity of the user cannot be associated with the corresponding account which affords robust safeguards for privacy. The scheme additionally embeds a supervisory authority tasked with overseeing the conduct of all participating entities, thereby forestalling both contentious disputes and the malfeasant utilization of accounts. In conclusion, this scheme is equipped to withstand logistics privacy infringements instigated by internal incursions and enhance oversight mechanisms to deter malfeasant conduct among all participating entities.

2. Preliminaries

2.1. Blockchain

The concept of blockchain was first introduced in the paper "Bitcoin: A Peer-to-Peer Electronic Cash System" [32]. Blockchain technology fundamentally represents a decentralized, distributed ledger database technology. It comprised a sequence of data blocks, each encrypted and containing information about transactions on the Bitcoin network, thereby forming a collectively maintained, reliable database. The blockchain is characterized by its anonymity, immutability, and decentralization. Given its unique characteristics, blockchain technology holds significant potential for a broad spectrum of applications, including digital currencies, supply chain management [33], and the Internet of Things (IoT).

2.2. Smart Contracts

Nick Szabo first proposed smart contracts in 1994 [34]. Smart contracts [35] serve as computer protocols designed to expedite the digital dissemination, authentication, or enforcement of contractual agreements. A smart contract is a digitally inscribed contract within a blockchain, characterized by its openness, transparency, organization, and resistance to tampering. The maturation of blockchain technology affords an expansive developmental landscape for smart contracts.

2.3. Decentralized Identity (DID)

Decentralized Identity (DID) constitutes an innovative modality of identity that resides wholly within the purview of the owner. This implies that all information pertaining to the identity is solely in the user's possession and no one can access this information without explicit permission.

DIDs can be directly registered on the blockchain or a distributed network, thereby providing effective privacy and security protection for users. A typical DID system encompasses four entities: issuer, inspector–verifier, holder, and identifier registry.

Issuers are entities tasked with managing and releasing user identity information. They substantiate a user's identity details by generating verifiable statements, and employ technologies like digital signatures to uphold the statement's authenticity and integrity. Inspector–verifiers represent an entity that accepts and verifies verifiable claims. They provide some type of service, such as verifying claims when requested by users. Holders are entities that request, receive, and hold verifiable claims from issuers. They can store issued claims in a verifiable claims wallet for future use. Identifier registries manage databases of decentralized identifiers (DIDs), enabling verifiers to authenticate users and claims.

2.4. One-Time-Use Accounts

One-time-use accounts are implemented based on one-time addresses. One-time addresses work on the principle that the user has a unique public–private key pair. The points on the elliptic curve have homomorphic additivity, and a random noise term is added to the key pair to obtain a one-time public and private key for real transactions. In the context of a one-time-use account, each initiation of a transaction prompts the sender to stochastically generate a provisional public key, predicated on the recipient's address, for the purpose of transaction reception. In the Unspent Transaction Output (UTXO) model, each UTXO transaction typically carries a tag that identifies the owner of the UTXO, represented by an address. The recipient who has the private key corresponding to the temporary public key can use the temporary private key to consume the UTXO in the future.

The user randomly generates the private key k and public key $K = kG$, where G is the elliptic curve parameter. Since the points on the elliptic curve have homomorphic additivity, a random noise term is added to the key pair to obtain a one-time public and private key for real transactions.

The recipient generates a public–private key pair k, K. The long-term public key K is published in the outside world. The sender selects a random number r when sending a transaction and sets the one-time public key $K' = rG + K$. The transaction is sent to K' while passing the random number r to the recipient.

To transfer r to the recipient in a private manner, the Diffie–Hellman secret exchange protocol is used. The sender uses the public key K disclosed by the recipient to hide r. The hidden result is passed to the recipient; other users only see R and cannot obtain the hidden r. The Hash function H maps points on an elliptic curve to scalars. The sender selects a random number r, calculates $K' = H(rK)G + K, R = rG$, and sends R publicly to the recipient. The recipient receives R, calculates $K' = H(kR)G + K$, and compares the R and K obtained. If they are consistent, it means that the one-time address belongs to the recipient and the one-time private key is recovered $k' = H(kR)G + k$.

2.5. CP-ABE

John Bethencourt et al. [36] first proposed attribute-based encryption (ABE). ABE facilitates the expression and enforcement of intricate access control policies in a manner that is both straightforward and adaptable, aiming to ensure data privacy protection. ciphertext policy attribute-based encryption (CP-ABE) [37] is a specialized form of ABE. It serves as an attribute-centric access control mechanism that encrypts and safeguards data by linking access rights with specific attribute conditions.

In CP-ABE, the roles of individual participants are demarcated by attribute designations, thereby constructing an access architecture that delineates the authorized constellation of attributes. A set within the access structure is deemed authorized, while a set outside the access structure is considered unauthorized. The access policy is embedded within the ciphertext data and the user's private key is associated with a set of attributes. A user can decrypt the ciphertext only if the attributes within the user's private key satisfy the access policy of the encrypted data. Therefore, through the employment of the CP-ABE methodology, the data proprietor is empowered to stipulate access permissions without necessitating intimate knowledge of the specific identities of all authorized entities.

2.6. *Diffie–Hellman Secret Exchange Protocol*

In the formula that the protocol depends on, $(a * G)b = (b * G) * a$, where a and b are constants. The sender and the receiver negotiate the parameters of the elliptic curve and the base point G. These parameters are all public.

The sender generates a random key pair (public key: $K_a = k_a G$, private key: k_a). The receiver generates a random key pair (public key: $K_b = k_b G$, private key: k_b). The sender multiplies the recipient's public key by his own private key to obtain the shared key $k_s = (k_b G)k_a$. Similarly, the recipient multiplies the sender's public key by his own private key to obtain the shared key $k_s = (k_a G)k_b$. According to the formula mentioned earlier, $(a * G)b = (b * G) * a$, the shared keys obtained by the sender and the receiver are equal. In this way, the two parties complete the key exchange through the protocol.

3. Scheme Design

Presented in this section are the implementation details of the scheme. Section 3.1 is the comprehensive architecture of the proposed scheme. Section 3.2 is the overall process of this scheme, including the generation of one-time-use accounts, identity verification, and identity supervision. Section 3.3 is a description of the flow of smart contract deployment transactions in this scheme. Section 3.4 contains the related algorithms for attribute encryption to implement access control. Section 3.5 details the experiments implemented on attribute encryption and identity account connection wallet.

3.1. *Scheme Architecture*

The scheme includes senders, receivers, identity providers, logistics companies, distribution centers, and regulators. Identity providers are responsible for managing users' real identities and their anonymous accounts. Logistics companies are responsible for verifying the signatures of all parties involved and assigning routes for the delivery of goods. Regulators include some government departments and criminal investigation agencies. The distribution center implements the cargo transportation plan. The ORDER contract is deployed on the blockchain network and invoked during transactional processes. The architecture of the scheme is shown in Figure 1.

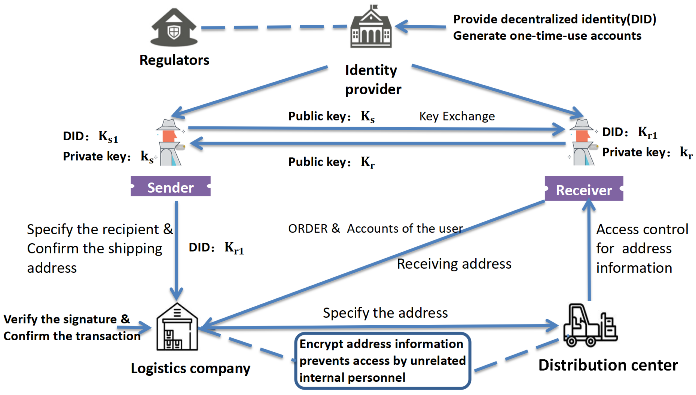

Regulators
Provide decentralized identity(DID)
Generate one-time-use accounts
Identity provider
Public key: K_s
Key Exchange
DID: K_{s1}
Private key: k_s
DID: K_{r1}
Private key: k_r
Public key: K_r
Sender
Receiver
Specify the recipient & Confirm the shipping address
DID: K_{r1}
ORDER & Accounts of the user
Receiving address
Access control for address information
Verify the signature & Confirm the transaction
Specify the address
Encrypt address information prevents access by unrelated internal personnel
Logistics company
Distribution center

Figure 1. Scheme architecture.

3.2. Scheme Process

Described in this section is the overall process of the identity scheme.

3.2.1. Identity Registration: One-Time-Use Accounts

In this scheme, in order to resist internal intrusions and improve identity privacy, all users' orders, payments, forwarding, and other services are completed through one-time-use accounts on the chain. The identity provider will first generate the one-time-use account for the user.

A one-time-use account consists of a one-time public and private key. The concept of one-time public keys, as mentioned in Monero [38], is used to generate a decentralized identity (DID) that controls access and hides the association between addresses and real identities. G represents a public parameter that signifies a base point of an elliptic curve and $Hash$ is a function that encodes points on an elliptic curve into a scalar, as shown in Table 1. The initial steps are dedicated to generating a one-time-use account. This computation can be performed by a third-party peer-to-peer tool or managed by the platform, which only provides the computation and transmission of these keys. The storage is typically generated locally or temporarily, and does not record any information related to the user's identity so this will protect identity privacy.

Table 1. Symbol descriptions.

Notation	Descriptive
G	Parameters on elliptic curves
r_r	Random numbers on elliptic curves
R_r	The random number after multiplication with the common parameter G
$Hash$	Plaintext is mapped to a shorter binary string
K_r,K_s	Receiver and sender public keys
k_r,k_s	Receiver and sender's private keys
K_{r1}, K_{s1}	One-time public key from hash calculation
sk_r,sk_s	Private key from the hash calculation

The receiver and sender randomly generate private keys, denoted as k_r, k_s. Then, the receiver and sender disclose public keys to each other, denoted as $K_r = k_r G, K_s = k_s G$.

The receiver generates a random r_r to calculate R_r and sends R_r to the sender, as shown in Equation (1).

$$R_r = r_r G \tag{1}$$

After receiving R_r, the sender generates a random number r_s and calculates the receiver's one-time public key K_{r1} which is shown in Equation (2).

$$K_{r1} = Hash(r_s K_r)G + K_r \tag{2}$$

The sender calculates the one-time public key K_{s1} and obtains the K_{s1} as DID, which is shown in Equation (3). Only the sender has the private key sk_s, which is shown in Equation (4).

$$K_{s1} = Hash(k_s R_r)G + K_s \tag{3}$$

$$sk_s = Hash(k_s R_r) + k_s \tag{4}$$

The sender utilizes K_{s1} to authenticate to the logistics company that initiates the order and uses the private key to digitally sign the intended shipping address denoted as address-from. The sender specifies the receive as K_{r1} and then computes R_s and sends it to the receiver, as shown in Equation (5).

$$R_s = r_s G \tag{5}$$

$$K'_{r1} = Hash(k_r R_s)G + K_r = K_{r1} \tag{6}$$

Then, the receiver obtains R_s and calculates K'_{r1}, as shown in Equation (6). The receiver's DID is set to K_{r1}. The receiver has its private key sk_r, as shown in Equation (7). Therefore, the receiver uses K_{r1} to log into the logistics company and uses the private key to sign the recipient's address denoted as address-to.

$$sk_r = Hash(k_r R_s) + k_r \tag{7}$$

The logistics company verifies the signatures of the sender and of the receiver, calculates the amount according to the address, and initiates the online collection. In addition, the distribution center can query address information. The distribution center point can, however, only query the next station address and they cannot obtain the complete logistics link. K_r is used to encrypt the information in the first and middle parts of the entire logistics, and K_s is used to encrypt the information in the middle and back parts of the logistics. Individuals in possession of the private keys are empowered to decrypt and scrutinize the status of logistics.

To reduce malicious behavior, every change to the sending or receiving address must be signed using the private keys of both parties. Therefore, employing one-time-use accounts for transactions can avoid privacy issues caused by repeated transactions with a single account.

The one-time-use account generated is securely associated with the user's identity through digital signatures or encrypted tokens. To ensure the validity of the one-time-use account, the system can employ several mechanisms, including time-based validation methods, digital signatures, or asymmetric key verification. These techniques play a crucial role in guaranteeing that the account remains valid for a specified duration or until the intended purpose is accomplished. Once the predetermined period expires or the purpose is fulfilled, the account is either invalidated or rendered unusable.

Users maintain only one account, simplifying management, and the process of generating DID for each transaction is concealed from the user through the client's access to the logistics company's interface. When transferring funds to another person's public address, the funds are actually dispatched to a randomly generated, one-time destination address known as the stealth address, as shown in Figure 2. This one-time address is instantly generated for the transaction and is public. However, the sender's public key and the recipient's public key remain private, and the sender's public address does not appear in public transaction records. Only the sender and receiver are aware of the concealed address of the order funds. Recipients who wish to inquire about funds need to possess a view private key to scan the blockchain to verify whether the funds have arrived. Only the recipients of the funds know their viewing private keys. Importantly, these addresses are not associated with the sender's or recipient's real identity to reduce identity leaks and prevent insider intrusions.

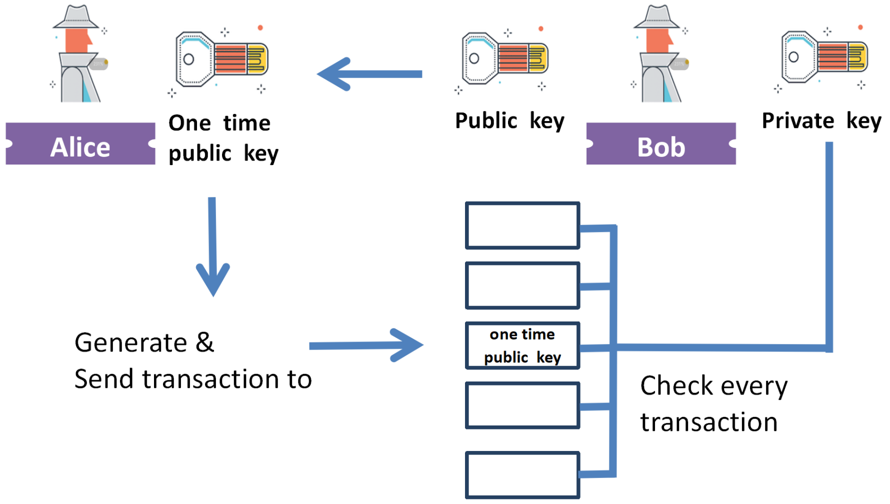


Figure 2. The transaction process of the one-time public key.

3.2.2. Identity Verification

When the user proves to the logistics company that the wallet account is bound to a verified identity, they need to provide proof of eligibility for the wallet account, authorized blockchain scope, and identity ID. The logistics company will verify and check whether the identity is expired or locked. If the verification is passed, the wallet account has then been authenticated and authorized. The user can then access the service and call the ORDER contract through this account to complete the transaction. Transaction data is logged so that it can be provided to regulators if required. The logistics company can set the login validity period of wallet accounts. After passing the verification, the user can directly access the service within the validity period without repeated verification.

3.2.3. Identity Supervision

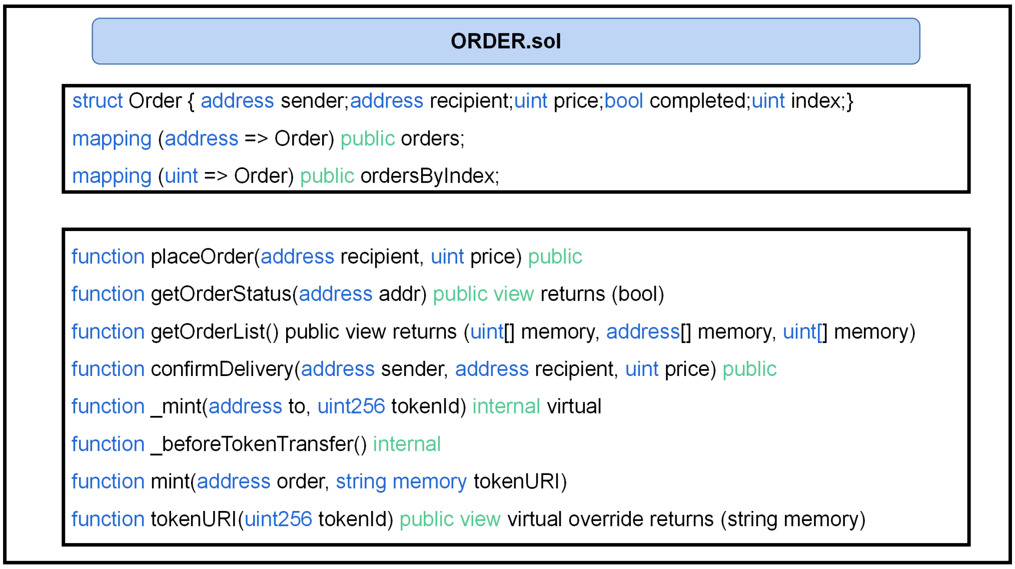
The purpose of supervision is to be able to correlate identities to obtain evidence when crimes and disputes occur. Regulators utilize data monitoring and analysis tools to collect and analyze large volumes of data generated within the regulated systems. These tools often include advanced analytics and data visualization capabilities that enable regulators to identify patterns, anomalies, and potential risks within the system.

Compliance management systems aid regulators in monitoring and overseeing regulatory compliance activities within the system. These systems often include features for documenting regulatory requirements, conducting audits, and monitoring adherence to compliance standards, ensuring that the system operates within the prescribed regulatory framework.

Finally, regulators request access to identity information held by identity providers and wallet accounts held by users in order to promptly detect and stop violations. When a transaction dispute occurs, the regulators can investigate the logs, and make rulings to safeguard the legitimate rights and interests of all involved parties.

3.3. Smart Contracts

Smart contracts play the role of automatic execution, security, and traceability in decentralized identity in addition to providing strong support for the construction of a decentralized identity system. Users authenticate through the account furnished by the identity provider and invoke the smart contract to complete the logistical transaction. The ORDER contract is designed to implement the core business logic of order transactions, including functions such as initiating the transaction order and querying. The contract is shown in Figure 3.



ORDER.sol

```
struct Order { address sender;address recipient;uint price;bool completed;uint index;}
mapping (address => Order) public orders;
mapping (uint => Order) public ordersByIndex;
```

```
function placeOrder(address recipient, uint price) public
function getOrderStatus(address addr) public view returns (bool)
function getOrderList() public view returns (uint[] memory, address[] memory, uint[] memory)
function confirmDelivery(address sender, address recipient, uint price) public
function _mint(address to, uint256 tokenId) internal virtual
function _beforeTokenTransfer() internal
function mint(address order, string memory tokenURI)
function tokenURI(uint256 tokenId) public view virtual override returns (string memory)
```

Figure 3. ORDER contract.

The mappings are used to record the information related to order transactions and record the corresponding relationship between orders and indexes. The struct Order includes elements such as the sender, recipient, price, order status, and order index for querying. The associative array contains the order array and the index array.

Algorithm 1 outlines the function for submitting an order, which verifies whether the sender has sufficient balance. It then creates a new order and adds the order to the index array. When submitting a new order, the recipient's address and order price must be entered. This function will add the new order to both the order array and the index array.

Algorithm 1 placeOrder( ) public

```
Input:  address recipient,
        address sender,
        unit price,
        unit index
  1.Require(msg.sender.balance>=price)
  2.Orders[msg.sender]=Order
  3.ordersByIndex[orders.length]
  4.Msg.sender.transfer(price)
```

Algorithm 2 outlines a function to check the status of an order. By inputting the sender's address, the function returns whether the order is complete or not.

Algorithm 2 getOrderStatus() public view

```
Input:  address addr
  1.return orders[].completed;
```

Algorithm 3 is user-invoked and outlines a function to query the list of orders. It creates an empty array for storing order information and adds the order information to this array. It then returns the array, which contains information about the order type, sender, and order price.

Algorithm 3 getOrderList() public view returns

```
Input:  uint[] memory,
        address[] memory,
        uint[] memory
  uint[] memory ordertypes = new uint[](orders.length);
  address[] memory senders = new address[]();
  uint[] memory prices = new uint[]();
```

Algorithm 4 defines a function to confirm receipt and checks whether the order sender's address is correct. The sender address, receiver address, and order price are input. After confirming receipt, the function updates the address and index.

Algorithm 4 confirmDelivery() public

```
Input:  address sender,
        address recipient,
        uint price
  require(orders[].recipient == recipient)
  orders[sender].completed = true;
  ordersByIndex[orders[].index].completed = true;
```

Algorithm 5 details the 'mint' function, which takes as inputs the order address and a 'tokenURI' string. The address is designated for receiving funds and, within the function, a check is performed to ensure that the order address is not a zero address. The 'tokenURI' is utilized to identify a specific token, allowing it to be referenced within the contract for both processing and manipulation.

Algorithm 5 mint() public

```
Input:  address order,
        string tokenURI
  uint256 newItemId = tokenIds.current();
  mint(order, newItemId);
  setTokenURI(newItemId, tokenURI);
  tokenIds.increment();
  return newItemId;
```

3.4. Attribute Encryption of Address Information

Access control serves as a vital security mechanism in logistics transportation, safeguarding logistics information from unauthorized access by illegitimate users as well as establishing specific access rights so that only authorized users can access the information. Traditional access control schemes, primarily reliant on user identification, often suffer from inflexibility, coarse granularity, and inadequate security due to their static allocation of rights.

The scheme employs ciphertext policy attribute-based encryption (CP-ABE) to encrypt the address to hide the sender and receiver information, which can allow data leakage caused by internal intrusion to be resisted and thus improve identity privacy protection.

The attribute structure encompasses location attributes, time attributes, and access attributes. In the encryption process of logistics information, the geographical location coordinates serve as the location attribute, the courier's regular working hours serve as the time attribute, and the courier's authorized identity serves as the access attribute. Consequently, this system guarantees that only couriers situated in the correct geographical location during the designated working hours and possessing the appropriate access rights can acquire the decryption key for decrypting the information.

This encryption algorithm refines access granularity to the attribute level, formulating access control policies based on the attributes of the authorized group. By integrating these attributes into the user's private key, the shared data is encrypted into ciphertext. This algorithm incorporates authorization group attributes into shared data and private keys. Therefore, it can be ensured that only users whose attributes meet the requirements of the access control policy can decrypt the ciphertext and access the shared data, as shown in Algorithm 6.

Algorithm 6 Encryption algorithm

```
Input: Node[] accessTree,plaintext,userAttList
Output: Node[] nodes,delivery result
  Initialization                                                  ▷ generate parameters
  Generate swarm elements $g$
  Generate random number $\alpha$ and $\beta$
  Keygen                                                          ▷ calculate the private key
  mskProp←$g^{\alpha}$
  pkProp←$(g, g^{\beta}, e(gg)^{\alpha})$
  Generate random elements $t$
  skProp←$(g^{\alpha}g^{\beta t}, g^{t})$
  for userAttList do
      Each attribute i of attList in the user attribute list
      if $D_i = H(i)^t$ then
          skProp←$(D, D_0, D_i)$
      end if
  end for
  Encrypt ← Node[] accessTree                                     ▷ encrypted plaintext
  Generate random number $s$                                      ▷ $s$ is the shared secret value
  Calculate the ciphertext component C
  Plaintext M ∈ G
  C = $Me(g,g)^{\alpha s}$
  $C_0$ = $g^{s}$
  Each node gets the corresponding secret slice
  for Node[] accessTree do
      if is Leafnode then
          Choose a random number $r$
          Calculate $C_i^1$,$C_i^2$
          ciphertext $ct$=$(C, C_0(C_i^1,C_i^2))$
      end if
  end for
  Decrypt ← Node[] accessTree  ▷ Only when the key attribute set satisfies the ciphertext
  access tree can it be decrypted
  Load ciphertext, private key
  for Node[] accessTree do
      Overlapping attributes in the key attribute set $S$ and the leaf node $T$ set, calculate $P_i$
      if Recovery succeeded then
          calculate $e(gg)^{\alpha s}$
          Plaintext M ← C/e(gg)$^{\alpha s}$
      end if
  end for
```

3.4.1. Initialization

Generate the pairing-related public parameters (e, g, G_1, G_T, Z_r), as shown in Table 2. Then, choose the parameter λ to determine a bilinear pair corresponding to it. Choose random numbers $\alpha \in Z_r$ and $\beta \in Z_r$, and compute $Y = e(g,g)^{\alpha}, g^{\beta}$. Therefore, the master key(MK) is g^a and the public key is pk, as shown in Equation (8).

$$pk = (Y, g^{\beta}) \tag{8}$$

Choose a random number t and compute $D = g^{\alpha}g^{t\beta}$, $D_0 = g^t$. For attribute i in the user list, compute $D_i = H(i)^t$. Then, generate the user's private key sk, as shown in Equation (9).

$$sk = [D, D_0, (D_i)] \tag{9}$$

Each attribute authority manages one or more classes of attributes, and the attributes managed by different attribute authorities have intersecting parts.

Table 2. Symbol descriptions.

Notation	Descriptive
C, C_0, C_i^1, C_i^2	Plaintext encryption
D, D_0, D_i	Random number calculation private key
G_1, G_T	Random group element
α, β	Random integers
e	The public parameters
g	The public parameters
Z_r	Set of integers
Y	Random number calculation public key
pk	The public key
sk	The private key
t	Random number
M	Plaintext message
H	Plaintext is mapped to a shorter binary string
ct	Standardized ciphertext
S	Secret key attribute collection
T	Standardized access control tree
i	Leaf node attributes
λ_i	Secret slice corresponding to i

3.4.2. Encryption

In order to encrypt the plaintext $M \in G_T$, choose a random number $s \in Z_r$ and compute C = $Me(g,g)^{\alpha s}$, $C_0 = g^s$. Split s along the access tree as a secret. Suppose the secret slice assigned to each leaf node with attribute i is λ_i; then, compute C_i^1, C_i^2, as shown in Equation (10). The ciphertext is ct, as shown in Equation (11).

$$C_i^1 = g^{\lambda_i \beta} H(i)^{-r_i}, C_i^2 = g^{r_i} \tag{10}$$

$$ct = (C, C_0, (C_i^1, C_i^2)) \tag{11}$$

3.4.3. Decryption

The secret key can only be decrypted if the set of attributes S of the secret key is sufficient to satisfy the secret access tree T. Suppose the overlapping attribute among the attributes of the leaf nodes of T and the attributes of the S set is i; then, compute P_i, as shown in Equation (12).

$$P_i = e(C_i^1, D_0)e(C_i^2, D_i) = e(g,g)^{t\lambda_i\beta} \tag{12}$$

Starting from the root node, perform the recursive calculation. Eventually, the secret value of the root node can be recovered in the form of $e(g,g)^{\beta ts}$.

$$e(C_0, D) = e(g,g)^{as}e(g,g)^{\beta ts} \tag{13}$$

In order to get the ciphertext, calculate $e(C_0, D)$, as shown in Equation (13). Then, further derive $e(g, g)^{as}$, as shown in Equation (14).

$$e(g, g)^{as} = e(C_0, D)/e(g, g)^{\beta ts} \quad (14)$$

Finally, the plaintext M is obtained, as shown in Equation (15).

$$M = C/e(g, g)^{as} \quad (15)$$

The algorithm is based on the secret sharing matrix access structure and each piece of user data corresponds to an access control policy. Consequently, visitors possessing disparate attributes are accorded divergent access privileges to data, resulting in variances in the scope of data information to which they are granted access. Unauthorized logistics insiders cannot further access the information, thereby preventing unauthorized personnel from accessing the information and avoiding leakage of internal information.

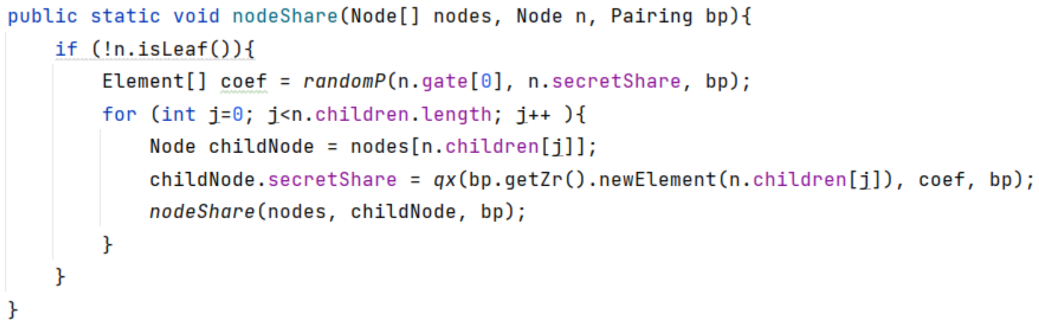
In the preliminary phase, the logistics company generates the CP-ABE encryption public key *Pk* and master key *Mk*, which are periodically replaced after a certain duration of use. Logistics companies employ various attribute strategies to encrypt the attributes of these segments, based on the planned delivery route for the logistics information. Initially, the logistics company utilizes the private key to decipher the message transmitted by the sender, which contains the user's address information. Subsequently, a viable delivery route is planned, considering the company's express transportation routes and transfer station conditions together with the logistics information encrypted using key encapsulation based on ciphertext policy attributes.

Upon the package reaching the courier, the following operations are conducted: scanning to acquire the package and uploading the package alongside its corresponding attribute set. Subsequently, the terminal searches for the package and associates the encrypted logistics information to ascertain whether the attribute set uploaded by the courier aligns with the access tree T. If the access policy is met, the ciphertext CT of the logistics information is sent to the courier. The courier uses the private key *sk* to decrypt the plaintext M of the logistics information. The courier decrypts the ciphertext, thereby accessing the logistics information, which includes the phone numbers and precise logistics addresses of both the sender and the recipient. With this information at hand, the courier is able to perform precise and accurate logistics delivery. Following the acquisition of accurate logistics delivery information, the courier ensures the express delivery reaches the designated receiving location specified by the recipient. Upon the successful completion of authentication by both parties, the recipient can then successfully retrieve the package.

3.5. Simulation Experiments

The experiment consists of two parts: implementing the attribute encryption algorithm and smart contract to connect the user's account wallet. First, the Java Pairing-Based Cryptography Library (JPBC) is used in the attribute encryption algorithm. JPBC is a Java encapsulation of the Pairing-Based Cryptography Library (PBC), which is usually used for algorithm simulation of attribute-based encryption.

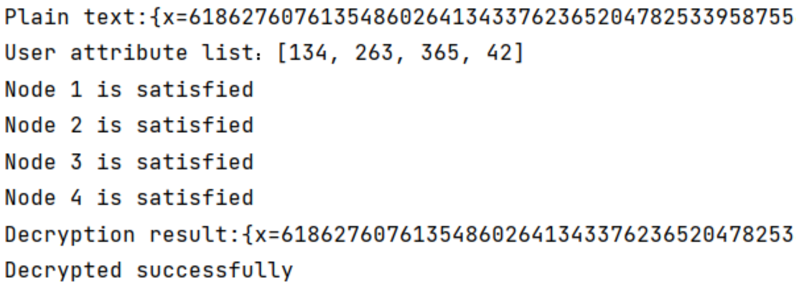
Secret sharing is one of the core steps of attribute encryption. In the access tree, if the current node is not a leaf node, a random polynomial will be generated. The constant term of the polynomial is the secret value of the current node and this value will be used for sharing. The algorithm runs recursively and the secret continues to be shared, as shown in Figure 4.



```
public static void nodeShare(Node[] nodes, Node n, Pairing bp){
    if (!n.isLeaf()){
        Element[] coef = randomP(n.gate[0], n.secretShare, bp);
        for (int j=0; j<n.children.length; j++ ){
            Node childNode = nodes[n.children[j]];
            childNode.secretShare = qx(bp.getZr().newElement(n.children[j]), coef, bp);
            nodeShare(nodes, childNode, bp);
        }
    }
}
```

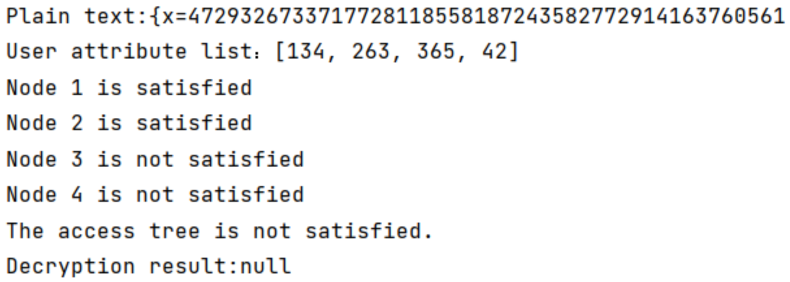
Figure 4. Secret sharing algorithm process.

When the user authorizes the logistics personnel and meets the set attribute list, the encrypted address ciphertext will be decrypted, as shown in Figure 5. Other unauthorized employees will not be able to view the address ciphertext, thus enhancing privacy protection, as shown in Figure 6.



```
Plain text:{x=6186276076135486026413433762365204782533958755
User attribute list：[134, 263, 365, 42]
Node 1 is satisfied
Node 2 is satisfied
Node 3 is satisfied
Node 4 is satisfied
Decryption result:{x=6186276076135486026413433762365204782533958755
Decrypted successfully
```

Figure 5. The result of ciphertext decryption when the attribute node is satisfied.



```
Plain text:{x=4729326733717728118558187243582772914163760561
User attribute list：[134, 263, 365, 42]
Node 1 is satisfied
Node 2 is satisfied
Node 3 is not satisfied
Node 4 is not satisfied
The access tree is not satisfied.
Decryption result:null
```

Figure 6. The result of ciphertext decryption when any attribute node is not satisfied.

The time cost of using different attribute nodes to test CP-ABE startup, key generation, encryption, and decryption functions is shown in Figure 7. The detailed configuration is shown in Table 3. This simulation experiment is written in Java and the encrypted data size is 128 bytes. The encryption module has a higher overhead. When testing with 10 attributes, the total time of CP-ABE does not exceed 1.5 s, of which the CP-ABE encryption operation performed by the logistics company management server takes less than 0.6 s and the CP-ABE decryption operation performed by the logistics company management server takes less than 0.2 s. In general logistics systems, the usage attributes are around 5. This can meet the needs of normal logistics and transportation processes.

Table 3. Software and hardware environment configuration.

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

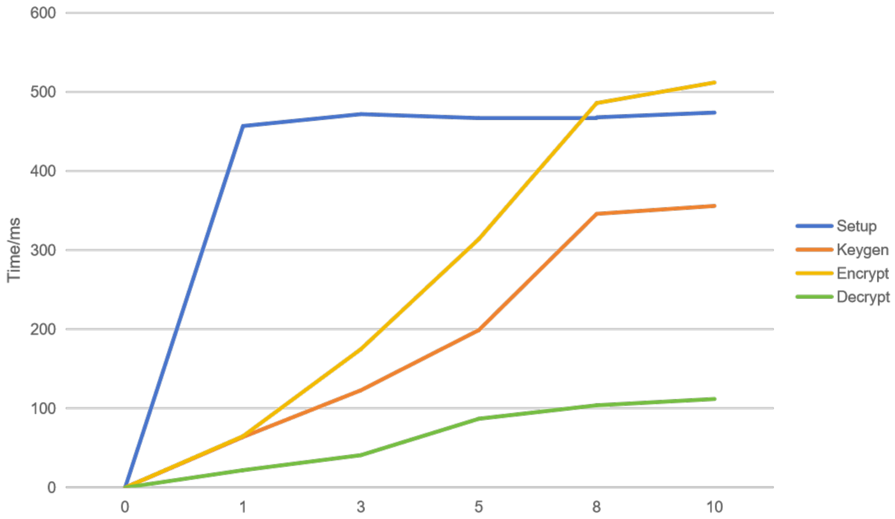
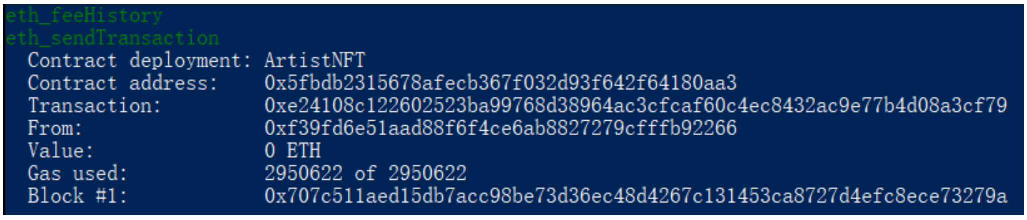


Figure 7. CP-ABE encryption algorithm overhead.

Smart contracts are utilized to automate the execution of predefined tasks in this project. Specifically, in the context of the smart contracts project, these contracts are employed to manage the ownership and transactions of digital assets. The smart contract code is composed using Solidity and is deployed and executed on the blockchain. Upon the deployment of the smart contract on the blockchain, a non-fungible token (NFT) in compliance with the NFT standard (e.g., ERC721 or ERC1155) can be created. For the purpose of simulating transactions, a local test chain is deployed to acquire a number of test Ether coins, as shown in Figure 8.

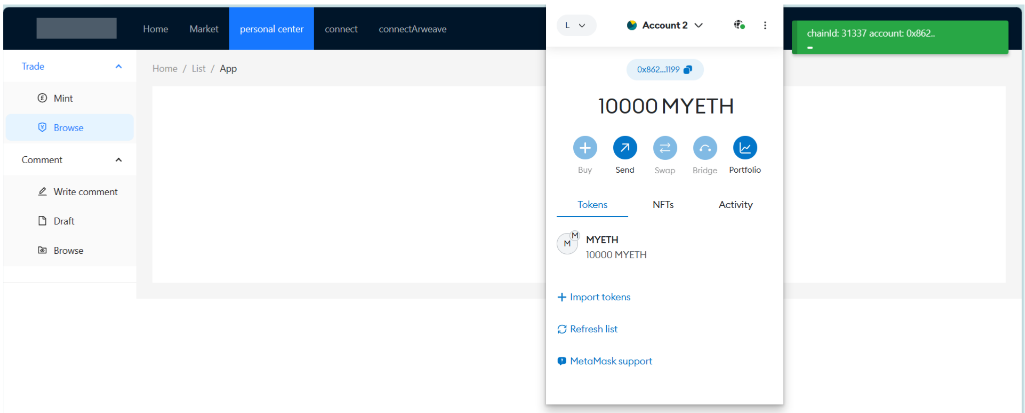


```
eth_feeHistory
eth_sendTransaction
  Contract deployment: ArtistNFT
  Contract address:    0x5fbdb2315678afecb367f032d93f642f64180aa3
  Transaction:         0xe24108c122602523ba99768d38964ac3cfcaf60c4ec8432ac9e77b4d08a3cf79
  From:                0xf39fd6e51aad88f6f4ce6ab8827279cfffb92266
  Value:               0 ETH
  Gas used:            2950622 of 2950622
  Block #1:            0x707c511aed15db7acc98be73d36ec48d4267c131453ca8727d4efc8ece73279a
```

Figure 8. Deploy the test chain locally to obtain test coins.

The experimental process involves designing the scheme and integrating wallets to execute transactions. The scheme implements decentralized storage using the InterPlanetary File System (IPFS), a distributed file system that enables file storage and access through a decentralized network architecture. In the context of smart contract projects, IPFS stores immutable data, such as metadata associated with digital collections. Housing metadata on IPFS can ensure data security and integrity. It is important to note that, when using IPFS to store non-fungible token (NFT) metadata, the file itself is not directly stored on IPFS. Instead, a hash value of the file is stored. This hash value safeguards the file from modification or corruption during transit, and can be used to verify the file's contents. The experiment is connected to the blockchain account, which can be used for transactions, as shown in Figure 9.

This empirical study substantiates the efficacy of both the encryption procedure and the account mechanism, thereby safeguarding the user's identity privacy throughout the transactional continuum and addressing the risk of data breaches instigated by internal attacks.



chainId: 31337 account: 0x862...

Figure 9. Connecting test tokens to the account wallet can be used for trading.

4. Discussion

In this scheme, the identity provider solely offers decentralized identity (DID), without interfering with the address of the subsequent identity user involved in logistics. The logistics company can access the actual addresses of the senders or receivers, but cannot associate the addresses with the users' real identities or any other information. Consequently, internal logistics staff is restricted from indiscriminately accessing users' identity data throughout the course of logistics transportation.

The logistics process usually involves the participation of multiple entities such as logistics companies, delivery centers, users, etc. In contrast to other domains, the logistics sector also encompasses the phase of freight transportation. Therefore, beyond the scope of internal data breaches, the logistics realm also confronts the risks of theft or manipulation during transportation. The following paragraphs analyze this from each of the perspectives of privacy and security.

From the privacy perspective, each time the user logs in with a one-time-use account, the logistics company cannot ascertain the actual identities of the senders or receivers. Instead, the logistics company only identifies the person actually responsible for sending and receiving the package, which can also be supervised. The design of the one-time-use account based on the DID minimizes the disclosure of identity information. With the design, it becomes challenging for the logistics company to correlate the user's multiple different addresses with identity information. At the same time, it reduces the amount of information obtained by other parties during the transportation process, thereby enhancing privacy and reducing the risk of data leakage. Additionally, the scheme allows senders and recipients to hide the address information from each other, which could reduce instances of unscrupulous merchants using address and phone number information to retaliate for negative reviews.

From the security perspective, if the private key is lost, the courier will continue to deliver goods according to the currently set path, with only modifying and viewing privileges being lost. The sender and recipient can contact the logistics company to change the route by co-signing. If the private key is stolen, the attacker will have the ability to modify the logistics information. However, stealing the goods requires a real identity, rendering arbitrary modification of logistics information meaningless.

Regulators have access to identity data and wallet accounts held by identity providers and users to associate real identities with wallet accounts for the purposes of sanctions or criminal investigations. This helps ensure the safety and integrity of the logistics process. Importantly, regulators provide evidence of malicious behavior in the account to be investigated before the identity provider grants access to corresponding identity data. To prevent the concentration of power in a few regulators, corresponding laws and supervision mechanisms must be in place. To avoid the situation where regulators become overly powerful and force identity providers to disclose user data, identity providers should have at least the same powers as regulators.

In terms of the traditional models, such as where logistics companies hide information on express delivery orders, companies cannot be prevented from abusing information. Thus, protection of user privacy is solely reliant on the companies' conscientiousness. However, companies often have internal attackers and the databases often face hacker attacks. As long as one logistics company is breached, a large amount of user information will be leaked. This scheme reduces the value of information held by logistics companies as an overall mechanism, prevents internal intrusions, and protects user data privacy.

The security analysis of a scheme is conducted to assess its resilience against potential vulnerabilities and various attack scenarios. Weaknesses in key generation, storage, or distribution may lead to the compromise of the confidentiality of encrypted logistics data. Additionally, vulnerabilities in encryption algorithms can render encrypted data susceptible to potential attacks, such as brute force attacks, chosen ciphertext attacks, or key recovery attacks. To mitigate these risks, the scheme should incorporate robust encryption key algorithms, adhere to key rotation practices, and implement stringent access controls to ensure that only authorized users have access to encryption keys. Moreover, the encryption algorithm should adhere to recommended key lengths and enforce secure encryption and decryption processes. It is imperative to conduct regular security audits and encryption algorithm evaluations to identify and rectify any potential encryption weaknesses or vulnerabilities. To bolster security measures, the incorporation of secure communication channels, deployment of end-to-end encryption protocols, and robust authentication mechanisms are essential in thwarting man-in-the-middle attacks.

Essentially, this scheme is suitable for a multi-party communication network including trusted third parties and identity verification nodes, and it provides identity privacy protection services for the communication process of the network. Some existing IoT systems such as smart cities and the Internet of Vehicles also work in multi-party communication networks, including trusted third parties and identity verification nodes, so the network structure they work in is similar to that in this scheme. Therefore, this scheme can be transplanted to those IoT systems with a similar network structure and provide them with protection.

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

Table 4. Comparison of schemes.

	Public Key Infrastructure	**Identity-Based Encryption**	**Homomorphic Encryption**	**Differential Privacy**	**The Scheme**
Cryptographic Primitives	Asymmetric cryptographic primitives	User identities as public keys	Computation on encrypted data	Adds noise to query results	Bilinear pairings and user attributes
Management Strategies	Certificate Authority (CA) to manage digital certificates	Identities and private keys simplifying the key retrieval process	Manage parameters for homomorphic operations	Selection of noise parameters	Attribute keys and specific access policies
Security Guarantees	Digital signatures and secure key management	Associating user identities with private keys	Guarantees privacy during computation	Data points remain indistinguishable	Selective disclosure
Is the account anonymous to the identity provider?	No	No	No	No	Yes
Will the loss of the private key affect its use?	Partial solutions support recovery	Partial solutions support recovery	Partial solutions support recovery	Partial solutions support recovery	No

5. Applications and Analysis

In order to guarantee protection against privacy breaches and potential vulnerabilities, it is essential to delve deeply into the technical intricacies of data security. Implementing secure and efficient encryption key management involves the generation, storage, and distribution of encryption keys, as well as key rotation and revocation mechanisms to prevent unauthorized access and data leakage. At the same time, implementing comprehensive vulnerability mitigation measures, such as regular security audits, penetration testing, and vulnerability assessments, can help identify and resolve potential security vulnerabilities and weaknesses in logistics systems, and proactively address deficiencies in media such as system architecture, software components, and network infrastructure.

The integration of one-time-use accounts and CP-ABE into the current logistics systems can pose various technical usability challenges. Incorporating disposable accounts and CP-ABE into the existing logistics system might necessitate modifications to the system architecture and data management processes. The implementation of CP-ABE could introduce additional computational overhead and latency during the encryption and decryption procedures. To address these challenges, one can consider leveraging parallel processing capabilities, such as multi-threading or distributed computing, which can effectively distribute computing workloads across multiple processing units. This strategy can significantly enhance overall processing speed and system responsiveness by partitioning intricate tasks into smaller parallelizable components.

The distributed computing mentioned above, such as MapReduce or Apache Hadoop, facilitates the processing of extensive datasets across multiple nodes or clusters. By harnessing the power of distributed computing techniques, the system can efficiently handle intricate computations and large-scale data processing tasks, ensuring scalability and high-performance data analysis. Additionally, the implementation of caching mechanisms involves the storage of frequently accessed data or computation results in cache memory, leading to reduced response times for subsequent requests. By integrating these caching techniques, the system can effectively minimize computational overhead and enhance data retrieval efficiency, particularly for frequently accessed data records or computations. The incorporation of these scalability algorithms and techniques enables the system to adeptly manage a substantial volume of users and data records, guaranteeing optimal performance, responsiveness, and resource utilization, even when facing a growing workload.

Adapting the scheme for application in diverse network structures beyond logistics requires several key technical adaptations and adjustments. Notably, IoT networks often comprise a significant number of interconnected devices with varying computational capabilities and communication protocols. To accommodate this, the scheme needs to be tailored for IoT scenarios, necessitating the development of a scalable and lightweight architecture capable of efficiently managing the diverse network topology and resource constraints of IoT devices. Furthermore, it should be designed to be compatible with various IoT protocols and communication standards, such as MQTT, CoAP, or LoRaWAN, thereby enabling secure data transmission and promoting interoperability across different IoT devices and platforms.

In some different logistics environments, different characteristics may be analyzed as follows: Warehouse environments mainly include protecting sensitive data related to inventory management, order fulfillment, and shipment tracking. A supply chain management framework ensures the effectiveness of data exchange among multiple stakeholders. Transportation management systems primarily protect data related to vehicle tracking, route optimization, and transportation schedules. Before being compatible with the above systems, the compatibility of cryptographic algorithms, data formats, and communication protocols needs to be evaluated to ensure smooth data exchange and interoperability between the scheme and existing infrastructure. Through the data migration process, data attributes from the current system are mapped to those required by the proposed scenario, ensuring a seamless transition without data loss or corruption. Additionally, the reinforcement of data transmission channels, the establishment of access control mechanisms, and the execution of comprehensive security audits can aid in identifying and mitigating potential security vulnerabilities.

6. Conclusions

This paper proposes an identity privacy protection scheme in the logistics field. This scheme solves the issue of data breaches caused by internal intrusions.

- One-time-use accounts replace single accounts, enhancing data security and user authentication within logistics systems. Users can use the accounts for logistics transactions while preventing internal privacy leaks.
- In order to ensure the user's personal privacy and logistics route information, the integrated attribute encryption CP-ABE effectively implements access control strategies to ensure authorized data access and prevent unauthorized data leakage. Through enhanced data integrity and confidentiality measures, critical logistics data are safeguarded during storage, transmission, and processing.
- The introduction of regulators oversees the conduct of all involved entities, thereby safeguarding the rights and interests of all stakeholders.

This scheme establishes a trusted environment between users and other logistics participants, thereby fostering the salubrious evolution of the logistics sector. In addition, this scheme is suitable for parts of IoT scenarios with similar network structures, such as smart cities, Internet of Vehicles, etc. Future work will focus on developing an example project for demonstration purposes and is intended to optimize the scheme's efficiency and other aspects.

Author Contributions: N.S. and Y.Z. were the advisors. C.Z. and Y.Z. designed the scheme. C.Z. carried out the implementation. C.Z. wrote the manuscript. N.S., Y.Z. 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.

Data Availability Statement: Not applicable.

Conflicts of Interest: Yuanyi Zhang has been involved as a consultant and expert witness in Company Shanghai Shentie Information Engineering Co. The authors declare no conflict of interest.

References

1. Ding, Y.; Jin, M.; Li, S.; Feng, D. Smart logistics based on the internet of things technology: An overview. *Int. J. Logist. Res. Appl.* **2021**, *24*, 323–345. [CrossRef]
2. Lan, S.; Yang, C.; Huang, G.Q. Data analysis for metropolitan economic and logistics development. *Adv. Eng. Inform.* **2017**, *32*, 66–76. [CrossRef]
3. Niu, B.; Dai, Z.; Chen, L. Information leakage in a cross-border logistics supply chain considering demand uncertainty and signal inference. *Ann. Oper. Res.* **2022**, *309*, 785–816. [CrossRef]
4. Hunter, T. *2022 Data Asset Breach Analysis Report*; 2023. Available online: https://threathunter.cn/reportDetail/ (accessed on 10 August 2023)
5. Ouyang, J.; Chen, X. Personal Information Two-dimensional Code Encryption Technology in the Process of E-commerce Logistics Transportation. *SAIEE Afr. Res. J.* **2022**, *113*, 52–57. [CrossRef]
6. Feng, H. Application of QR Code Technology in the Design of User Information Privacy Protection Logistics System. *Int. J. Front. Eng. Technol.* **2021**, *3*, 6–10.
7. Zhang, X.; Li, H.; Yang, Y.; Sun, G.; Chen, G. LIPPS: Logistics information privacy protection system based on encrypted QR code. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; pp. 996–1000.
8. Rani, M.M.S.; Euphrasia, K.R. Data security through qr code encryption and steganography. *Adv. Comput. Int. J. (ACIJ)* **2016**, *7*, 1–7.
9. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Boston, MA, USA, 11–14 December 2017; pp. 557–564.
10. Li, H.; Han, D.; Tang, M. Logisticschain: A blockchain-based secure storage scheme for logistics data. *Mob. Inf. Syst.* **2021**, *2021*, 8840399. [CrossRef]
11. Sun, Z.; Han, D.; Li, D.; Wang, X.; Chang, C.C.; Wu, Z. A blockchain-based secure storage scheme for medical information. *EURASIP J. Wirel. Commun. Netw.* **2022**, *2022*, 40. [CrossRef]
12. Zhou, Y.; Chen, L. Secure Storage and Deletion Based on Blockchain for Cloud Data with Fine-grained Access Control. *Dianzi Yu Xinxi Xuebao* **2021**, *43*, 1856–1863.

13. Tijan, E.; Aksentijević, S.; Ivanić, K.; Jardas, M. Blockchain technology implementation in logistics. *Sustainability* **2019**, *11*, 1185. [CrossRef]
14. Perboli, G.; Musso, S.; Rosano, M. Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *IEEE Access* **2018**, *6*, 62018–62028. [CrossRef]
15. Waseem, M.; Adnan Khan, M.; Goudarzi, A.; Fahad, S.; Sajjad, I.A.; Siano, P. Incorporation of blockchain technology for different smart grid applications: Architecture, prospects, and challenges. *Energies* **2023**, *16*, 820. [CrossRef]
16. Goudarzi, A.; Ghayoor, F.; Waseem, M.; Fahad, S.; Traore, I. A Survey on IoT-Enabled Smart Grids: Emerging, Applications, Challenges, and Outlook. *Energies* **2022**, *15*, 6984. [CrossRef]
17. Zangui, M.; Zhou, Y.; Yin, Y.; Chen, S. *Privacy-Preserving Methods to Retrieve Origin-Destination Information from Connect Vehicles*; Technical Report; University of Florida, Center for Multimodal Solutions for Congestion Mitigation: Gainesville, FL, USA, 2013.
18. Papadamou, K.; Charalambous, M.; Papagiannis, P.; Stroinea, I.; Passas, N.; Xenakis, C.; Sirivianos, M. IdeNtity verifiCatiOn with Privacy-preservinG credeNtIals for Anonymous Access to Online Services. INCOGNITO_D4. 1_revised_final_v3. pdf. Available online: https://incognito.socialcomputing.eu/news-events/ (accessed on 25 October 2023).
19. Bissessar, D.; Liu, D.; Nahmias, S.; Harvey, J.; Hubbard, P. *Architecture and Assessment: Privacy Preserving Biometrically Secured Electronic Documents*; Technical Report; 2015. Available online: https://candid.drdc-rddc.gc.ca/ (accessed on 24 October 2023).
20. Stallings, W. Handling of personal information and deidentified, aggregated, and pseudonymized information under the California consumer privacy act. *IEEE Secur. Priv.* **2020**, *18*, 61–64. [CrossRef]
21. Chaum, D. Security without identification: Transaction systems to make big brother obsolete. *Commun. ACM* **1985**, *28*, 1030–1044. [CrossRef]
22. 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]
23. Kang, J.; Yu, R.; Huang, X.; Zhang, Y. Privacy-preserved pseudonym scheme for fog computing supported internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 2627–2637. [CrossRef]
24. Maram, D.; Malvai, H.; Zhang, F.; Jean-Louis, N.; Frolov, A.; Kell, T.; Lobban, T.; Moy, C.; Juels, A.; Miller, A. Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability. In Proceedings of the 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 24–27 May 2021; pp. 1348–1366.
25. Kang, M.; Lemieux, V. A decentralized identity-based blockchain solution for privacy-preserving licensing of individual-controlled data to prevent unauthorized secondary data usage. *Ledger* **2021**, *6*. . [CrossRef]
26. Halpin, H. Nym credentials: Privacy-preserving decentralized identity with blockchains. In Proceedings of the 2020 Crypto Valley Conference on Blockchain Technology (CVCBT), Virtual, 15 June 2020; pp. 56–67.
27. Luecking, M.; Fries, C.; Lamberti, R.; Stork, W. Decentralized identity and trust management framework for Internet of Things. In Proceedings of the 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), Toronto, ON, Canada, 2–6 May 2020; pp. 1–9.
28. Mohammadinejad, H.; Mohammadhoseini, F. Privacy protection in smart cities by a personal data management protocol in blockchain. *Int. J. Comput. Netw. Inf. Secur.* **2020**, *11*, 44. [CrossRef]
29. Feng, Q.; He, D.; Zeadally, S.; Khan, M.K.; Kumar, N. A survey on privacy protection in blockchain system. *J. Netw. Comput. Appl.* **2019**, *126*, 45–58. [CrossRef]
30. Bünz, B.; Agrawal, S.; Zamani, M.; Boneh, D. Zether: Towards privacy in a smart contract world. In *Proceedings of the International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 423–443.
31. Xu, H.; Zhang, L.; Sun, Y.; BE-RAN: Blockchain-enabled open RAN with decentralized identity management and privacy-preserving communication. *arXiv* **2021**, arXiv:2101.10856.
32. Nakamoto, S. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Bus. Rev.* **2008**. . [CrossRef]
33. Fu, J.; Cao, B.; Wang, X.; Zeng, P.; Liang, W.; Liu, Y. BFS: A blockchain-based financing scheme for logistics company in supply chain finance. *Connect. Sci.* **2022**, *34*, 1929–1955. [CrossRef]
34. Szabo, N. Formalizing and securing relationships on public networks. *First Monday* **1997**. Available online: https://firstmonday.org/ojs/index.php/fm/article/download/548/469 (accessed on 25 October 2022).
35. Wood, G.; Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
36. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. In Proceedings of the 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, 20–23 May 2007; pp. 321–334.
37. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Proceedings of the International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 53–70.
38. Van Saberhagen, N. CryptoNote vs. 2.0. 2013. Available online: https://www.getmonero.org/ (accessed on 15 October 2022).

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.