mathematics

MDPI

Article

Achieving a Secure and Traceable High-Definition Multimedia Data Trading Scheme Based on Blockchain

Shuguang Zhao [1], Zhihua Zeng [1], Jiahui Peng [2,*] and Feng Yu [3,4,5,*]

[1] State Key Laborataory of Nuclear Power Safefy Monitioring Technology and Equipment, China Nuclerar Power Engineering Co., Ltd., Shenzhen 518172, China
[2] China Southern Power Grid Supply Chain Group (Guangxi) Co., Ltd., Nanning 530022, China
[3] School of Computer Science & Engineering, Guangxi Normal University, Guilin 541004, China
[4] Key Laboratory of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, Guilin 541004, China
[5] Guangxi Collaborative Innovation Center of Multi-Source Information Integration and Intelligent Processing, Guangxi Normal University, Guilin 541004, China
* Correspondence: pengjh051x@im.csg (J.P.); yufeng@gxnu.edu.cn (F.Y.)

Abstract: The Internet has penetrated into every aspect of life. Large amounts of data are generated by multimedia collection equipment every day. As an asset, data can achieve value circulation through transactions. However, the existing centralized transaction model is not secure enough, has the risk of user privacy leakage, and the protection of data copyright is insufficient. In this paper, in order to solve the transaction security and traceability problems of multimedia data, especially high-definition data such as vector graphics, we implement a transaction scheme STTS without third-party based on blockchain. For high-definition multimedia data, we use zero watermarking combined with oblivious transfer to embed copyright information. A two-stage verification process is then implemented by using group signature and a secret sharing scheme to complete data distribution. Finally, the smart contract is used to complete copyright tracking. We test the performance of our scheme by simulating the real transaction environment in the Internet of Things (IoT) and demonstrate the feasibility of our scheme, which can be applied to large-scale multimedia data trading schemes in the IoT.

Keywords: data trading; high-definition multimedia data; blockchain; group signature; secret sharing scheme

MSC: 94A60

Citation: Zhao, S.; Zeng, Z.; Peng, J.; Yu, F. Achieving a Secure and Traceable High-Definition Multimedia Data Trading Scheme Based on Blockchain. *Mathematics* **2023**, *11*, 2224. https://doi.org/10.3390/math11102224

Academic Editors: Sheng Li, Zhenjun Tang and Guorui Feng

Received: 11 April 2023
Revised: 1 May 2023
Accepted: 8 May 2023
Published: 9 May 2023

1. Introduction

The popularity of cloud–edge collaboration has led to the explosive development of the Internet of Things, and the collection of multimedia data is occurring more widely. Multimedia data, especially high-definition data, has obvious value attributes and can transfer value through transactions [1,2]. Attention has been widely paid to how high-definition multimedia data can achieve secure transactions [3]. High-definition multimedia data can be used in many fields. For example, vector maps are widely used in the field of geographic information, and remote sensing maps can effectively identify forest fires [4,5]. In addition, high-definition multimedia data have also been effectively applied in the supply chain field [6]. For example, in supply chain service and deployment tasks, high-definition video can be used to monitor the arrival of materials, ascertain whether the trucks are driven and delivered according to the prescribed routes, and effectively monitor the bid evaluation process through high-definition video. If the high-definition video is not clear enough, the efficiency of these works will be greatly reduced [7].

Despite the many benefits of trading high-definition multimedia data, there is still an inevitable security issue. For example, vector graphics store a small amount of data and

are easy to copy and distribute [8]. In addition, in the process of centralized trading, the transaction security of the data owner is not guaranteed, and the problem of data leakage and user privacy leakage cannot be effectively avoided (for example, data and transaction information can be obtained on the Internet). No matter who the leakage affects, the whole process of the transaction should be tracked and the copyright should be bound [9]. Once the responsible person can be traced, punishment is required, namely accountability. To some extent, solving problems effectively can increase the activity of the market [10].

To solve the security and traceability problems of multimedia data transactions, three conditions need to be met. First of all, copyright information should be embedded in the data, especially high-definition data such as vector images, which requires strict embedding information. General digital watermark and digital fingerprint embedding may affect its clarity. Second, use of centralized trading methods should be avoided because the centralized trading platform will collect private user data, and the centralized storage of transaction data will have the risk of tampering and loss. Third, the communication frequency between IoT devices should not be too large, especially if cryptography is used to ensure security. Too much communication will greatly reduce the interaction time between IoT devices. Many problems need to be considered to meet the three requirements at the same time, especially in the real application of the IoT, and the first two conditions can be solved by using blockchain technology [11,12] and zero-watermarking technology [13]. As a platform without a third party, blockchain can effectively prevent the tampering of transaction data and provide traceable storage for transaction records and even copyright authentication records. Watermarking technology for high-definition multimedia data is mainly divided into three categories. The first is reversible watermarking [14], which completely recovers the original data after extracting the watermark but loses protection after data recovery, which results in copyright protection not being possible to guarantee permanently. The second is lossless watermarking [15]. Watermark information is embedded by modifying the attribute domain. However, this scheme will retain redundant space when defining attributes and lead to information loss when processing data embedding watermarking, resulting in poor effect in real application. The third technology is zero-watermarking technology [13,16], which belongs to the structural watermarking algorithm. Currently, it is divided into two categories: spatial domain and frequency domain. Because the construction mode of watermark is closely related to the precision of coordinates, it can provide permanent copyright protection without damaging data information. However, zero watermark is used in centralized server control scenarios, which are considered unsafe. How to solve this limitation is the key to the wide application of zero watermarking.

Current work is focused on using blockchain to solve the transaction security problem of multimedia data [17]. The key to solving security and fairness issues on blockchain is smart contract [18], because it is distributed and transparent, and all participants on the chain can see the transaction information and calculation process. In addition to smart contracts, hash technology [11] and encryption technology are also widely used in multimedia data security transactions, mainly used to solve copyright problems. For example, Sun et al. [19] use perceptual hash to complete copyright tracking. Holland et al. [20] designed an image transaction method with copyright protection based on blockchain. Huang et al. [21] combined cryptography and blockchain for trusted access control, which is similar to digital rights management [22]. The reason for using blockchain is to avoid the hassle of third parties. Although the above work has some effect on the transaction of multimedia data on the blockchain, there are few studies on the security and traceable transaction mode of high-definition multimedia data under the Internet of things. The scheme [23] uses zero watermark to protect the copyright of vector images on the blockchain, but the smart contract is only used for the preservation of watermark information. It does not protect the security of the whole transaction mode. Mangipudi et al. [24] uses oblivious transfer combined with watermark to realize a traceable sharing mechanism in the Internet of Things. However, the access control permission based on zero-knowledge proof has low efficiency, so whether it can be applied to the environment of Internet of Things with a huge

amount of devices has not been clarified. Some works also use SGX, a trusted execution environment, and smart contracts to realize secure access control. For example, Han et al. [25] implemented a secure access control framework based on attribute-based access control policies based on blockchain and SGX. Fang et al. [26] implemented a high-performance smart contract execution environment based on permissioned blockchain and SGX. While all of these efforts effectively combine SGX and smart contracts, there is no mention of how data can be securely transferred.

Our scheme realizes a secure and traceable high-definition multimedia data transaction scheme based on blockchain called STTS. It should be noted that we mainly use vector graphs as the research object. We use zero-watermark combined with 1 out of 2 oblivious transfer to realize the reliable transmission of data under the Internet of Things, and use blockchain to save watermark information for copyright protection of high-definition multimedia data. Then, the secure data access control mechanism is realized by using the group signature and secret sharing scheme, and the key of the acquired data is divided as the secret. Then, two-stage verification between the group administrator and transaction providers is used to complete the data download. In addition, since high-definition multimedia data require a lot of storage space, we use the distributed storage system IPFS to complete image storage. In summary, the main contributions are as follows.

(1) Trusted copyright information embedding is realized by using the 1 out of 2 oblivious transfer protocol combined with zero watermarking. We use smart contract to realize the tamper-proof, traceable and traceable copyright information storage mechanism of watermarking.

(2) The group signature scheme combined with the secret sharing scheme is used to realize data access permission control in the multi-device environment of the Internet of Things. The private key is classified as a secret, and two-stage authentication between the transaction provider and the group administrator is completed by using the smart contract. In addition, trusted computing on smart contracts, we invoke the enclave module of the trusted execution environment.

(3) We used Raspberry Pi to build a real Internet of Things environment for test simulation, used the alliance architecture to simulate the interaction between multiple devices and verification experiments, and used smart contracts to complete the rewards and punishments within the system. The experiments show that STTS can ensure security and traceability in the Internet of Things environment while having high communication efficiency. The gas and time test of the smart contract meet the performance requirements of the alliance and can be applied to the real IoT environment.

2. Preliminaries

2.1. Oblivious Transfer

Michael O. Rubin proposed the oblivious transfer protocol in 1983 [27]. The main function of this protocol is that the receiver can obtain the message from the sender with a probability of $1/2$, but the sender cannot know whether the receiver has received the message, which ensures the communication security of both sides. Suppose that Alice and Bob each hold data $f(A)$ and $f(B)$, and both Alice and Bob have the secret (S_A, S_B) to decrypt the data. If Alice acquires the secret $S(S \neq S_B)$ from Bob through a secure channel and uses S to decrypt the file, such illegal operations may directly cause data corruption. Therefore, forcing Bob to sign each message he sends prevents him from passing the wrong secret. If Alice uses the secret to decrypt the data and causes them to be corrupted, Alice can file a complaint based on Bob's signature as evidence. In addition, it is necessary to ensure the atomicity of the secret transmission; that is, Alice must receive S_B immediately after sending S_A, and the same is true for Bob.

2.2. Zero-Watermark

Firstly, cat map [28] is used to scramble or encrypt the image. Equation (1) describes the transformation process:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & b \\ a & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod(N) \tag{1}$$

For any $N \times N$ image or video composed of multimedia data, assuming that image pixel coordinates transform (x_n, y_n) into (x_{n+1}, y_{n+1}) by scrambling parameter a and b, the original image can be restored after periodic mapping of cat map. The inverse transformation process is shown in Equation (2).

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} ab+1 & -b \\ -a & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} mod(N) \tag{2}$$

The image is normalized by reference [12], as shown in Equations (3) and (4).

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f(x,y) \tag{3}$$

$$\varpi_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \tag{4}$$

where

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \tag{5}$$

$f(x,y)$ is an image of $M \times N$ dimension, m_{pq} is geometric moment, ϖ_{pq} is central moment, and $p, q \in \{0, 1 \ldots n\}$ is a parameter.

2.3. Secret Sharing

The function of secret sharing is to share secrets among participants, which is mainly used to prevent information from being lost, damaged or tampered with. The secret sharing scheme (SSS) was proposed by Sharmir and Blakley in 1979 [29]. The Shamir-based secret sharing mechanism assumes that the secret publisher s, the participant $P\{P_1, P_2, \ldots, P_n\}$. The secret is shared among n participants. Only when more than t participants cooperate can the secret be recovered. The shared secret s is divided into $\{s_1, s_2, \ldots, s_n\}$ and distributed to $\{t_1, t_2, \ldots, t_n\}$.

2.4. Short Group Signature

Short group signatures were proposed by Boneh et al. in 2004 [30]. In short signatures, any member of a group can sign a message, but the resulting signature does not reveal the identity of the signer. Specifically, given a short group signature G and group $n\{n_1, n_2, \ldots, n_i\}$, the verifier cannot obtain the identity information of n_i through G. Compared with the group signature project based on Strong-RSA [31], the short length of the short group signature is more suitable for environments with high communication requirements.

2.5. Blockchain and Smart Contract

Blockchain is essentially a distributed database, a technology based on cryptography, consensus protocols, P2P networks and smart contracts. Satoshi Nakamoto proposed the concept of bitcoin in 2008 [32]. The current blockchain is mainly divided into three categories.

- **Public blockchain.** Allowing everyone to access the network and transact, any miner can implement the consensus protocol, which has been replaced by the equity consensus protocol of the public blockchain [33].

- **Private blockchain.** Miners' access is severely restricted only through a blockchain network that can be joined with the permission of the administrator, and this type of blockchain is suitable for internal organizational use [34].
- **Consortium blockchain.** Some organizations form coalitions to build blockchain networks, which are considered weakly decentralized because they contain the authority of an administrator [35].

Smart contracts allow users to execute scripts on the blockchain to complete certain actions [12]. The original smart contracts were proposed on Ethereum to complete the calculation and accounting of transactions. Smart contracts are composed of functions and data, which require pre-defined trigger conditions and response actions of functions. Smart contracts help blockchain complete decentralized operations and play a fair role.

3. Problem Statement

In this section, we will explain the problems studied in this paper and the framework of the scheme, mainly explaining the steps of the scheme and the problems to be solved.

3.1. System Model

In this scheme, there are mainly six entities, which are trusted agent, smart contract, group administrators, transaction providers, sender and receiver. The roles of entities will be explained below.

Trusted Agent (TA): TA generates the necessary parameters and authenticates participants when they appeal the group key.

Smart Contract: The smart contract completes the dual verification process of the secret sharing scheme and group signature, sends the key to the participant, and records the transaction information, especially the copyright information, to the world state of Ethereum. Most importantly, smart contracts guarantee the security and fairness of transactions.

Group administrator: The group administrator and TPs complete the verification of the data distribution, and only after the verification can the data be downloaded. The group administrator is the key to ensure the security of the group signature. He needs to constantly check and compare the verification information from the Internet of Things devices to confirm that the TP has the conditions for secret reconstruction.

Transaction providers (TPs): TP is a device of the Internet of Things, which is responsible for providing services for transactions. Specifically, it saves the data processed by watermark and 1 out of 2 oblivious transfer. When there is a transaction request, it completes the secret reconstruction and data sending after verification and comparison by the group administrator.

Sender: They can be regarded as a data owner who can make money in the blockchain network by processing the data they want to trade into a secure format.

Receiver: They can be seen as a data buyer who wants to download their multimedia data from the blockchain network for academic or other purposes, or a pirate who downloads the data for illegal resale.

The system framework of this paper is shown in Figure 1. Firstly, TA generates some necessary parameters for identity authentication, and the sender divides the data into blocks. Then, the blocks of data are copied, and the zero-knowledge watermark is embedded in two identical forms of data. The data are then sent to the TPs via 1 out of 2 oblivious transfer, which ensures that the TPs cannot acquire the complete data and that several TPs conspired to reassemble the data. At this time, if a recipient initiates a transaction request, the smart contract is required to confirm the transaction request and notify the sender meeting the transaction requirements. After the sender confirms the transaction, TPs sends a secret reconstruction request to the group administrator. After the two parties perform two-stage verification, the receiver obtains the encrypted information of the IPFS address through the smart contract, decrypts the IPFS address using the sender's public key, and downloads the corresponding data from IPFS. Finally, the smart contract completes the accounting of the transaction.

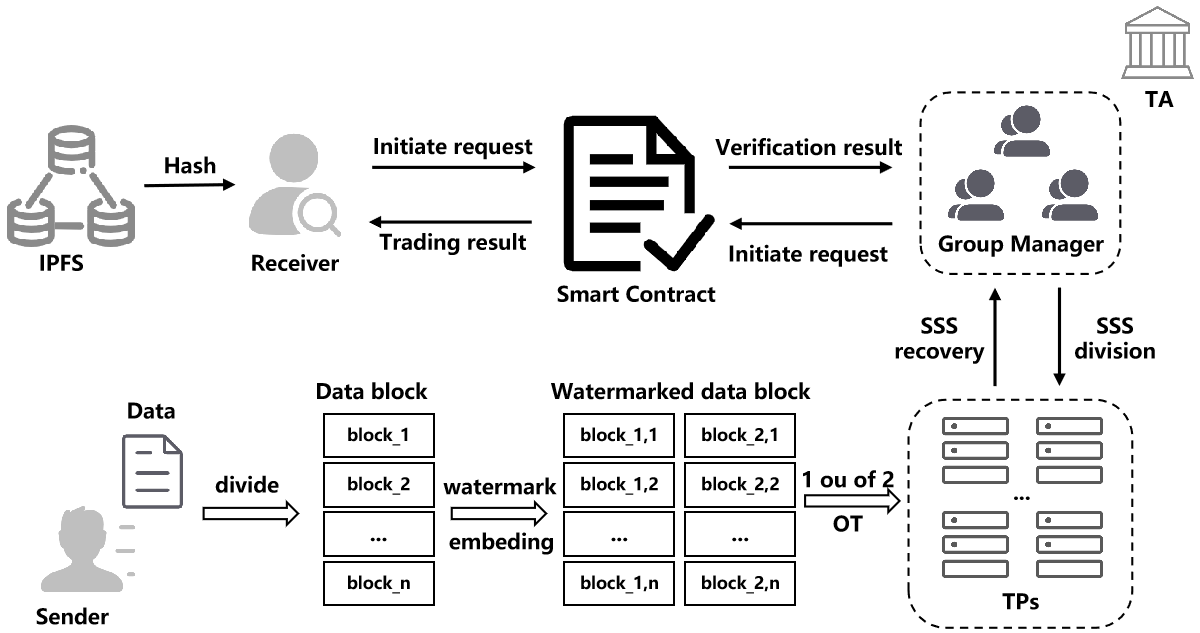

Figure 1. Framework of the scheme.

3.2. Research Problem

In the above system model, there are two major problems to be solved, which are summarized as the following three research questions.

Atomicity. For both sides of the transaction, the atomicity of the transaction means that the sender receives the proceeds after the receiver receives the data, and the receiver receives the multimedia data they wish to purchase after payment.

Security. The scheme can ensure both the information of the transaction participants and the anonymity of the identity, and the data can be stored securely in the distributed storage environment.

Privacy. The scheme first ensures that the user information stored in the system is not disclosed to the public, and then ensures that no one can infer the information and preferences of transaction participants from the transaction information.

4. STTS: Achieving a Secure and Traceable Vector Graph Trading Scheme Based on Blockchain

4.1. High-Definition Multimedia Data Copyright Protection Scheme

As shown in Figure 1, the sender performs data processing under the chain. First, the data needs to be divided into blocks, then the same data are copied and zero watermark is embedded in the data. We adopt the zero-watermarking scheme proposed in the literature [36], which has advantages in efficiency and robustness. It should be noted that our scheme is mainly aimed at realizing efficient copyright protection on blockchain, rather than studying zero watermarking. Therefore, any efficient zero-watermarking scheme can be adapted to STTS. After the data are embedded with zero watermarking, they are distributed to TPs using 1 out of 2 oblivious transfer. The specific steps are as follows.

(1) The sender first sends the personal information to the TA for authentication and then binds all N licenses to the data and encrypts all N licenses. The encrypted N licenses are set as $L_1, L_2, \ldots, L_N$, and the N encryption keys are $M_1, M_2, \ldots, M_N$, respectively. $L_1, L_2, \ldots, L_N$ are sent to the sender.

(2) TA generates RSA public key (N, e) and private key d. For $1 \leq i \leq N$, TA computes $K_i(H(i))^d \bmod N$, $E_i = G(K_i||i) \bigoplus (M)i)$ (H and G are secure Hash functions), hides $M_1, M_2, \ldots, M_N$ as $E_1, E_2, \ldots, E_N$, and sends $E_1, E_2, \ldots, E_N$ to the sender.

(3) After receiving $L_1, L_2, \ldots, L_N$ and $E_1, E_2, \ldots, E_N$, the sender selects a 128-bit random number r, calculates $Y = r^e H(n) \bmod N$, hides their choice n, and then sends Y to TA.

(4) TA signs the received information: $Sig = Y^d \bmod N$, and then sends it to the sender. The sender verifies $K_n = Y_d / r = H(n)^d$ after receiving the signature, and then obtains TA's license key $M_n = E_n \bigoplus G(K_n||n)$ and sends it to TPs.

4.2. Access Control Mechanism Based on Group Signature and Secret Sharing Scheme

In this section, we will introduce in detail the proposed access control mechanism based on the group signature and secret sharing scheme. The trusted agent (TA) is responsible for initializing and generating the necessary parameters. The group administrator and TPs verify each other through the smart contract and then retrieve the data they want to download after recovering the split secret.

- **Initialization.**

Initialize parameters. TA generates a group of public and private keys, generates n and a random threshold k based on the number of TPs, where the value of k does not affect the initial parameter because the verification process needs to verify the equation, and then inputs a security parameter $\epsilon > 1, k, l \in N$. Let G_1 and G_2 be the group, p is order, $g_1 \to G_1$, $g_2 \to G_2$, g_1 and g_2 are the generators of G_1 and G_2, and the system randomly selects the parameter $\lambda_1, \lambda_2, \tau_1, \tau_2$, where $\lambda_1 > \epsilon(\lambda_2 + k) + 2, \lambda_2 > 4l, \tau_1 > \epsilon(\tau_2 + k) + 2, \tau_2 > \lambda_2 + 2$. Define integer ranges: $R_1 = [2^{\lambda_1} - 2^{\lambda_2}, 2^{\tau_1} + 2^{\tau_2}], R_2 = [2^{\tau_1} - 2^{\tau_2}, 2^{\tau_1} + 2^{\tau_2}]$. We define two random numbers $\mu_1, \mu_2 \in Z_p^*$, and the group private key is $G_{sk} = (\mu_1, \mu_2)$.

- **A secret sharing scheme for two-way authentication.**

Based on RSA public key cryptography mechanism and secret sharing scheme, we use smart contract to realize the verification process. There are three stages: distribution and verification of subkeys, distribution of group secret keys and reconstruction of group secret keys.

(1) Distribution and verification of subkeys. The group administrator randomly selects n different x_i in Z_p^* as the subkey of U_i (TPs) and then randomly selects $a_i \in Z_p^*$ and $d_i \in Z_p^*$, to construct the polynomial $f_i(x) = a_i + d_1 x + d_2 x^2 = \cdots + d_{i-1} x^{i-1} \ mod \ R$ with order $i - 1$. Then calculate

$$\begin{cases} F_{ij} = f_i(ID_j) \ mod \ R, \\ Y_{ij} = g^{F_{ij}} \ mod \ N, \\ K_{ij} = x_i \bigoplus F_{ij}. \end{cases} \tag{6}$$

where F_{ij} is the polynomial value corresponding to the identity of TPs, Y_{ij} is public information (stored in the ledger via smart contracts) of F_{ij}, and K is the secret share of TPs.

Then the group administrator publishes $\{K_{ij}, Y_{ij}, g^{a_i}, g^{d_1} \ldots, g^{d_{i-1}}\}$ on the smart contract, and sends x_i to U_i through the contract. After U_i receives x_i, it obtains $\{K_{ij}, g^{a_i}, g^{d_1} \ldots, g^{d_{i-1}}\}$ from the contract, calculates $x_{ij} = x_i \bigoplus K_{ij}$, and then uses:

$$g^{x_{ij}} = g^{a_i} \prod_{m=1}^{i-1} (g^{d_m})^{(ID_i)^m} \ mod \ N \tag{7}$$

Verify the authenticity of x_i. If Equation (7) is not true, it means that the subkey received by U_i is false, and U_i can ask the group administrator to redistribute the subkey to it.

(2) Distribution of group secret keys.

The group administrator adopts different thresholds based on the $S_1, S_2, \ldots, S_k$ security level. Without loss of generality, set the threshold value corresponding to S_i as i, and

distribute information to be published on the contract. The group administrator randomly selects different integer r_i on Z_p^* to calculate the public information as:

$$\begin{cases} T_i = (g^{r_i})^d \ mod \ N, \\ H_i = g^{a_i r_i d} \bigoplus S_i. \end{cases} \tag{8}$$

Finally, the group administrator exposes $\{r_i, T_i, H_i\}$ on the smart contract.

(3) Reconstruction of group secret keys.

It is assumed that any l or more TPs in the secret S_l that need to be reconstructed send the reconstruction information to the group administrator, who can verify the TPs, thus improving the success rate of secret reconstruction. Let $W(|W| = L < k)$ be a subset of any l members of G; set $U_j \in W$ to participate in the reconstruction of secret $S_l \in S$. The operation steps of TPs are as follows:

a. After each $U_i \in W$ downloads (K_{lk}, T_l) from the smart contract, using the subkey x_j to calculate the group key reconstruction information is conducted as follows:

$$\begin{cases} B_{lj} = K_{lj} \bigoplus x_i, \\ A_{lj} = (T_l)^{B_{lj}} \mod N. \end{cases} \tag{9}$$

The TPs send A_{lj} to the group administrator through the contract.

b. After receiving A_{lj} sent by TPs, the group administrator downloads (Y_{lj}, r_l) from the contract and calculates:

$$(A_{lj})^e = (Y_{lj})_l^r \tag{10}$$

Verify the authenticity of A_{lj}. If Equation (10) is not true, it means that A_{lj} is false and TPs can be asked to send it again. Otherwise, proceed to step c.

c. After downloading H_l from the smart contract, the group administrator calculates:

$$S_l^{'} = H_l \bigoplus (\prod_{U_i \in W} (A_{lj})^{\Delta_j} \mod N) = S_l, \ \Delta_j = \prod_{U_i \in W, U_i \neq U_j} \frac{-ID_i}{ID_j - ID_i} \tag{11}$$

- **Joining in group.**

TPs can apply for joining a group through the group ID. After receiving the request, the group administrator verifies the certificate to determine the validity of the membership.

The TPs applying to join the group generate group public key G_{pk} and group private key G_{sk} locally, and G_{pk} is sent to the group administrator. The group administrator selects a random number $x \in Z_p^*$, encrypts the random number with its own public key to obtain ciphertext $m_j = encrpt(G_{pk}, x_i)$, and sends it to the group administrator P_i.

Group administrator P_i calculates the product of ciphertext to obtain ciphertext $m_p = \prod_{j=1}^{j=k} m_j$, and calculates $m_o = m_p m_k$, where $m_k = encrypt(G_{pk}, v)$, sends ciphertext m_p to TP to be added to the group, and TP decrypts m_o to obtain $w = decrypt(m_o, G_{sk})$, and calculates $y_i = g_1^{\frac{1}{w}}$ to obtain the complete private key (x_i, y_i), where $x_i = \sum_{j=1}^{j=k} x_j$, $y_i = g_1^{\frac{1}{v+\sum_{j=1}^{j=k} x_j}}$, and calculates the identity $Hash(y_i)$.

- **Transaction on-chain**

After completing signature verification and joining the group, TPs send the verified transaction and signature to the smart contract for saving. The specific algorithm of transaction linking is shown in Algorithm 1.

Algorithm 1 Transaction on-chain

Input: System parameter: $\{Y_i, g, h, c, b_1, b_2, h_1, h_2, \theta_1, \theta_2, k\}$
Output: The result returned by the smart contract ρ
Select $g, h, c, b_1, b_2, h_1, h_2, \theta_1, \theta_2 \in Z_p^*$
Compute:
$S_1 = b_1^c T_1^{s_1 - c_2{}^{\kappa_1}} / b_1^{s_2 - c_2{}^{\kappa_1}} Y_i^{s_3}$ and $S_2 = T_2^{s_1 - c_2{}^{\kappa_1}} / g^{s_3}$;
$S_3 = T_2^c g^{s_4}$ and $S_4 = T_3^c g^{s_1 - c_2{}^{\kappa_1}} h^{s_4}$;
Verify:
$c \stackrel{?}{=} H(g||h||Y_i||b_1||b_2||T_1||T_2||T_3||S_1||S_2||S_3||S_4)$;
return ρ: *true* or $false$;

where H() represents the hash function. After the verification is passed, it means that the receiver can obtain the secret of reconstruction and download the purchased image from IPFS after decrypting with its own private key.

4.3. Piracy Tracking

STTS completes the piracy tracking process using smart contracts. Pirates pretend the data are their own in order to make illegal income. The pirates can repackage the data items and publish them in the system. If the receiver searched for illegal data, they may send the request to the pirate and exchange keys, which can result in the detriment to legitimate participants. We designed two tracking methods, namely watermark information comparison and data item hash comparison, which are completed by intelligent contract.

Watermark information comparison refers to the sender finding that their data are pirated in the system or the Internet, extracting zero-watermark information after downloading the data, and then uploading them to the smart contract for comparison with the historical transaction records to determine the pirates and hold them accountable.

The data item check mechanism in the blockchain verifies the data and broadcasts the verification information. If the sender receives the verification information, it can make a complaint to the smart contract, which can quickly compare the ownership of the data. The data are divided into a number of data items, each of which generates a hash. After receiving the data, the receiver can select any part of the data and hash it, which is then spliced with the nonce hash of each transaction and sent to the smart contract for preservation. If the real producer receives the verification packet after using the subscription mechanism, the smart contract quickly performs hash comparison after sending the appeal to the smart contract.

In order to effectively pursue accountability, an appropriate reward and punishment mechanism should be established. We use smart contracts to deduct deposits for users who break the rules.

5. Security Analysis

Theory 1. The scheme satisfies unforgeability.

The first case is that the group manager can reconstruct the group key $S - l \in S$ by using the real reconstruction information A_{lj} through Equation (11). In the process of group key reconstruction, it can effectively prevent dishonest participant $U_j^{'}$ from forging $A_{lj}^{'}$, satisfying Equation (10). In order to forge $A_{lj}^{'}$ satisfying Equation (10), $U_j^{'}$ must first have the RSA private key d of TPs and then change Y_{ij} or r_i on the smart contract, for which it is impossible to modify the content on the blockchain. In addition, the content on the update contract can only be changed if the mutual verification is passed, and others can only view it, so $U_j^{'}$ cannot forge $A_{lj}^{'}$ that satisfies Equation (10).

In the second case, watermark information is modified, and zero-watermark information is embedded into high-definition multimedia data to embed copyright information without affecting data quality. Once watermark information is modified, data quality will decline, which means data will lose or be reduced in value. In our scheme, trading interests

are the default cause of piracy. The pirate will not modify the watermark information, because it has no meaning to them.

Theory 2. The scheme satisfies anonymity.

The environment used in this article is Ethereum, which is inherently anonymous. Secondly, we use the dual verification mechanism of group signature combined with secret sharing to ensure that the information of the participants remains invisible to each other. In the secret sharing scheme, even if the attacker acquires k secret shares, the secret value cannot be recovered. Moreover, the verification mechanism is added in the reconstruction process, which will make illegal reconstruction more difficult. Even if more than n shares are destroyed, the secret value can be recovered. In addition, in order to prevent the system from being affected by the breakdown of some group administrators, we set a limit on the minimum number of authentication devices during the verification process to ensure the high availability of the solution.

Theory 3. The scheme satisfies traceability.

First, the group administrator and TPs can obtain A_{lj} through Y_{lj}. Once a dispute occurs, they can appeal to TA through the key provided by the smart contract, and TA can calculate $A_{lj} = T_1/T_2{}^{Y_{lj}}$ to confirm the participants involved in the illegal behavior. Secondly, the smart contract ensures the effective comparison of copyright information when piracy occurs. All transaction information is stored in the world state of the smart contract, which cannot be modified and can be traced to the source. Once a dispute arises, the smart contract can quickly compare whether the copyright information is consistent.

6. Performance Analysis

This section is mainly divided into two parts, namely comparison with other schemes and performance evaluation. We looked for similar work and compared the algorithms in terms of time complexity and completeness. In addition, we tested the performance of the scheme in many aspects, which further demonstrates the feasibility of the scheme.

6.1. Comparison

Firstly, we compared with similar schemes in several aspects, including whether to use blockchain, anonymity, verifiable signature mechanism (two-stage verification), copyright protection and penalty mechanism. From Table 1, we can see that compared with the existing scheme, our scheme takes more factors into account, which makes our scheme more secure.

Table 1. Comparison of schemes in terms of completeness.

	Scheme [21]	**Scheme [22]**	**Scheme [23]**	**Scheme [24]**	**Our Scheme**
Blockchain	✓	✓	✓	✓	✓
Anonymous	✓	×	×	×	✓
Verifiable signing mechanism	×	×	×	×	✓
Copyright protection	✓	✓	✓	×	✓
Punishment mechanism	×	×	×	×	✓

Then, we analyze the differences in communication and verify computation between our scheme and similar schemes in Table 2. The verify computation here refers to the complexity required for the entire validation, regardless of the difference between one-way and two-stage, t represents the number of data blocks. Our solution reassembles the data after validation using casual transfers, which reduces the number of data blocks. In communication, although we use two-stage authentication, the complexity does not increase.

Table 2. Comparison of schemes in terms of verification process.

	Scheme [21]	**Scheme [22]**	**Scheme [23]**	**Scheme [24]**	**Our Scheme**
Communication	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
Verify computation	$tO(n)$	$tO(n)$	$tO(n)$	$tO(n)$	$O(n)$

6.2. Performance Evaluation

We set up an Ethereum environment on a Dell T7920 workstation, running memory of 32 G, network bandwidth of 100 M, using 4th generation Raspberry Pi for TPs simulation. We used Truffle + Ganache to write smart contracts and MetaMask for wallet management.

We first tested the performance of the smart contract in the scheme. Testing the performance of the smart contract in Ethereum is actually testing the performance of the function. Our scheme mainly includes five functions, which are storage (), key verification (), watermark hash comparison (), initiate transaction request (), data search () and penalty ().

Function storage () is used to store transaction information, data information and hash watermark and IPFS. Storage function is the basic functions of smart contracts that store ledger information into the state of the world for distributed ledger storage. In the test process, we found that the gas and time consumption of the three are almost the same, so it is described as a function.

Function Key verification () refers to the intelligent contract used by the group administrator and TPs in the process of group signature and secret sharing to exchange and verify public and private keys. This function completes the verification process described in Section 4.2. It should be noted that the smart contract is involved in the verification process on the chain, namely the verification of zero-knowledge proof results. The specific calculation process of zero-knowledge proof is carried out below the chain, so the verification function will not consume too much time and gas, which is suitable for IoT environment.

Function watermark hash comparison (): The hash comparison of watermark () refers to the similarity calculation of the hash value of the zero-knowledge watermark stored in the world state to obtain the appeal result when there is a piracy complaint. We use simhash in this function to calculate Hamming distance, and two hashes that meet a certain threshold will be considered similar.

Function Data search () means that the receiver can search for the data they want to buy in the blockchain, and the data information is stored by the sender through a smart contract. We use cosine similarity to compare search keywords, which is similar to hash. Keyword vectors that meet the threshold will be returned, and the smart contract displays similar result keywords.

Function Punishment () means that the smart contract can automatically deduct the deposit of the corresponding participant in the event of illegal behavior.

As shown in Figure 2, we built the Ethereum network through Truffle + Ganache + MetaMask and tested the gas and time consumption of the above five main functions. The test results show that the performance of the functions meets the basic requirements of the Ethereum network.

We then tested the time of two-stage verification based on the group signature and secret sharing scheme. The test subjects were group administrator and TP, as shown in Figure 3. We tested on different number of participants and different number of data blocks separately. When testing with different numbers of participants, the verification time of TP will increase with the increasing number of participants, but the time of group administrators does not increase significantly. This is because the number of group administrators is relatively fixed, but the number of TP can continuously increase according to the needs of the IoT, so the verification time of TP will increase linearly. On the contrary, as the number of data blocks increases, the validation time of both TP and group administrator will increase. For TP, the increase in data blocks means the complexity of casual transmission and the complexity of TP reassembly of data blocks will also increase. For

the group manager, an increase in the number of blocks means an increase in the number of TPS that are validated against him, so the group manager's validation time increases linearly. In general, the verification time is within a reasonable range, which can meet the communication time requirements of the Internet of Things.

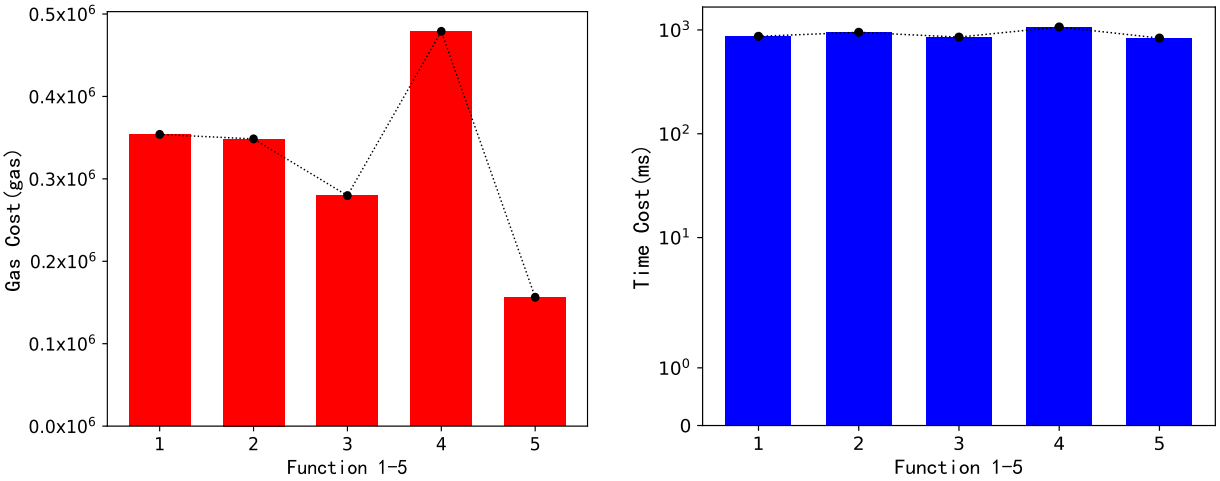

Figure 2. Gas and time consumption of the main functions.

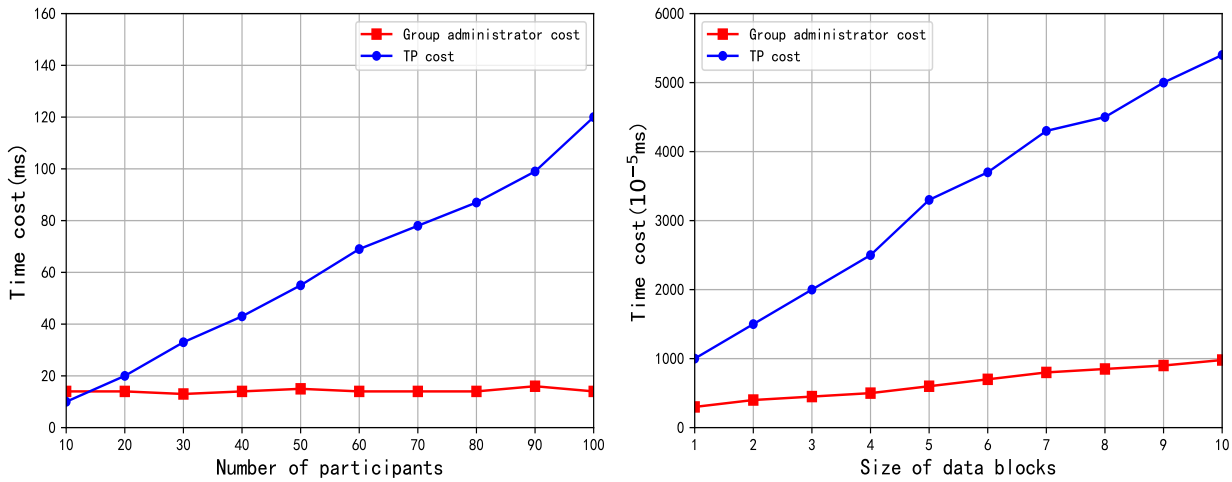

Figure 3. Time consumed by different participants and different data blocks.

We tested the split and refactoring time of the secret, testing the time consumption for 1–10 shares. According to Figure 4, we can know that the time of secret segmentation is within 1 s, which is reasonable, because the secret segmentation is run locally, and the time of secret reconstruction increases with the increase in the share. This is because the execution time of smart contract is also included in the verification process with the group administrator, so the time is longer than that of segmentation. We also tested the time to use smart contracts for accountability when piracy was detected. As the number of watermark hashes stored on the ledger increased, the time to compare also increased. Our scheme did not increase the time to compare 100 pieces of data too much, which we considered reasonable in smart contracts.

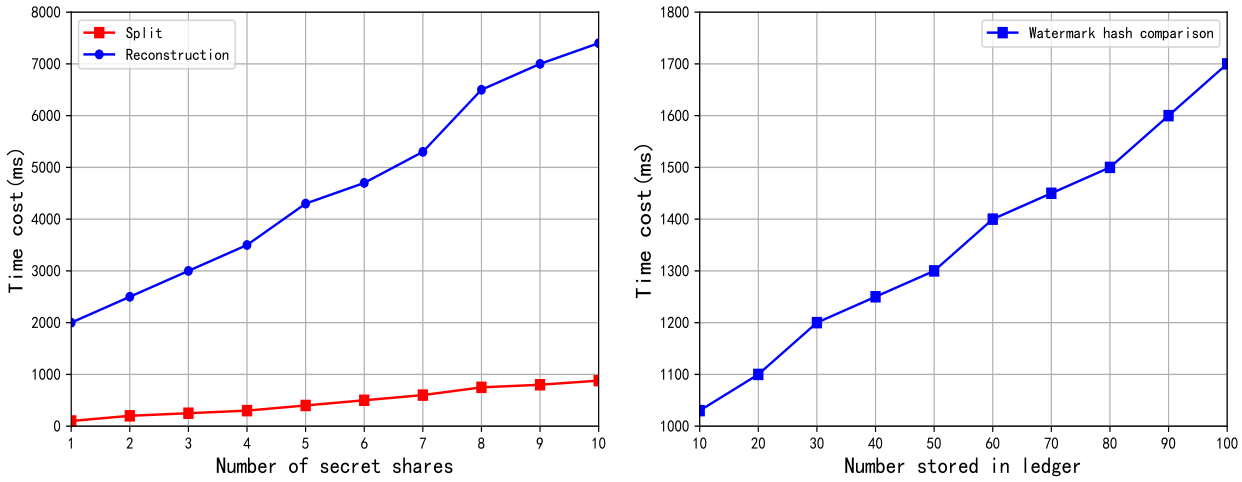

Figure 4. Time consumption of SSS and watermark hash comparison.

Finally, we tested the time consumption of oblivious transfer and the communication latency for different number of transactions with different numbers of devices. Figure 5 shows the results of our tests. The 1 out of 2 oblivious transfer is a cryptographic protocol that refers to the choice of one of two secrets, and the person who chooses has no way of knowing what he has chosen, thus guaranteeing trusted transmission. With the increasing number of devices, the computational complexity will also increase and so will the time; this is reasonable because as TPs increases, so does the complexity of the oblivious transfer (more blocks need to be generated, and the complexity of recombination increases), but because the operation takes place off-chain, it does not affect the timing of the transaction. Since the execution of the protocol is local, the communication efficiency of the transaction will not be affected. We also test the communication delay of search and storage under different transaction quantities, which are almost consistent with the ordinary function delay, indicating that the increase in transaction quantity does not have a great impact on the communication delay, indicating that the scheme is feasible.

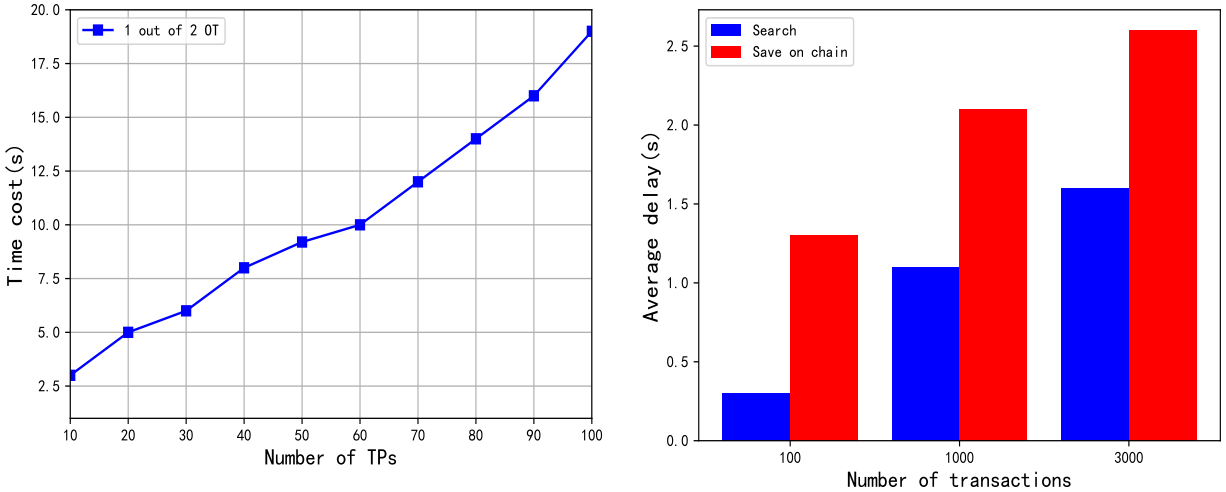

Figure 5. Time consumption of 1 out of 2 OT and transaction delay.

7. Discussion and Limitations

Due to performance limitations, our scheme is not the approach for all uses. For example, Tomasz proposed a smart contract of similar interface in his study [37], which realizes the reuse and security of smart contract by using trading rules and DApp. This work is very suitable for large-scale application scenarios, especially in the cloud–edge collaborative Internet of Things environment, where distributed IoT nodes can quickly invoke smart contracts while using the transaction rules designed by the scheme to ensure security. However, the performance of the scheme designed in the work has not been intuitively demonstrated, so it is impossible to understand the practicability of the scheme. There is also a lot of other work going on with blockchain and smart contracts, and just like our solution, a personalized way needs to be developed based on scenarios while taking into account performance metrics and time complexity.

8. Conclusions

In this paper, we propose a secure and traceable transaction scheme for high-definition multimedia data based on blockchain. We use oblivious transfer combined with zero watermarking to achieve the secure embedding of copyright information. We then use a group signature and secret sharing scheme to achieve the trusted distribution of data. We explained the procedure and logic of the scheme through detailed description of the scheme and algorithm and further demonstrated the feasibility and efficiency of our scheme through comparison of experiments and schemes. The results of simulation experiments showed that the scheme also has a certain application value in the multi-device Internet of Things. In future work, we will fully combine the trusted execution environment SGX enclave to ensure the security of smart contract. Specifically, we will put the calculation with high complexity into the enclave for execution and use RSA encryption mode to securely transmit the calculation results. One can also securely block an enclave in order to control the calling rights of the smart contract. In addition, we believe it makes more sense to use non-fungible tokens (NFTs) for copyright protection because NFTs are in the system from the creation of the work, the entire transaction process is recorded through smart contracts, and each work has a unique ID that makes it impossible to pirate.

Author Contributions: Conceptualization, S.Z. and J.P.; Methodology, S.Z. and F.Y.; Software, F.Y.; Validation, S.Z. and Z.Z.; Formal analysis, Z.Z.; Investigation, Z.Z.; Writing—original draft, J.P.; Supervision, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work is funded in part by the National Natural Science Foundation-funded Project of China (No.62062016, No.U21A20474), Guangxi Natural Science Foundation-funded Project (No.2019JJA170060), Jiangsu Provincial Key Laboratory of Network and Information Security under Grants No. BM2003201, Guangxi Science and technology project (GuikeAA22067070 and GuikeAD21220114). Finally, we thank the Guangxi "Bagui Scholar" Teams for Innovation and Research Project, Center for Applied Mathematics of Guangxi (Guangxi Normal University), the Guangxi Talent Highland Project of Big Data Intelligence and Application.

Data Availability Statement: No data were used to support this study

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

References

1. Tang, Z.; Chen, L.; Zhang, X.; Zhang, S. Robust Image Hashing with Tensor Decomposition. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 549–560. [CrossRef] [CrossRef]
2. Tang, Z.; Zhang, X.; Li, X.; Zhang, S. Robust image hashing with ring partition and invariant vector distance. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 200–214. [CrossRef] [CrossRef]
3. Wang, L.; Liu, X.; Lin, X. A Fair and Privacy-Preserving Image Trading System Based on Blockchain and Group Signature. *Secur. Commun. Netw.* **2021**, *2021*, 1–18. [CrossRef] [CrossRef]
4. Lyu, F.; Cheng, N.; Zhu, H.; Zhou, H.; Xu, W.; Li, M.; Shen, X.S. Intelligent context-aware communication paradigm design for IoVs based on data analytics. *IEEE Netw.* **2018**, *32*, 74–82. [CrossRef] [CrossRef]

5. Liang, J.; Qin, Z.; Xiao, S.; Ou, L.; Lin, X. Efficient and secure decision tree classification for cloud-assisted online diagnosis services. *IEEE Trans. Dependable Secure Comput.* **2019**, *18*, 1632–1644. [CrossRef] [CrossRef]
6. Dutta, P.; Choi, T.M.; Somani, S.; Butala, R. Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transp. Res. Part e Logist. Transp. Rev.* **2020**, *142*, 102067. [CrossRef] [CrossRef] [PubMed]
7. Mihardjo, L.; Sasmoko, S.; Alamsjah, F.; Elidjen, E. The influence of digital customer experience and electronic word of mouth on brand image and supply chain sustainable performance. *Uncertain Supply Chain. Manag.* **2019**, *7*, 691–702. [CrossRef] [CrossRef]
8. Cao, L.J.; Men, C.G.; Gao, Y. A recursive embedding algorithm towards lossless 2D vector map watermarking. *Digit. Signal Process.* **2013**, *23*, 912–918. [CrossRef] [CrossRef]
9. Zhao, S.; O'Mahony, D. Bmcprotector: A blockchain and smart contract based application for music copyright protection. In Proceedings of the 2018 International Conference on Blockchain Technology and Application, Xi'an, China, 10–12 December 2018; pp. 1–5. [CrossRef]
10. Xiao, L.; Huang, W.; Xie, Y.; Xiao, W.; Li, K.C. A blockchain-based traceable IP copyright protection algorithm. *IEEE Access* **2020**, *8*, 49532–49542. [CrossRef] [CrossRef]
11. Meng, Z.; Morizumi, T.; Miyata, S.; Kinoshita, H. Design scheme of copyright management system based on digital watermarking and blockchain. In Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), Tokyo, Japan, 23–27 July 2018; Volume 2, pp. 359–364. [CrossRef]
12. Savelyev, A. Copyright in the blockchain era: Promises and challenges. *Comput. Law Secur. Rev.* **2018**, *34*, 550–561. [CrossRef] [CrossRef]
13. Ren, N.; Zhao, Y.; Zhu, C.; Zhou, Q.; Xu, D. Copyright protection based on zero watermarking and blockchain for vector maps. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 294. [CrossRef] [CrossRef]
14. Wang, X.; Shao, C.; Xu, X.; Niu, X. Reversible data-hiding scheme for 2-d vector maps based on difference expansion. *IEEE Trans. Inf. Sec.* **2007**, *2*, 311–320. [CrossRef] [CrossRef]
15. Sun, J.; Zhang, G.; Yao, A.; Wu, J. Lossless Digital Watermarking Technology for Vector Maps. *Acta Electron. Sin.* **2010**, *38*, 2786–2790. [CrossRef] [CrossRef]
16. Zhao, H.; Du, S.; Zhang, D. Zero-Watermark Scheme for 2D Vector Drawings Based on Mapping. In Proceedings of the 2011 IEEE 12th International Conference on Computer-Aided Industrial Design & Conceptual Design, Vols 1 and 2: New Engines for Industrial Design: Intelligence-Interaction-Services, New York, NY, USA, 27–29 November 2011; pp. 366–370. [CrossRef]
17. Huang, C.; Liu, D.; Ni, J.; Lu, R.; Shen, X. Achieving Accountable and Efficient Data Sharing in Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2020**, *17*, 1416–1427. [CrossRef] [CrossRef]
18. Zou, W.; Lo, D.; Kochhar, P.S.; Le, X.B.D.; Xia, X.; Feng, Y.; Chen, Z.; Xu, B. Smart contract development: Challenges and opportunities. *IEEE Trans. Softw. Eng.* **2019**, *47*, 2084–2106. [CrossRef] [CrossRef]
19. Sun, X.; Zhou, J. Deep Perceptual Hash Based on Hash Center for Image Copyright Protection. *IEEE Access* **2022**, *10*, 120551–120562. [CrossRef] [CrossRef]
20. Holland, M.; Nigischer, C.; Stjepandić, J. Copyright protection in additive manufacturing with blockchain approach. In *Transdisciplinary Engineering: A Paradigm Shift*; IOS Press: Amsterdam, The Netherlands, 2017; pp. 914–921. [CrossRef]
21. Huang, H.; Chen, X.; Wang, J. Blockchain-based multiple groups data sharing with anonymity and traceability. *Sci. China Inf. Sci.* **2020**, *63*, 1–13. [CrossRef] [CrossRef]
22. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for digital rights management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef] [CrossRef]
23. Wang, B.; Jiawei, S.; Wang, W.; Zhao, P. A blockchain-based system for secure image protection using zero-watermark. In Proceedings of the 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Delhi, India, 10–13 December 2020; pp. 62–70. [CrossRef]
24. Mangipudi, E.V.; Rao, K.; Clark, J.; Kate, A. Towards automatically penalizing multimedia breaches. In Proceedings of the 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS & PW), Stockholm, Sweden, 17–19 June 2019; pp. 340–346. [CrossRef]
25. Han, J.; Zhang, Y.; Liu, J.; Li, Z.; Xian, M.; Wang, H.; Mao, F.; Chen, Y. A Blockchain-Based and SGX-Enabled Access Control Framework for IoT. *Electronics* **2022**, *11*, 2710. [CrossRef] [CrossRef]
26. Fang, M.; Zhang, Z.; Jin, C.; Zhou, A. High-performance smart contracts concurrent execution for permissioned blockchain using SGX. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greecw, 19–22 April 2021; pp. 1907–1912. [CrossRef]
27. Rabin, M.O. Transaction protection by beacons. *J. Comput. Syst. Sci.* **1983**, *27*, 256–267. [CrossRef] [CrossRef]
28. Fang, W.S.; Wu, L.L.; Zhang, R. A watermark preprocessing algorithm based on arnold transformation and logistic chaotic map. In *Advanced Materials Research*; Trans Tech Publications: Zurich, Switzerland, 2012; Volume 341, pp. 720–724. [CrossRef]
29. Yang, C.C.; Chang, T.Y.; Hwang, M.S. A (t, n) multi-secret sharing scheme. *Appl. Math. Comput.* **2004**, *151*, 483–490. [CrossRef] [CrossRef]
30. Boneh, D.; Boyen, X.; Shacham, H. Short group signatures. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2004; pp. 41–55. [CrossRef]

31. Ateniese, G.; Camenisch, J.; Joye, M.; Tsudik, G. A practical and provably secure coalition-resistant group signature scheme. In Proceedings of the Annual International Cryptology Conference, Kyoto, Japan, 3–7 December 2000; Springer: Kyoto, Japan, 2000; pp. 255–270. [CrossRef]
32. Nakamoto, S. A Peer-to-Peer Electronic Cash System. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 10 April 2021).
33. Zheng, Z.B.; Xie, S.A.; Dai, H.N.; Chen, X.P.; Wang, H.M. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564. [CrossRef]
34. Atzei, N.; Bartoletti, M.; Cimoli, T. A Survey of Attacks on Ethereum Smart Contracts (SoK). *Lect. Notes Comput. Sci.* **2017**, *10204*, 164–186. [CrossRef]
35. 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] [CrossRef]
36. Yang, J.; Hu, K.; Wang, X.; Wang, H.; Liu, Q.; Mao, Y. An efficient and robust zero watermarking algorithm. *Multimed. Tools Appl.* **2022**, *81*, 20127–20145. [CrossRef] [CrossRef]
37. Górski, T. Reconfigurable Smart Contracts for Renewable Energy Exchange with Re-Use of Verification Rules. *Appl. Sci.* **2022**, *12*, 5339. [CrossRef] [CrossRef]