Future Generation Computer Systems 89 (2018) 746–764

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Blockchain for digital rights management

Zhaofeng Ma [a,b,*], Ming Jiang [c], Hongmin Gao [a,b], Zhen Wang [a,b]

[a] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] Information Security Center, Beijing University of Posts and Telecommunications, Beijing 100876, China
[c] The Third Research Institute of China Electronics Technology Group Corporation, 100015, Beijing, China

HIGHLIGHTS

- We proposed a new trusted model DRMChain for digital rights management based on blockchain.
- The DRMChain builds up an external flexible storage and internal blocks creation architecture.
- The DRMChain provides a DRM-protected scheme supporting for identity and privacy protection.
- The DRMChain innovates a violation tracing approach with conditional identity management.

ARTICLE INFO

Article history:
Received 25 January 2018
Received in revised form 10 July 2018
Accepted 14 July 2018
Available online 23 July 2018

Keywords:
Digital rights management
Blockchain
Content protection
Privacy protection
Conditional tracing
Violation checkout

ABSTRACT

Online digital content service becomes more and more easily, however, free consumption and excessive spreading without rights protection will hurt the content providers' benefits and causes business loss, another problem is once the content provider supply illegal or politically sensitive content such as terroristic opinion or multimedia content, it will cause serious social problem such as fright or social crisis. To solve this problem, in this paper we proposed a blockchain-based scheme for digital rights management(named DRMChain), which supports the right content serves the right users in a right way, the DRMChain can provide trusted and high-level credible content protection and conditional traceability of violation content service. In the proposed DRMChain, we use two isolated blockchain application interfaces (BAI) to respectively store plain and cipher summary information of original and DRM-protected digital content, and considering large capacity of digital content such as image, audio or video, we proposed external flexible storage of plain/cipher digital content and creates hashID of the content itself and links with the blockchain. In DRMChain scheme, we named the BAI plain interface as BAIP for summary metadata storage of original content, and the BAI cipher interface as BAIC for DRM-protected content service. In the DRMChain scheme we proposed efficient and secure authentication, privacy protection and multi-signature-based conditional traceability approaches, and thus the DRM license, usage control and constrain information can be easily retrieved form the blockchain, and customs can query all the consumption transaction lists of free or paid consumption history to prevent baleful fee-deduction. Analysis and performance evaluation manifest the DRMChain scheme provides a reliable, secure, efficient and tamper-resistance digital content service and DRM practice.

1. Introduction

Digital content consumption is now becoming popular, more and more people often visit and watch videos or images resource through web browser or mobile App-based software. However illegal content usage (such as illegal download and spread the right-reserved content) may do harm to content providers, or hurt the right-holder's business stakeholder [1–4], upon the value-added content or business data, it is necessary to use technique solutions to prevent the data being stolen or being illegally used, and together should enhance the usage control of content access. In fact, digital rights management [5–9] is an important technology for content protection of rights holder's profits or business stakeholder [1–3,6,9], upon which many institutes and researchers paid much attention and do more research work on DRM [1–5,10–20], however, current DRM technologies such Windows DRM, Silverlight, RealNetworks, Flash AIR, Apple HLS DRM focused on content encryption and license management, however it is obviously lack of original content management violation checking and tracing of the one who should responsible for the violation [12–16].

* Corresponding author at: School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China.
E-mail address: mzf@bupt.edu.cn (Z. Ma).

https://doi.org/10.1016/j.future.2018.07.029
0167-739X/

Upon the above problems, new DRM architecture should require efficient and reliable technologies that can provide creditable, tamper-resistant and high-level secure and flexible supporting [1–6]. Fortunately, blockchain is a decentralized, reliable and secure computing paradigm in P2P network environment [21–24], which provides distributed ledger technology (DLT) that store the completed blocks in chronological order with tamper-resistance and security, it allows participants to keep track of digital transactions without central recordkeeping, Each node holds a copy of the blockchain downloaded automatically, the record's authenticity can be verified by the entire community using the blockchain instead of a single centralized system [25–28]. Blockchain can be applied for IT asset management and supply chain management, trademarks copyrights protection, credit certificate proof [29–40]. The most famous and successful practice and applications of blockchain are Bitcoin [21,22], Ethereum [23], Hyperledger [24] et al.

As for recent research on blockchain [21–40], Wright A. summarized decentralized blockchain technology and in the future creation of the Internet, which has the potential to decentralize data management [34]. Zyskind G. proposed decentralizing privacy protection method which used blockchain to protect personal data [35]. Kosba A. E. [36] studied blockchain model of cryptography and privacy-preserving using smart contracts, in which a decentralized smart contract system that does not store financial transactions in the clear on the blockchain, thus retaining transactional privacy from the public's view. Ao Lei [37] et al. proposed a framework for providing secure key management within the heterogeneous network. The security managers (SMs) play a key role in the framework by capturing the vehicle departure information, encapsulating block to transport keys and then executing rekeying to vehicles within the same security domain. M Vukolić [38] studied scalable blockchain fabric which compared the consensus Proof-of-Work vs. BFT Replication, and also discuss recent proposals to overcoming these scalability limits and outline key outstanding open problems in the quest for the "ultimate" blockchain fabric(s). Ali Dorri et al. [39] studied blockChain from cryptocurrencies to smart contracts, and then propose a blockchain-based architecture to protect the privacy of the users and to increase the security of the vehicular ecosystem. Remo Manuel Frey et al. [40] focused on the effect of a blockchain-supported, privacy-preserving system on disclosure of personal data from a psychological perspective.

To solve the security and reliability of the digital rights management, in this paper we proposed a blockchain-based scheme for digital rights management (named DRMChain), which supports the right content serves the right users in a right way, the DRMChain can provide trusted and high-level credible content protection and conditional traceability of violation content service. In the proposed DRMChain, we use two isolated blockchain application interfaces (BAI) to respectively store plain and cipher summary information of original and DRM-protected digital content, and considering large capacity of digital content such as image, audio or video, we proposed external flexible storage of plain/cipher digital content and creates hashID of the content itself and links with the blockchain, in which the DRMChain has the following advantages and novelty:

(1) We proposed a new trusted model DRMChain for digital rights management based on blockchain.

(2) The DRMChain builds up an external flexible storage and internal blocks creation architecture.

(3) The DRMChain provides a DRM-protected scheme supporting for identity and privacy protection.

(4) The DRMChain innovates a violation tracing approach with conditional identity management.

2. DRM requirement and suitability

2.1. DRM security requirement

In traditional case, DRM only considers how to protect the content from being illegally used such as consumed the content without licensing or payment, however once the content is encrypted, it gets difficult to audit the content especially when the content include illegal, sexual or bloodcurdling material. In this paper, we proposed new paradigm of the DRM for content protection in an open and credible platform for DRM services such as provide content consumption, licensing purchase rather than in a private website. The new security and requirements include: (1) content verifiability and tamper-resistance, (2) identity management and privacy protection of content provider, (3) Content protection, (4) Usage control, (5) Licensing, (6) Violation tracking.

2.1.1. Content verifiability and tamper-resistance

Before the content is uploaded the open and credible content platform, the content platform requires that the content source is verifiable and content is auditable and can find who should responsible for the content once the content is viewed as illegal that is the content is verifiable for auditing. Once the content is uploaded in the open platform it should be stored as evidence and should be tamper-resistant.

2.1.2. Identity management and privacy protection

A good DRM scheme should have the attributes that can ensure the user's privacy, and meanwhile can identify the user when he/she uploads or spreads illegal, ethical or political-related content. And together it is important to prevent the internal administrator from leaking users' identity data or privacy.

2.1.3. Content protection

Before the content provides service to public, it is necessary to protect the content from being freely used or spread, content encryption [1–4] is used to prevent the media being freely used, and watermarking [10–15] is usually adopted for content right tracing or confirmation.

2.1.4. Usage control

Once the content is protected by encryption or watermarking approach, it should include abundant usage control rules such as constraints and conditions for content consumption.

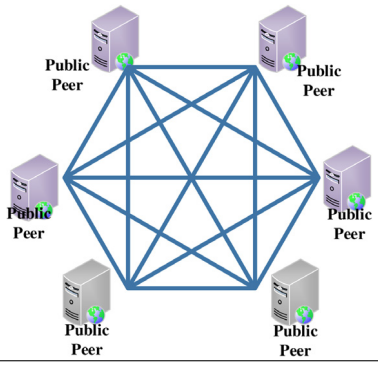
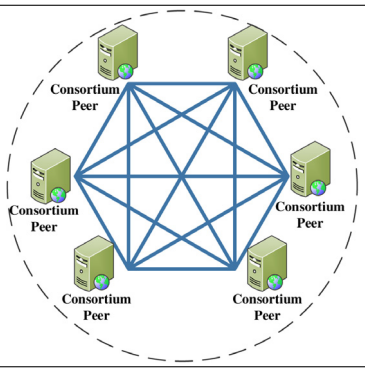
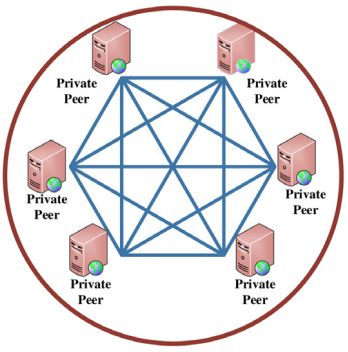
2.1.5. DRM licensing

When public users consume the protected content, he/she first buy or get its license for usage such as reading, listening, or playing the content. The license declares the basic rights such as usage times, period, domain, rental, translation or compilation, or watermark that defined the ownership of the content.

2.1.6. Violation tracking

During the consumption, when the content is considered including illegal ownership violation or the content including sensitive information or opinion or illegal data, thus the platform administrator then can track who should responsible for the content and trace the identity of the content provider.

Table 1
Comparison of different blockchains.

Item	Public blockchain	Consortium blockchain	Private blockchain
Topology			
User range	All public peer can join in the public Node	Only Authorized Organization or team can join in the Consortium P2P Blockchain Network.	Only authorized private peer such as an enterprise or organization can access the network.
Node rights	All public peer has the equal rights such as read, write, execute.	All the operation such as write, read and query must obey the access control policy.	The access and behavior is only open the private node.
Attribution	The public peer can access the blockchain anonymously and the data and info are public to all	The consortium blockchain can support real identity and behavior and data auditing (AML/KYC).	Private rights and high security, but limited usage value.
Trans rate (times/s)	7–15	1000	More than 1000

2.2. Blockchain suitability for DRM

As for the blockchain classification [27–35], there are public blockchain, consortium blockchain and private blockchain. The comparison of each blockchain is listed in Table 1. Upon the digital rights management requirement, considering the large capacity of multimedia content storage such as image, audio or video, the suitable framework of the blockchain is "building up blocks in internal blockchain platform, but storing the content itself in external database". The DRMChain scheme proposed efficient and secure authentication, privacy protection and multi-signature-based conditional traceability approaches, and thus the DRM license, usage control and constrain information can be easily retrieved form the blockchain, and customs can query all the consumption transaction lists of free or paid consumption history.

Upon the digital rights management requirement, the blockchain should only be used for authorized or multipart administrator to manage the content in a credible and tamper-resistant mode, which can provide trusted content violation traceability, in which the reading, writing or auditing operation must obey the access control policy. Thus according to the above analysis in this paper we select consortium blockchain for the digital rights management, which is used to store the original content source for tamper-resistant evidence and violation tracing, then the content itself, the content ownership, rights holder, content obligation, constraints, obligation and security requirements can be included in the consortium blockchain for detailed and authorization operation.

3. DRMChain: blockchain-based scheme for digital rights management

3.1. The proposed DRMChain scheme

In this paper we proposed a blockchain-based scheme for digital rights management (named DRMChain), which supports the right digital rights-protected content serves the right users in a right way, the RightChain can provide trusted and high-level credible content protection and conditional traceability of violation content service. In the proposed DRMChain, we use two isolated blockchain application interfaces (BAI) to respectively store plain and cipher summary information of original and DRM-protected digital content, and considering large capacity of digital content such as image, audio or video, we proposed external flexible storage of plain/cipher digital content and creates hashID of the content itself and links with the blockchain. In DRMChain scheme, we named the BAI plain interface as BAIP for summary metadata storage of original content, and the BAI cipher interface as BAIC for DRM-protected content service. In the DRMChain scheme we proposed efficient and secure authentication, privacy protection and multi-signature-based conditional traceability approaches, and thus the DRM license, usage control and constrain information can be easily retrieved form the blockchain, and customs can query all the consumption transaction lists of free or paid consumption history to prevent baleful fee-deduction. We implemented the DRMChain platform for digital right management in the based on Ethereum and IPFS P2P storage, performance evaluations manifest the DRM-Chain is reliable, secure, efficient and tamper-resistance with high-level credibility, in which the authorization users can upload their right-reserved digital content, but once the content is suspected illegal or rights infringement, the DRMChain can trace and checkout the violation content and provider user, the DRMChain provides a reliable and tamper-resistant DRM practice and can apply in many fields. Analysis and performance evaluation manifest the DRM-Chain scheme provides a reliable, secure, efficient and tamper-resistance digital content service and DRM practice.

3.2. The DRMChain trusted model

3.2.1. The DRMChain external IPFS storage

In the DRMChain scheme, before the content is provided for business consumption, the content provides original metadata plaintext data, and stores the metadata in blockchain p2p network, which is strictly limited for access or data obtain, which will be the original content as raw data for DRM processing and original evidence for possible auditing and checking. The most important advantage of the proposed scheme is to adopt blockchain for sensitive and tamper-resistant data storage, once the data was storage into the blockchain it will permanently be stored in the P2P network and cannot modified or delete, which can provide

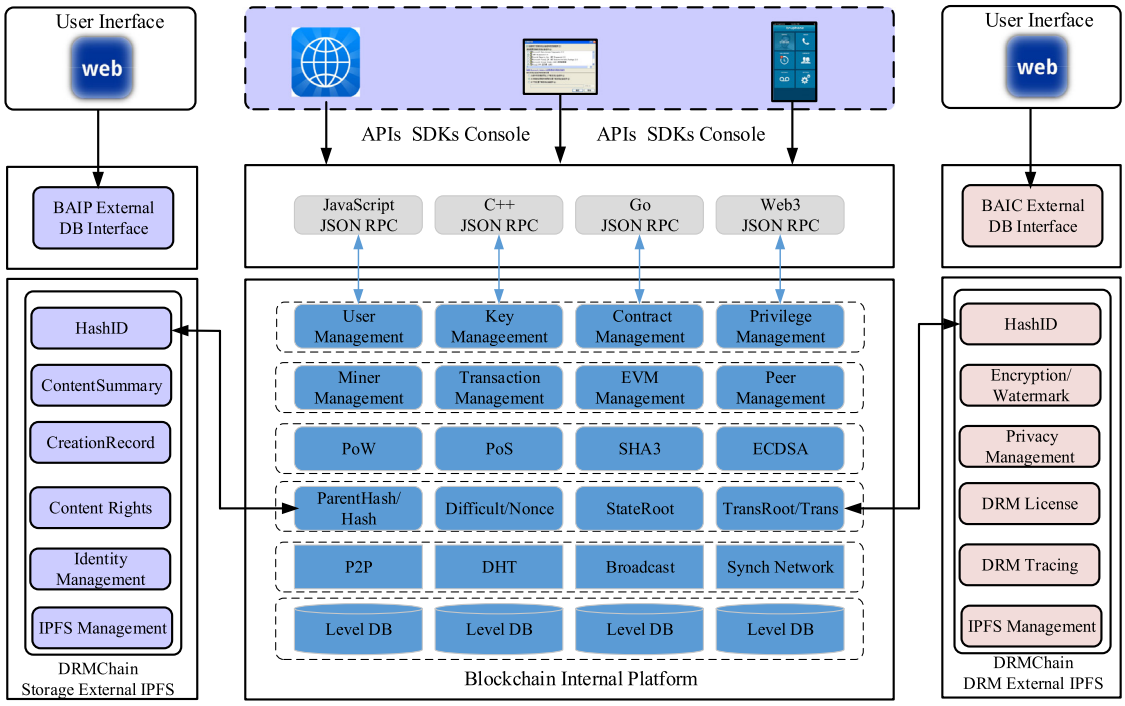


Fig. 1. The DRMChain trusted model.

strong and high level reliability and security. Even some blockchain nodes deliberately announced or the nodes truly corrupted, the other nodes can provide strong and trusted service for evidences and business-related service. While the large amount data can be stored in external IPFS, and the DRMChain platform can provide trusted and tamper-resistance transaction data confirmation with unique block number and transactionID, by which user can query the transaction data by block number, block hash, transaction hash in the blockchain platform, and by the IPFS hashID user can query content summary, creationRecord, ContentRight, and identity information in external IPFS, The DRMChain trusted model is described as Fig. 1.

3.2.2. The DRMChain blockchain platform

For the constraint of the amount of multimedia content, it is not suitable for store full multimedia content in the blockchain platform in the DRMChain scheme, the original plain content is hashed and stored in the DRMChain external IPFS p2p network, which can retrieve all the original information of the digital content, and the DRM-protected information is stored information in the DRMChain external IPFS network, which can provide DRM service and security management, both of the original plain content and DRM-protected content are respectively related by the content hashID, and linked with the hashID, the content summary can be stored in the blockchain for permanent, reliable and secure data service.

3.3. The DRMChain external ipfs DRM

3.3.1. DRMChain identity and privacy management

To protect the core privilege and rights of content provider, and for possible violation in future usage and service, the DRMChain scheme require effective and verifiable identity authentication, and collect basic and critical information of content provider. In another side, because the scheme collects content provider's identity information, in the proposed DRMChain, we proposed an entire and secure approach to protect user's privacy.

3.3.2. Content protection processing for DRM service

To ensure the security and availability of the data encryption and authentication of scheme, we proposed efficient key agreement protocol for secure communication between client users and blockchain nodes in DRMChain, and develop a mater/slavery key management for content encryption. Before providing content service for consumption, the DRMChain scheme encrypts the content, and then provides policy configuration, license management, and usage control for independent users. Then public customers can get the ciphered content and achieves the DRM services from the DRMChain platform for business benefits.

3.3.3. Violation tracing

Once the content provider supplies illegal or politically sensitive content such as terroristic opinion or multimedia content, the DRMChain will trace the content source and check the original content, and identifies the real identity and deals with the content, and give corresponding punishment according to the violation level, such as delete the DRM service content, or forbidden the content provider from upload content again, or close the content provider's account for service.

4. Security infrastructure of DRMChain

4.1. The elliptic curve cryptosystems [41–43]

An elliptic curve E defined over F_q is a set of points $P = (x_p, y_p)$ where x_p and y_p are elements of F_q that satisfy a certain equation, if $q = p$ is an odd prime and $p > 3$, then a and b shall satisfy $4a^3 + 27b^2 \neq 0 (\mathrm{mod}\, p)$, and every point $P = (x_p, y_p)$ on E (other than the point 2) shall satisfy the equation in F_p: ${y_p}^2 = {x_p}^3 + ax_p + b$. For further background of the case that $q = 2^m$ and other details on elliptic curves, see [41–43].

Supposing that GF(p) is a finite field with characters $p \neq 2,3$, for a, b $\in$ GF(P) where $4a^3 + 27b^2 \neq 0 \ (\mathrm{mod}\, p)$. Elliptic Curve $E_{(a,b)}(\mathrm{GF(p)})$ in GF(p) is defined as the point set (x,y) $\in$ GF(p) $\times$ GF(p) that satisfies the equation $y^2 = x^3 + ax + b$, where the infinite point O is included in $E_{(a,b)}(\mathrm{GF(p)})$. All points in GF(p) is an

Abelian group, where the identical element is O. Supposing P and Q are points in $E_{(a,b)}(GF(p))$, if P = O, then $-P = O$, $P + (-P) = O$; denote P = (x_1, y_1), Q(x_2, Y_2), then −P = (x_1, y_1), and P +(−P) = O, if Q ≠ −P, $P + Q = (x_3, y_3)$, where $P + Q = (x_3, y_3)$

$$x_3 = u^2 - x_1 - x_2 \tag{1}$$

$$y_3 = u(x_1 - x_3) - y_1 \tag{2}$$

$$u = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1} & P = Q \end{cases} \tag{3}$$

4.2. Elliptic curve parameter

Elliptic curve domain parameters over F_p consists of the following parameters:

(1) A field size $q = p$ that defines the underlying finite field F_q, where $p > 3$ should be a prime.

(2) If the elliptic curve was randomly generated, a bit string SEED with length at least 160 bits is needed (this is the Optional parameters).

(3) Two field parameters a and b in F_q which is used to define the equation of the elliptic curve E:

$$y^2 = x^3 + ax + b \tag{4}$$

(4) A point $G = (x_G, y_G)$ of prime order on E, *where* $G \neq 0$ is a must condition.

(5) The order n of the point G, should be satisfied $n > 2^{160}$ and $n > 4\sqrt{q}$;

(6) The cofactor $h = \#E(F_q)/n$ *is* an optional parameter.

For convenience, Elliptic curve domain parameters over F_p can be written as:

$$P_{ECC} = (q, FR, a, b, G, n, h) \tag{5}$$

4.3. Content symmetric encryption

In the DRMChain scheme, the content protection we use symmetric encryption AES algorithm for content encryption, and adopt hash algorithm SHA1 for digest algorithm, and ECDSA for digital signature [41–47]. The Keccak-256 hash function (as per the winning entry to the SHA-3 contest) is denoted KEC (and generally referred to as plain Keccak).

4.4. Algorithms in blockchain

As for the blockchain itself [21–25], the platform is based on double SHA256, SHA3, RIPEMD160 and ECC-based public cryptosystem ECDSA. Especially the public address of most blockchain is based on BASE58. The ECC algorithm in blockchain system uses secp256k1 elliptic curve which is different to the ECDSA parameters. The Secp256k1 elliptic curve is defined by the 6 parameters D = (p,a,b,G,n,h) where

P = $2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^{6-} - 2^{4-1}$
= FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
FFFFFFFF FFFFFFFF FFFFFFFF FFFFFC2F

A = 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000

b = 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000007

Compressed base point G:

G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB
2DCE28D9 59F2815B 16F81798

Non-compressed base point G:

G = 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB
2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465
5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F
FB10D4B8

The order of G:

n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6
AF48A03B BFD25E8C D0364141

Co-factor:

$h = 01$

4.5. Blockchain consensus mechanism

The main consensus mechanisms include proof of work (PoW), proof of stack (PoS), distributed POS (DPoS) and practical Byzantine Fault Tolerance (PBFT), other consensus mechanisms include Paxos, Raft, and Ripple, which can satisfy different applications of blockchain scenes.

5. Blockchain-based DRM data management

In the DRMChain scheme, the original DRM metadata include 3 kinds of core information, Content Metadata; CreationRecord metadata; TransforRight metadata. et al., in which the DRM metadata describes basic and core content description, ownership, rights, license, obligation and constraints. The early metadata is expressed as XrML, but later, the XrML is viewed as low efficient way in implementation with SAX or DOM parsers. To improve the efficiency and universality, in our proposed DRMChain scheme, we using JSON format as interface to store the DRM metadata, in which the DRMChain scheme can build up the blocks in time-order with tamper-resistance and security, by which once the providers or issuers published illegal or improper content such as political, religious or ethical. In DRMChain we can easily trace who should responsible for the violation.

When the DRM-related data is verified and confirmed in the blockchain, then the DRMChain block data structure and transaction are created as Fig. 2, in which the blockchain header includes difficulty, extraData, gasLimit, gasUsed, hash, number, timestamp, transactions, transactionsRoot and uncles, the users can query the blockchain by hash, number, timestamp, transactions, transactionsRoot to get all the information of the blockchain data. Once the block data is created, each transaction status in p2p-based network is transferred from one state to another, the DRMChain transaction status transfer is detailed in Fig. 3, in which each transaction can be created a unique transactionID, in which the transaction data itself can be plain or cipher-mode according to the service requirement.

In the DRMChain scheme, before the content is provided for business consumption for benefits, the content provider should store original plaintext content in the blockchain p2p network, which is strictly limited for access, write, or data obtain, and will be the original content as raw data for DRM processing and original evidence for possible auditing and checking of violation.

The blockchain will permanently stores the DRM data in the P2P network and cannot allow it be modified or deleted in a tamper-resistance mode, which can provide strong and high level reliability and security. Even some blockchain nodes deliberately

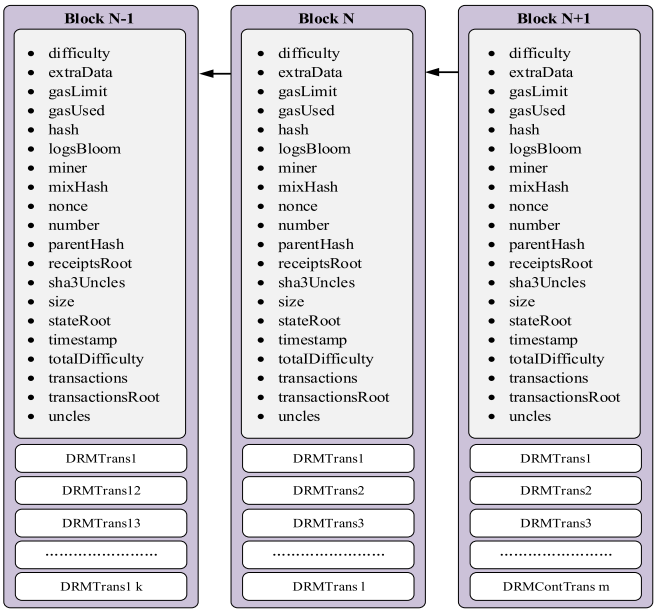


Fig. 2. The DRMChain data structure.

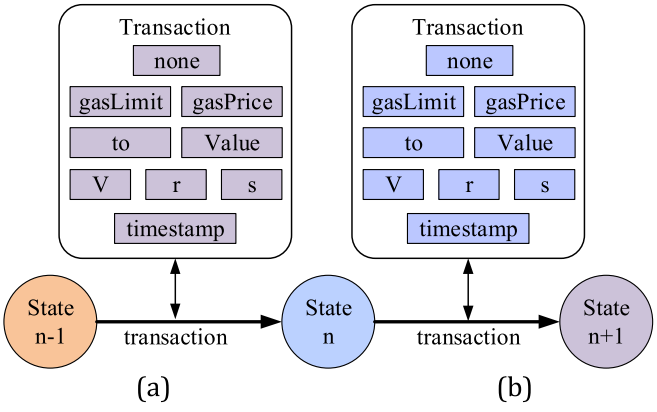


Fig. 3. The DRMChain transaction status transfer.

announced or the nodes truly corrupted, the most other nodes can provide strong and trusted service for evidences. The DRM metadata structure is defined as bellows which is easy to convert to JSON format. The DRMChain metadata [1–4,13] is described in Fig. 4.

5.1. DRMChain communication key agreement protocol

In the DRMChain scheme, we should ensure the communication security that prevents the communication is hijacked by attackers. In fact, the Diffie–Hellman key exchange algorithm [48] provides a mechanism which allows two parties to agree on a shared value without requiring encryption. However, it cannot resistant replay attack, man-in-the-middle attack and et al. [49], to improve the security of Diffie–Hellman key exchange algorithm, The OAKLEY protocol [50] is used to establish a shared key with an assigned identifier and associated authenticated identities for the two parties, in which two authenticated parties can agree on secure and secret keying material, both Diffie–Hellman key exchange algorithm and the OAKLEY protocol are based on Discrete logarithm problem (DLP), in current computing environment, based on the two protocol we proposed a more secure and efficient key exchange protocol

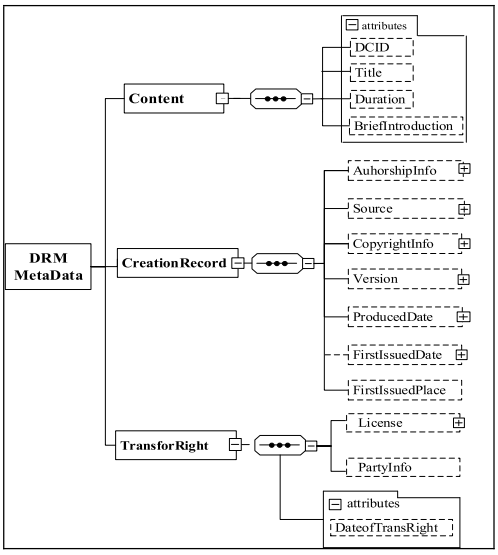


Fig. 4. The DRM metadata structure of DRMChain.

based on Ellipse Curve Cryptosystem (ECC), which wed define the protocol as SEKEC, in which during the client and server sides start up the interaction, we adopt a session management mechanism rather than the cookies mechanism in the negotiation procedure. In fact, the cookies can be forbidden by different users' client security policy, once the cookie is forbidden, the OAKLEY protocol does not work again. Comparing with the cookie approach, the session mechanism is built up on the server side and can provide a flexible and reliable management of interaction between client and server, which can manage and keep the state and session status for current and future interaction for convenient and efficient communication and interaction.

5.1.1. SEKEC key agreement security foundation

The SEKEC key exchange protocol depends on 4 components of the key determination protocol:

(a) Application-level user/client password authentication;

(b) Session mechanism based statue management between initiator and responder.

(c) Abel group based ECC half-key exchange with perfect forward security.

(d) Public key cryptosystem for identity hiding.

5.1.2. The SEKEC key agreement symbol definition

To comparing with the original OAKLEY protocol, we still inherit the symbol in OAKLEY protocol. The symbols in SEKEC key exchange protocol are listed in Table 2.

5.1.3. The SEKEC key agreement in DRMChain

Similar to the OAKLEY protocol, the SEKEC protocol will base on OAKLEY aggressive example with hidden identities. The following procedure indicates how SEKEC protocol two parties can complete a key exchange without using digital signatures. Public key cryptography hides the identities during authentication. The group exponentials are exchanged and authenticated, but the implied keying material (abG is not needed during the exchange).

In the DRMChain scheme, we uses the multiplicative group of integers modulo p, where p is prime, and g is a primitive root modulo p. These two values are chosen should ensure that the shared secret can take on any value from 1 to p-1. Alice and Bob agree on a finite cyclic group G of order n and a generating element g in G,

Table 2
The symbols in SEKEC key exchange protocol.

No.	Field	Expression
1	SESN-I	originator session.
2	SESN-R	Responder session.
3	MSGTYPE	For key exchange, will be ISA_KE&AUTH_REQ or ISA_KE&AUTH_REP; for new group definitions, will be ISA_NEW_GROUP_REQ or ISA_NEW_GROUP_REP
4	GRP	The name of the Diffie–Hellman group used for the exchange
5	aG, bG	G representing group generator in ECC cryptosystem
6	EHAO EHAS	Encryption, hash, authentication functions, offered and selected, respectively
7	IDP	An indicator as to whether or not encryption with abG follows (perfect forward secrecy for ID's)
8	ID(I)	The identity for the Initiator
9	ID(R)	The identity for the Responder
10	Ni	Nonce supplied by the Initiator
11	Nr	Nonce supplied by the Responder

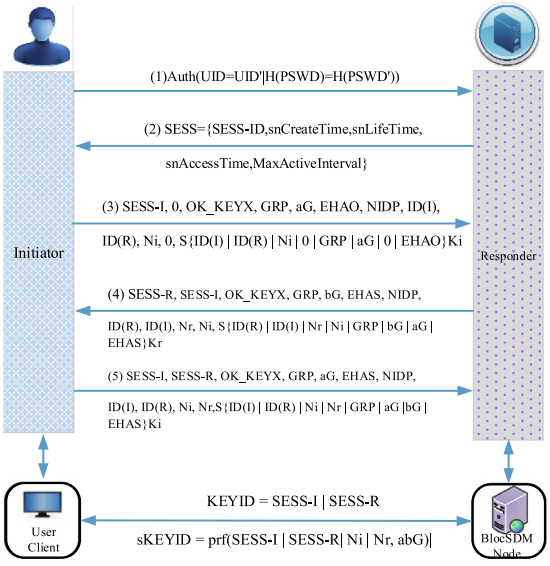


Fig. 5. The SEKEC protocol interaction procedure.

where the group G is written multiplicatively. The SEKEC protocol procedure is as Fig. 5.

Especially, for the symbol is described as in the OAKLEY protocol, to simplify the SEKEC in a clear mode, we omit some symbol such as MSGTYPE, GRP, EHAO, EHAS, NIDP, and focus on the core interaction steps and algorithms.

Step 1: the DRMChain client submits its username and password and tries to login the system whether it can match the user information and pass the verification or not:

$$bAuth = Auth(UID = UID' \parallel H(pswd) = H(pswd'))$$

If bAuth = true, it manifests the current user is valid.

Step 2: the DRMChain client user requests DRMChain Node to build up the key exchange in a session mode, The DRMChain Node then creates a session for the current client user in the server side (instead of OAKLEY protocol in the client side), and keeps the session in a whole transaction status, the session information is described as follows:

$$SESS = \left\{ \begin{matrix} SESS-I, SESS-R, CreateTime, LifeTime, \\ AccessTime, MaxActiveInterval \end{matrix} \right\} \quad (6)$$

Step 3: After creates the session, the client randomly selects a integer a(1< x< p−1), and then computes:

$$Q_A = aG = (x_{Q_A}, y_{Q_A}) \quad (7)$$

And together randomly creates a Nonce N_A, then organizes the core data as (omits MSGTYPE, GRP, EHAO, EHAS, NIDP data):

$$M_A = SID_A, UID_A, UID_B, N_A, Q_A \quad (8)$$

The use client signs the message M_A as follows:

Client user A signs the message as follows:

A randomly selects k_A computes:

$$k_AG = (x_A, y_A) \quad (9)$$
$$r_A = x_A \ mod \ n \quad (10)$$
$$e_A = h(M_A) \quad (11)$$

That is:

$$e_A = h(SID_A, UID_A, UID_B, N_A, Q_A) \quad (12)$$

Then user A computes:

$$s_A = r_Ak_A + e_Ad_A \ mod \ n \quad (13)$$
$$K = \mathrm{H}\left(SK|Ua|Ub|K_{grp}|SnID\right) \quad (14)$$

Then the client user A sends the message and its signature to server B:

$$A \rightarrow B : M_A = SID_A, UID_A, UID_B, N_A, Q_A, \ Sig_A \quad (15)$$

Step 4: Once the server B receives the message M_A from A, and then verifies the signature Sig_A as follows:

$$e_A = h(SID_A, UID_A, UID_B, N_A, (x_{Q_A}, y_{Q_A})) \quad (16)$$
$$u = r_A^{-1}s, \ \ v = r_A^{-1}e \quad (17)$$
$$X = uG - vQ = (x_1, y_1) \quad (18)$$
$$r_1 = x_1 \bmod n \quad (19)$$

If $r_1 = r_A$ is true, it manifests the signature of A is valid.

After verifies the message M_A, the server B randomly selects an integer b(1< b< p−1), and computes:

$$Q_B = bG = (x_{Q_B}, y_{Q_B}) \quad (20)$$

And together randomly creates a Nonce N_B, then organizes the core data as (as the above steps, here we also omit MSGTYPE, GRP, EHAO, EHAS, NIDP data):

$$M_B = SID_A, SID_B, UID_A, UID_B, N_A, N_B, Q_A, Q_B \quad (21)$$

The server B signs the message M_B as follows:
The server B randomly selects k_B computes:

$$k_BG = (x_B, y_B) \quad (22)$$
$$r_B = x_B \ mod \ n \quad (23)$$
$$e_B = h(M_B) \quad (24)$$

That is:

$$e_B = h(SID_A, SID_B, UID_A, UID_B, N_A, N_B, Q_A, Q_B) \quad (25)$$

Then the server B computes:

$$s_B = r_B k_B + e_B d_B \bmod n \quad (26)$$

$$Sig_B = (r_B, s_B) \quad (27)$$

Then the server B sends the message and its signature to client A:

$$B \rightarrow A: \quad M_B = SID_A, SID_B, UID_A, UID_B, N_A, N_B, Q_B, Sig_B \quad (28)$$

Step 5: When user client A receives the message from B, then organizes the core data as:

$$M_A = SID_A, SID_B, UID_A, UID_B, N_A, N_B, Q_A \quad (29)$$

And signs the message M_A using the private key used in step 3 as follows:

$$A \rightarrow B: \quad M_A = SID_A, SID_B, UID_A, UID_B, N_A, N_B, Q_A, Q_B, Sig_A \quad (30)$$

And client user A signs the message as follows:

$$k_A G = (x_A, y_A) \quad (31)$$

$$r_A = x_A \bmod n \quad (32)$$

$$e_A = h(SID_A, SID_B, UID_A, \mathrm{UID_B}, N_A, N_B, Q_A, Q_B) \quad (33)$$

$$s_A = r_A k_A + e_A d_A \bmod n \quad (34)$$

$$K = \mathrm{H}\left(SK|Ua|Ub|\mathrm{K_{grp}}|SnID\right) \quad (35)$$

When server B receives the message M_A, then verifies the validation, if the signature is true. Then the key negotiation procedure finished.

Step 6: then user A and server B key exchange is deduced as follows:

$$K_{AB} = bQ_A = baG = abG = aQ_B = K_{BA} \quad (36)$$

$$K_{AB} = (x_{K_{AB}}, y_{K_{AB}}) \quad (37)$$

We define the keyID as:

$$KEYID = SID_A|SID_B \quad (38)$$

And then the key negotiated as follows:

$$K = H(SID_A|SID_B|UID_A|UID_B|N_A|N_B|x_{K_{AB}}|y_{K_{AB}}) \quad (39)$$

According to the above negotiation the system then builds up the communication encryption key K.

5.2. DRMChain traceable identity management and privacy protection

5.2.1. DRMChain identity composition and management

In the DRMChain scheme, to identify the validation content provider, we proposed an enhanced and traceable authentication and privacy management approach, in which the DRMChain manager node can trace and confirm the real legal identity, the basic identity includes: unique userID (UUID), user identity, network identity (IP), device identity (MAC), location information, social network system (SNS) account (WeChat ID, Facebook ID, et al.) other identity include mobile phone number or email account. The DRMChain identity and privacy protection model is described in Fig. 6.

In the DRMChain scheme, to protect the user's identity information from being misused or arbitrarily spreading, the user's identity is strictly limited for usage without authorization. In the DRMChain scheme, we proposed encryption approach of identity information to prevent user's privacy from being leakage including from internal management member, such as internal administrators or auditors.

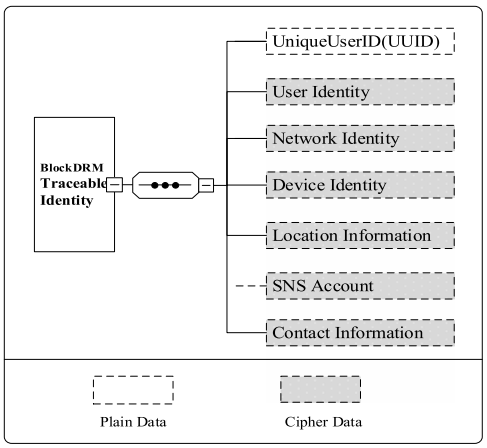


Fig. 6. The DRMChain identity and privacy protection model.

To securely manage the user's sensitive privacy data, and together for the later possible auditing behavior in case of user's data was deliberately modified or misused, the system can independently recover and open the sensitive data for auditing and tracing.

5.2.2. DRMChain multipart determined identity privacy protection

In the DRMChain scheme, we proposed a multiple-part determined privacy protection approach, in which the privacy data was encrypted by 3-parts-controlled symmetric key. In the DRMChain scheme, the 3 parts are:

(A) blockchain manage node.
(B) blockchain audit node.
(C) the blockchain client user.

The DRMChain multipart determined identity protection procedure is described as follows:

Step 1: The DRMChain platform manager selects a special symbol as its blockchain identity, and defines another use case identity CaseID (such as different application platform), and computed the Hash as its unique bcID, the bcID is defined as:

$$bcID = Hash\,(BlockChainID, CaseID) \quad (40)$$

Step 2: The manage node control the administration control word bcAdminCW, where the administration key K_A is determined as follows:

$$K_A = Hash(bcID \oplus bcAd \min CW) \quad (41)$$

Step 3: The DRMChain audit node keeps the auditing control word bcAuditCW, where the auditing key K_D is determined as follows:

$$K_D = Hash(bcID \oplus bcAuditCW) \quad (42)$$

Step 4: The DRMChain client user controls his/her UID as his/her control word (here because the UID is determined and as constant which will not allow change, whereas its password is often changed, thus here we adopt UID as the user control word).

$$K_U = Hash(bcID \oplus UID) \quad (43)$$

Step 5: The DRMChain super-administrator then creates the privacy data encryption key K_{AU} as follows:

$$K_{AU} = Hash\,(K_A \oplus K_U) \quad (44)$$

Table 3
Development environment parameters.

OS	Ubuntu16.0 4.2 X64 Server	Hardware	8 GB RAM 500 GB disk
RAM	16 GB	CPU	Intel i7-8550U
Blockchain	Ethereum	Develop Tool	solc node-v6.9.4 React, Java truffle testrpc
External DB	js-ipfs 0.27.0	Nodes amount	12
Digital signature	ECDSA	AES	128-CTR
Key agreement	SEKEC	Privacy protection	Multi-signature based on ECDSA
DRM protection	HTTP living stream,	DRM server	Nginx
DRM tool	FFmpeg	File extension	M3U8, ts

Step 6: the DRMChain super-administrator encrypts the key by his/her public key $K_{\text{pub:}}$

$$C_{K_D} = E_{K_{pub}}(K_D) \tag{45}$$

And the super-administrator signs the C_{kD}. The signature procedure is described as follows:

The DRMChain administrator randomly selects k computes:

$$kG = (x, y) \tag{46}$$
$$r = x \bmod n \tag{47}$$
$$e = h(K_{AU}) \tag{48}$$

and computes:

$$s = rk + ed \bmod n \tag{49}$$
$$sig = (r, s) \tag{50}$$

Then (r, s) is the signature of K_{D}.

Step 7: the DRMChain super-administrator stores the K_{D} information and its signature as following:

$$KDB_{K_D} = Store(C_{K_D}, sig(C_{K_D})) \tag{51}$$

Step 8: the user's identity information then is encrypted as follows:

$$C'_{Identity} = E_{K_{AU}}(NetID, PhyID, LocID, SocialID, CommID) \tag{52}$$

Step 9: the DRMChain audit node then computes:

$$C_{Identity} = E_{K_D}(C'_{Identity}) \tag{53}$$

Step 10: the DRMChain securely stores the user's identity information in a secure mode:

$$IdentityDB = Store\left(UUID, C_{Identity}\right) \tag{54}$$

5.3. DRMChain DRM processing for consumption

5.3.1. Content encryption for usage control

When the Data is stored in the DRMChain system, then according to the DRM policy, the DRM engine starts processing the content into a content-protected mode, such as content encryption, or content watermarking. In the proposed DRMChain scheme, we adopt encryption approach for content protection from being illegal used. In the DRMChain, as for the content encryption efficiency, we adopt AES algorithm to encrypt the content itself, thus unauthorized user cannot access the protected content.

Step 1: The DRMChain CryptoEngine in the blockchain side then encrypts the content payload data as follows:

$$C = E_{K_{AU}}(M_1|M_2|\ldots M_n) \tag{55}$$

When creates the secure communication channel, the system can work in a dynamic security mode, the sensitive data can be transfer in the secure channel.

Step 2: the DRMChain encryption engine then packets the License L and its signature into the cipher content:

$$DRMData = E_{K_{AU}}(C|L|Sig(L)) \tag{56}$$

Step 3: when received the encryption data from the client user, the DRMChain node decrypts the cipher data as follows:

$$b\,\text{Result} = Veri(Sig(L)) \tag{57}$$

Step 4: if bResult = True, then the DRMChain CryptoEngine decrypts the cipher payload for memory content play:

$$C = E_{K_{ADU}}(M_1|M_2|\ldots M_n) \tag{58}$$
$$M = D_{K_{ADU}}(E_{K_{ADU}}(M_1|M_2|\ldots M_n)) \tag{59}$$

In the above section, the M_i $(1 < i \leq n)$ is the payload of content, which may be different data unit defined according to its data structure(such as MPEG-2, WMV, FLV, H.265, JPG et al.) network protocol (such as RTP, RTSP, MMS, HTTP living streaming et al.), the encryption and decryption focus on the effective payload.

5.3.2. Watermark embedding algorithm

We select the M × N binary image as watermark W. $W = \{W(i,j)|0 \leq i < M, 0 \leq j < N\}$, and $W(i,j) \in \{0,1\}$. For security, we scramble the watermark and then scan the image into one dimensional signal, namely $W = \{W\}_i$, $i = 1, 2, \ldots, C$; $C = M \times N$, $w_i = 0$ *or* 1.

And we also select the $P \times Q$ image as host image.

The whole image is divided into 8×8 blocks, named $X_i, x_i(m, n)$

Is the pixel value of (m, n) of X_i. After zig-zag the DCT coefficients, the whole sequence is recorded as $C_i(j), (j = 0, 1, 2, \ldots, 63)$. i is the sequence number corresponding to the ith block of the image. We select a continuum of values to embed watermark, as

$C_i(k-2), C_i(k-1), C_i(k), C_i(k+1), C_i(k+2), k = 2, 3, \ldots, 61$

The specific methods are as follows:

if $w_i = 0$, then

we can get

$$C_i(k)' = \frac{1}{5}\sum_{l=k-2}^{k+2} C_i(l) - Q_i \tag{60}$$

else if $w_i = 1$, then

we can get

$$C_i(k)' = \frac{1}{5}\sum_{l=k-2}^{k+2} C_i(l) + Q_i \tag{61}$$

$D_i(k)'$ is the modified coefficient, Q_i is the factor for controlling the watermark strength. Q_i is defined as follows:

$$Q_i = a\delta_i \tag{62}$$

where δ_i is the average energy value of block image i, a is constant.

For decreasing the quantization influence, the analysis of quantitative condition process is created to control the strength of watermark.

Analysis of the quantitative condition is as follow:

$$\begin{aligned}
&when\ w_i = 0,\\
&while\ R\left(\frac{C_i(k)'}{Q(k)}\right)\cdot Q(k) \geq \frac{1}{5}\sum_{l=k-2}^{k+2} R\left(\frac{C_i(l)}{Q(l)}\right)\cdot Q(l) - q,\\
&do\ Q_i = Q_i + 1, C_i(k)' = \frac{1}{5}\sum_{l=k-2}^{k+2} C_i(l) - Q_i,\\
&when\ w_i = 1,\\
&while\ R\left(\frac{C_i(k)'}{Q(k)}\right)\cdot Q(k) \leq \frac{1}{5}\sum_{l=k-2}^{k+2} R\left(\frac{C_i(l)}{Q(l)}\right)\cdot Q(l) + q,\\
&do\ Q_i = Q_i + 1, C_i(k)' = \frac{1}{5}\sum_{l=k-2}^{k+2} C_i(l) + Q_i,
\end{aligned} \tag{63}$$

$Q(k)$ means the QP from the quantization table corresponds to $C(k)$. q is the controlling factor of analysis quantitative condition. Q_i increases with the increasing of q. $R(\cdot)$ indicates rounding down.

As JPEG compression may affect the tamper detecting watermark, the semi-fragile watermark should tolerate some common image processing operations, such as JPEG compression. In order to avoid affecting the robustness of copyright identification watermark, the watermarks for tamper detecting are embedded in DC coefficients using quantitative method. The Specific methods are as follows:

$$C_j(0)' = \mathrm{R}(\frac{C_j(0) + 0.5step}{2step}) \times 2step + step/2 \tag{64}$$

$C_j(0)$ is the DC coefficient in 8 × 8 block image, $C_j(0)'$ is the modified coefficient, $R(\cdot)$ indicates rounding down. *step* describes the quantization steps, $j = 1, 2, \ldots, \mathrm{P} \times \mathrm{Q}$.

5.3.3. Watermark extracting algorithm

(1) Responsibility watermark extracting algorithm

During the embedding process, we obtain the same continuum of values for watermark extraction:

$$\begin{aligned}
&C_i(k-2), C_i(k-1), C_i(k), C_i(k+1), C_i(k+2),\\
&\quad k = 2, 3, \ldots, 61
\end{aligned} \tag{65}$$

Extracting method is as follow:

$$\begin{aligned}
&if \quad C_i(k) > \frac{1}{5}\sum_{l=k-2}^{k+2} C_i(l)\\
&\quad\ w_i = 1,\\
&else \quad w_i = 0,
\end{aligned} \tag{66}$$

where w_i is the ith watermark bit. At last, anti-scramble the information to get the watermark extracted.

(2) Tamper detecting algorithm

After dividing the watermarked JPEG image into 8 × 8 blocks, DCT each block. The specific tamper detect methods are as follows:

$$\begin{aligned}
&if \quad \mathrm{mod}(R(\frac{C_i(0)}{step}), 2) == 0 \quad \text{no tamper}\\
&\text{else} \qquad\qquad\qquad\qquad\ \text{tamper}
\end{aligned} \tag{67}$$

$C_i(0)$ is the DC coefficient in 8 × 8 block image, $R(\cdot)$ indicates rounding down, *step* describes the quantization steps, $i = 1, 2, \ldots, \mathrm{P} \times \mathrm{Q}$.

If the block is tampered, location it with marks, such as black block image.

5.4. DRMChain multi-signature-based violation tracing

Once some content was considered as violation, then the DRMChain platform manager startups the investigation who should be responsible for the content. To avoid arbitrary decision or judge, based on the study and research work [33–38], in the DRMChain scheme we proposed a multi-signature-based evaluation decision (MSED) mechanism for multi-parts evaluation rather than one unique judge.

(a) DRMChain Violation Evaluation Task Release: DRMChain Manager Node responsible for the content violation task releasing, initializes, collects and verifies the multi-signature.

(b) DRMChain Peer Evaluation: respectively signs the blank evaluation decision table (BEDT) if and only if t-out-of –n parts sign the BEDT as definite decision results, the DRMChain then accepts the decision results as final evaluation result.

(c) DRMChain Conditional Violation Tracing: once t-out-of-n decision results give the definite violation evaluation result, then the DRMChain manager node starts up the tracing procedure for user's identity who to responsible the violation.

5.4.1. DRMChain violation evaluation task release

Once the content is considered as violation, the DRMChain manager node sends each peer node p_i the abstract of violation description (AVD), and a blank evaluation decision (BED) to be signed for the evaluation. The DRMChain manager node public the common parameter p, g Z_P, and H(.), and sends each node p_i signature timestamp T and require each node p_i signs the message in the specified time T_0, when p_i receives the message, then deals with the signature.

5.4.2. DRMChain broadcast multi-signature

Let m be the blank evaluation decision table (BEDT) as message to be signed, here we suppose there are n members U_i $(1 < i < n)$ which can sign the message, to finish the blind multi-signature, for each signature member Ui, he randomly selects a secret number di(di ∈ Zn) as his private key, and computes Qi = diG as his public key. X(.) means the function that gets the X coordinate.

(1) DRMChain broadcast multi-signature

Step 1: each signature member U_i $(1 < i < n)$ selects an integer ki, $1 \leq ki \leq n - 1$, and computes:

$$R_i = k_i G \tag{68}$$

and sends the result Ri to signature collector.

Step 2: the signature collector computes:

$$R = \sum_{i=1}^{n} R_i \tag{69}$$

$$r = R_x \bmod n \tag{70}$$

If (r, n) = 1, then send the result r to each signature member Ui(i = 1,2,…,n) and the message holder, otherwise, go to step1 to reconstruct the signature.

Step 3: the message holder U randomly select an integer$\alpha \in Z_q^*$, computes:

$$Q = \sum_{i=1}^{n} Q_i \tag{71}$$

The message computes $\beta = Q_x$

Step 4: the message holder computes:

$$e = H(\alpha \cdot m + \beta \cdot (H(m, T)) \bmod n) \tag{72}$$

and sends e to each signature $U_i(i = 1, 2, \ldots, n)$.

Step 5: each signature member $U_i(i = 1, 2, \ldots, n)$ computes:

$$s_i = k_i e + r\mathrm{d}_i \bmod n \tag{73}$$

$$G_i = s_i G \tag{74}$$

$$S = \sum_{i=1}^{n} s_i G \tag{75}$$

$$s = S_x \tag{76}$$

and sends the si to the message holder. Then (m, (r, s)) is the multi-signature of the message m.

(2) DRMChain multi-signature verification

The signature collector can verify the signature by verifying the equation as follows:

$$rQ = sG - eR \tag{77}$$

If the above result is true, then the multi-signature is valid, otherwise the signature is false.

$$\begin{aligned} sG - eR &= \sum_{i=1}^{n}(k_i e + rd_i)G \bmod n - eR \\ &= \sum_{i=1}^{n} ek_i G + r\sum_{i=1}^{n} d_i G - eR \\ &= e\sum_{i=1}^{n} k_i G + r\sum_{i=1}^{n} d_i G - eR \\ &= e\sum_{i=1}^{n} R_i + r\sum_{i=1}^{n} Q_i - eR \\ &= r\sum_{i=1}^{n} Q_i \\ &= rQ \end{aligned} \tag{78}$$

5.4.3. DRMChain multipart-determined identity tracing

Once the content provider is found t content violation, then the arbitrator dynamically computes the privacy key and then decrypts and recovery the identity information and trace the content provider accurately identity to deal with the content, which may give punishment decision. The DRMChain identity tracing procedure is described as follows:

Step 1: the DRMChain manage node computes the amount of evaluation results from auditing nodes, if and only if more than t auditing nodes give the agreement decision as violation result, the computing procedure is described a follows:

$$C = \sum_{i=1}^{n} c_i(r_i) > t_0 \tag{79}$$

$$\text{Where } c_i(r_i) = \begin{cases} 1, r_i = agreement \\ 0, r_i = disagreement \end{cases} \tag{80}$$

Step 2: if the $C > t_0$ then the DRMChain manage node finds the relationship R between the DCID and UID, and queries by the UID and return the identity cipher as follows:

$$R = \left\{DCID, UID, C_{identity}\right\} \tag{81}$$

Step 3: the DRMChain node then decrypts the cipher as follows:

Step 4: the DRMChain audit node then computes:

$$C'_{Identity} = D_{K_D}(C_{Identity}) \tag{82}$$

Step 5: the DRMChain manage node decrypts the identity of UID as follows:

$$\begin{aligned} I_{Identity} &= D_{K_{AU}}(E_{K_{AU}}(NetID, PhyID, LocID, SocialID, CommID)) \\ &= NetID, PhyID, LocID, SocialID, CommID \end{aligned} \tag{83}$$

Step 6: and then constructs and recover the violation information as follows:

$$V_{info} = \{DCID, UID, NetID, PhyID, LocID, SocialID, CommID\} \tag{84}$$

6. Security analysis of DRMChain scheme

6.1. Security analysis of SEKEC protocol

6.1.1. Message integrity of the 3 core procedure

In fact, we during the 3 turns in the SEKEC protocol, we use ECDSA as the signature algorithm for message signature. And the verification can bed as follows (i = A, B):

$$\begin{aligned} X &= u_i G - v_i Q \\ &= r_i^{-1} s_i G - r_i^{-1} e_i d_i G \\ &= r_i^{-1}(r_i k_i + e_i d_i)G - r_i^{-1} e_i d_A G \\ &= k_i G + r_i^{-1} e_i d_i G - r_i^{-1} e_i d_i G \\ &= k_i G \end{aligned} \tag{85}$$

Then $X_i' = (x_i', y_i')\mathrm{mod}nr_i' = x_i'\mathrm{mod}n$ there must exist $r_i' = r_i$.

6.1.2. Replay attack analysis

In the SEKEC protocol, in each step, we use Nonce number as the fresh timestamp each step

$$A \rightarrow B: \quad M_A = SID_A, UID_A, UID_B, N_A, Sig_A \tag{86}$$

$$B \rightarrow A: \quad M_B = SID_A, SID_B, UID_A, UID_B, N_A, N_B, x_{Q_B}, y_{Q_B}, Sig_B \tag{87}$$

If the attacker can forge a Nonce, and send the message and Nonce to the receiver, however he can NOT pass the signature verification, then the message's freshness is ensured, thus the SEKEC protocol is replay attack resistant.

6.1.3. Middle-man attack analysis

Similar to replay attack analysis, although the message is not encrypted in all the communication procedure, however because the final message send to the receiver is signed, once the message is replaced, it will NOT pass the validation in the signature validation stage.

$$A \rightarrow B: M_A = SID_A, UID_A, UID_B, N_A, Sig_A \tag{88}$$

Upon the message M_A, if the middle-man tries to substitute the message M_A:

$$M_A = SID_A, UID_A, UID_B, N_A, X_A \tag{89}$$

However, the session mechanism assures only valid session user can access the conversation which creates by the server side and keep the conversation in a reasonable time interval that defined by MaxActiveInterval, which satisfies the following condition:

$$CurrentTime - CreateTime < MaxActiveTime \tag{90}$$

Then the attacker cannot tamper the SID_A, or SID_B. The attacker can only attack and substitute UID_A, UID_B, or N_A.

6.1.4. The session security

In the proposed SEKEC protocol, when the client user communicates with the server, the server creates session for the client, and save the session for client access, and check the validation and according to the following

$$CurrentTime - CreateTime < MaxActiveTime \tag{91}$$

Table 4
The genesis block configuration parameters.

```
{
    "config": {
        "chainId": 15,
        "homesteadBlock": 0,
        "eip155Block": 0,
        "eip158Block": 0
    },
    "difficulty": "4",
    "gasLimit": "21000000",
    "alloc": {
    "0xef17fb45b8433a249a073e118581638cc8575d27":
    { "balance": "500000000000000000000000" },
    "0x18c7d7af78815064cd9c3e06f283308321f0a12f":
    { "balance": "500000000000000000000000" },
    "0xdec1a00564bb1de1941a48bb3112dcbe7b349e3d":
    { "balance": "500000000000000000000000" },
    "0x5cfc12fa1adbfb0321555b8895b7fe8af7034376":
    { "balance": "500000000000000000000000" },
    "0x1fcb569a200cb7b3510cc7274c15662bb813f8ee":
    { "balance": "500000000000000000000000" }
  }
}
```

Table 5
The blockchain information created from genesis block.

```
{
  difficulty: 4,
  extraData: "0x",
  gasLimit: 21000000,
  gasUsed: 0,
  hash:
"0x6cb90bf905157433c9e8d6202453c380582a44499331f61ef6bdf75
aee0e56b4",
  logsBloom: "0x00",
  miner: "0x0000000000000000000000000000000000000000",
  mixHash:
"0x0000000000000000000000000000000000000000000000000000
000000000000",
  nonce: "0x0000000000000000",
  number: 0,
  parentHash:
"0x0000000000000000000000000000000000000000000000000000
000000000000",
  receiptsRoot:
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e
363b421",
  sha3Uncles:
"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142f
d40d49347",
  size: 505,
  stateRoot:
"0x31352e470c0643e32696f578a8382d2275f52378490d325b372112
c0e66b4ca8",
  timestamp: 0,
  totalDifficulty: 4,
  transactions: [],
  transactionsRoot:
"0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e
363b421",
  uncles: []
}
```

The session can save time and provide a controllable mode without possible tamper by the client user, and can provide a secure and efficient conversation between individual users from client and server side; especially it provides a memory mode for historical access records for existing and historical access user.

6.2. Privacy security and efficiency analysis

6.2.1. Multi-parts determined security analysis

Once the DRMChain content is doubt to be violation, whether it is truly violated or not, the DRMChain provides multi-signature decision mechanism [51–56]. The security and efficiency is based on the following:

When the content is doubt violation, the DRMChain auditing node will evaluates the content itself by checking the original content, if and only if more than t auditing nodes give the agreement decision as violation result, and the evaluation results will be signed by more than t auditing nodes, that is:

$$C = \sum_{i=1}^{n} c_i(r_i) > t_0 \tag{92}$$

In fact, the signature cannot cheat the manager node if one auditing node wants to cheat the manage node and provide the false signature then the above equation will not pass the verification.

6.2.2. Privacy protection and recovery

In the DRMChain scheme the identity privacy information of content provider is encrypted, once he/she is doubt as violation, the identity information is decrypted and recovered by DCID.

$$R := \left\{DCID, UID, C_{identity}\right\} \tag{93}$$

and the identity information is related by DCID with UID as follows:

$$V_{info} = \{DCID, UID, NetID, PhyID, LocID, SocialID, CommID\} \tag{94}$$

Thus In the proposed DRMChain the privacy security and efficiency is ensured, because the content provider's identity is encrypted and stored in cipher mode, without authorization, even the sole internal manager cannot recover the content provider's identity information for the DRMChain multiparts determined multi-signature mechanism can prevent misuse or unauthorized operation of sensitive data.

7. Implementation and evaluation of DRMChain scheme

7.1. Implementation & evaluation of DRMChain

Based on the DRMChain architecture, we developed and implemented the DRMChain system as a blockchain application platform for digital rights management. In the DRMChain system, we build up the DRMChain platform based on Ethereum, and the js-ipfs 0.27.0 P2P network for external data storage, and the development tools Solc, Node-v6.9.4, React, Java, truffle and Testrpc are used for application development, the watermark-based and encryption-based DRMChain was completely implemented, and the development environment is listed in Table 3, and the main GUI of DRMChain is described as in Figs. 7 and 8.

We deployed 3 private Ethereum Nodes in Aliyun could platform that support for DRMChain management, which include nodes for BAIP for summary metadata storage of original content, and the BAIC for DRM-protected content service, such as content watermark, encryption, license, violation tracing. The DRMChain runtime environment is described as Table 3.

Table 6
Instance of IPFS information in DRMChain.

Key	Value
Ethereum contract address	0xb752ffa78d7634c0901df669d3f1fabab5057a76
ImageHash in IPFS	QmYANV86z9hKRkb5GJcCG9X5tnE3kVWqw8hLnmNVkjJa1K
ImageHash in DRMChain	QmYANV86z9hKRkb5GJcCG9X5tnE3kVWqw8hLnmNVkjJa1K
BlockHash in DRMChain	0xe0010353e960e50dcad4d1ca5f30b56fdf749212157402c722e25b5385c1ab96

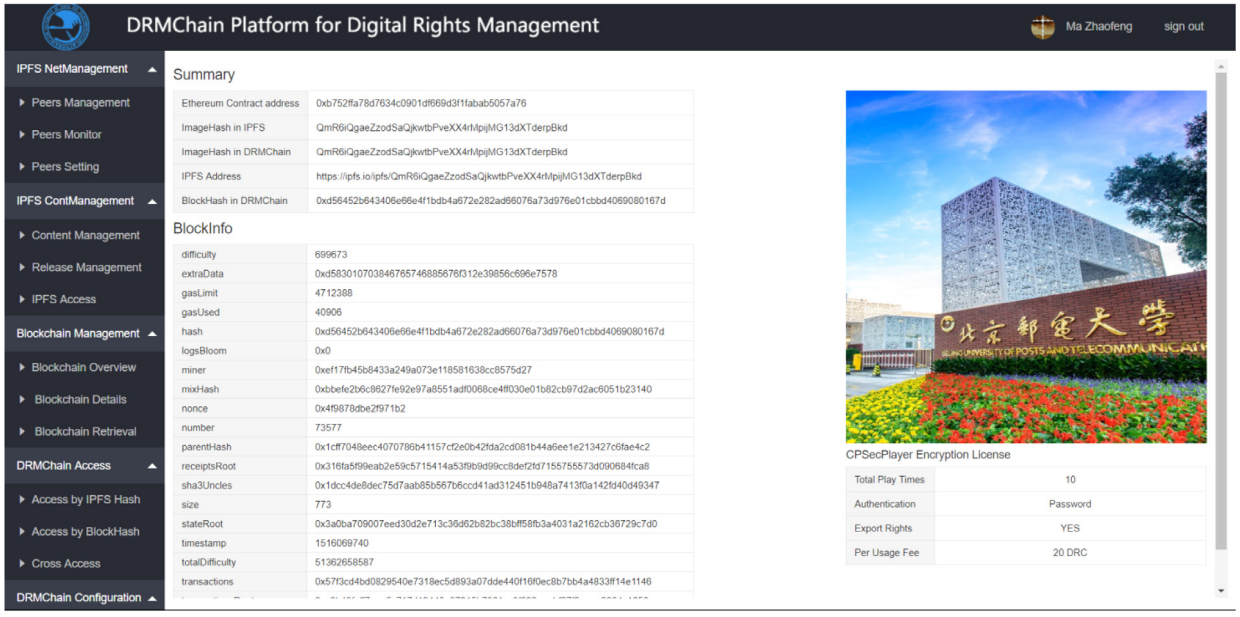


Fig. 7. DRMChain encryption-based license management in blockchain and IPFS information.

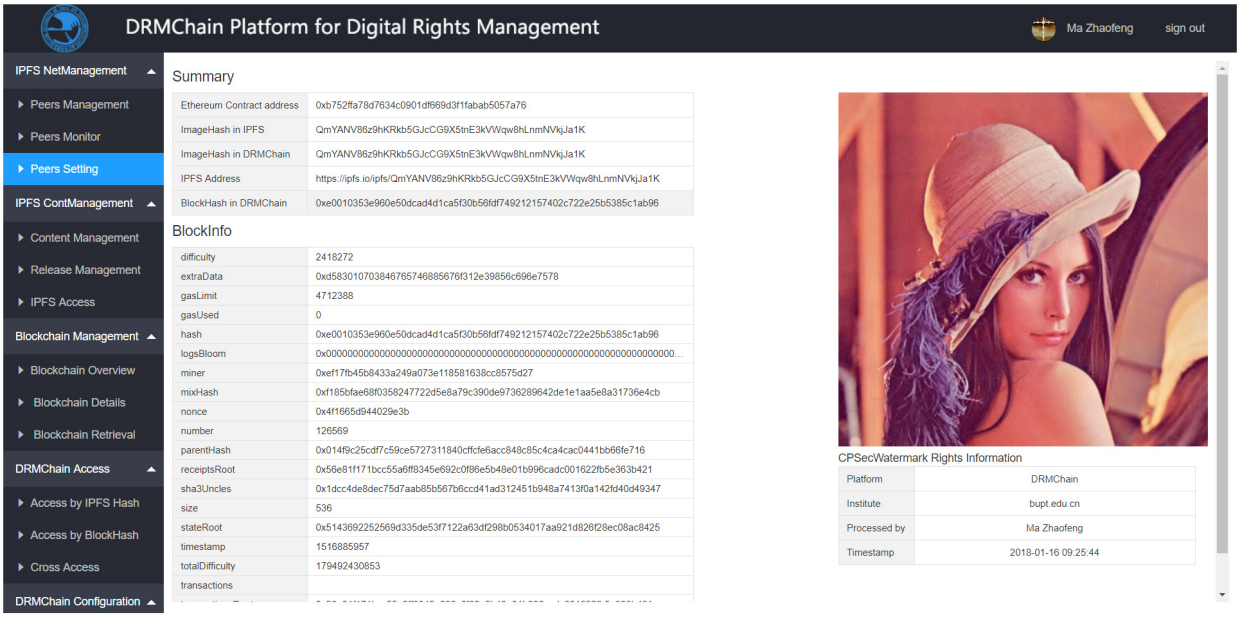


Fig. 8. DRMChain watermark-based rights information in blockchain and IPFS.

7.2. Experiments of DRMChain scheme

An instance of image type content in DRMChain includes 3 parts information: IPFS network, Blockchain platform, and digital rights management platform. The genesis block configuration and its blockchain information are listed in Tables 4 and 5. And the instance information Of IPFS and DRMChain Instance information of the DRMChain platform we have implemented for digital rights management are listed as Tables 6 and 7, and 8 listed the watermark rights information extracted from Lenna. The DRMChain is suitable for "building up blocks in internal blockchain platform, but

Table 7
Instance of block information in DRMChain.

BlockInfo
{
difficulty: 698316,
extraData:
"0xd583010703846765746885676f312e39856c696e7578",
gasLimit: 4712388,
gasUsed: 40906,
hash:
"0xe0010353e960e50dcad4d1ca5f30b56fdf749212157402c722e25b5385c1ab96",
logsBloom: "0X0",
miner: "0xef17fb45b8433a249a073e118581638cc8575d27",
mixHash:
"0x081c2f6c16967823ae70f68f6a3a50e1c12089dd332487e85512b44eb9eec8dc",
nonce: "0x35b6924807e97653",
number: 73557,
parentHash:
"0x11ac07300251305c712c42256bf55954ceb02c52f29dafe6c9c088d63e1e5308",
receiptsRoot:
"0x3965430c0ae4d90f72afe5cc73d6ef29630190687b76dedbc215ed1e2b0c5317",
sha3Uncles:
"0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347",
size: 773,
stateRoot:
"0x540db56290eb1bfc32c5ddd66d14d3c06edd18c30df79ec5f6140a42284e981a",
timestamp: 1516069500,
totalDifficulty: 51348662986,
transactions:
["0x9bf72c57ef5ca624f9102bd7c8dfe34dc31a4816500c537cc806dc016410
117f"],
transactionsRoot:
"0x7348371bd46f911f7be48bac264baf713915946a50a088522a046abc3a645ca2",
uncles: []
}

Table 8
Instance of rights information in DRMchain.

Key	Value
Platform	DRMChain
Institute	bupt.edu.cn
Author	Ma Zhaofeng
Timestamp	2018-01-16 09:25:44

storing the content itself in external database'' as described in our proposed trusted model.

We evaluated the DRMChain platform for content storage digital rights management of video content protection based on HTTP living stream which can support iOS, Android, Windows application with a wide range of user adaption, and the DRM server we used Nginx and the digital video and audio encode/decode tools we used FFmpeg 3.3 for the content processing. In which the media protection is based on Media Stream Segmenter, Media File Segmenter, Media Stream Validator, Variant Playlist Creator, and Metadata Tag Generator. And the content is encrypted by AES-128, and the crypto-middleware we used OpenSSL [45]. Evaluations of the DRMChain are described as follows.

Fig. 9. Watermark embedding and extracting in DRMChain.

7.3. Experiments and analysis of watermark

7.3.1. Experiments of watermark algorithm

Various experiments are carried out to assess the performance of the proposed scheme. PSNR (peak signal-to-noise ratio), is used in this paper to analyze the visual quality of the watermarked image $\hat{W}$ in comparison with the original image W. PSNR is defined as:

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})dB \tag{95}$$

where MSE is the mean squared error between the original image W and the watermarked image $\hat{W}$, given by

$$MSE = \frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[W(i,j) - \hat{W}(i,j)]^2 \tag{96}$$

In this experiment, 'Lenna' image of size 512 × 512 is used. A binary logo image of size 64 × 64 is used as watermark. Fig. 9 shows the host image, binary watermark and the corresponding watermarked image. The PSNR value of watermarked image is 38.96 dB.

7.3.2. Attacks and evaluation of DRMchain watermark

To evaluate the robustness and security of the scheme, we considered 4 kinds of attacks including: (1) copy and paste attack; (2) insert a circle attack; (3) insert a picture; (4) color inverse attack.

In the experiments (see Fig. 10) we firstly attack the exported image that had been embedded watermark, and then test whether we can extract the watermark, and can trace where the attack occurred. In the experiment, we gave the detailed experiments according to the 4 attacks: (1) copy and paste attack; (2) insert a circle attack; (3) insert a picture; (4) color inverse attack, where NC = 0.9884, 0.9939, 0.9934, 0.9394.

Fig. 10. Copy and paste attack, NC = 0.9666.

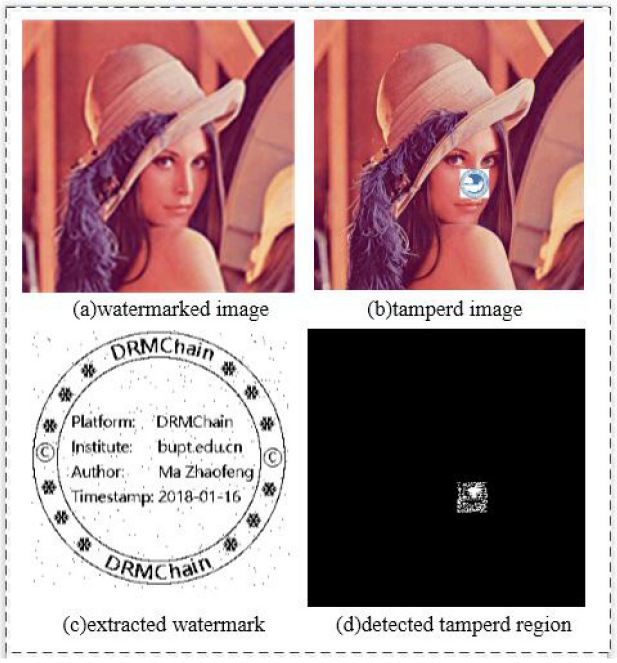


Fig. 11. Insert picture attack, NC = 0.9942.

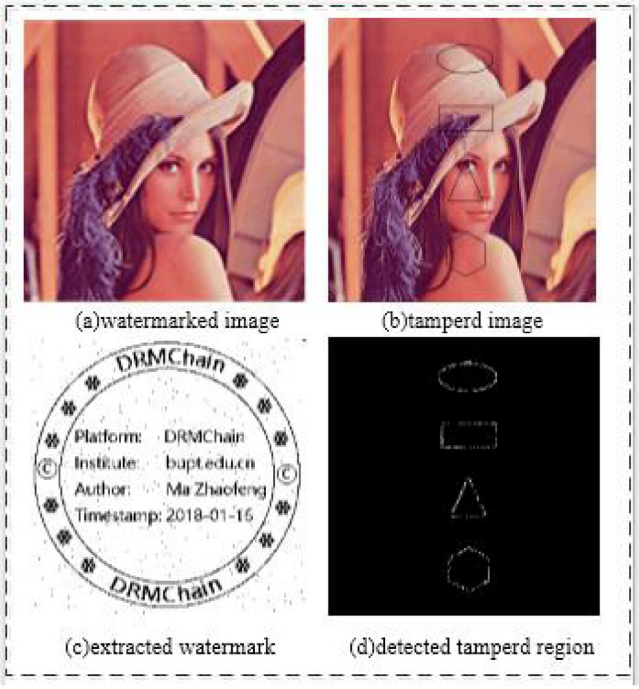


Fig. 12. Insert a circle attack, NC = 0.9951.

Upon the implemented DRMChain platform we evaluated the efficiency comparison of SEKEC and OAKLEY, efficiency comparison of plain and cipher video playing, and efficiency of multi-signature and its average signature, in which the video codec we used H.264 video with profile Baseline and level: 2.1, and the codec is avc1. The performance simulations and evaluations results of DRMChain are described from Figs. 9–13. Fig. 9 described watermark embedding and extracting in DRMChain without attacks, and Figs. 10–13 is the serial watermark attacks experiments of DRMChain. While Fig. 14 is the efficiency comparison of SEKEC and OAKLEY protocols, Fig. 15 is the efficiency comparison of plain and cipher video playing, and Fig. 16 is the efficiency of multi-signature and Its average signature, upon which the detailed evaluation and analysis of the DRMchain scheme is described in Section 7.4.

7.4. Evaluation of the DRMchain scheme

7.4.1. Availability of DRMchain scheme

The DRMChain Scheme provides a flexible DRM approach that enables user-controlled encryption but administrator and auditor can decrypt and audit the content once the released content is suspected violation or illegal usage, in the scheme, we proposed 3 parts control model trusted and creditable content encryption, secure key management, multi-signature for violation appraisal.

7.4.2. Extendibility of DRMchain scheme

We implemented the DRMChain platform for digital right management based on Ethereum blockchain platform. Large amounts of experiments manifest the DRMChain is reliable, secure, efficient and tamper-resistance with high-level credibility, in which the authorization users can upload their right-reserved digital content, but once the content is suspected illegal or rights infringement, the DRMChain can trace and checkout the violation content and provider user, the DRMChain provides an extendible, reliable and tamper-resistant DRM practice and can apply in real scene for content protection.

7.4.3. Security of DRMChain scheme

In DRMChain we used DCT-based watermark algorithm for image content rights protection and ownership confirmation. The algorithm is proposed and evaluated as efficient and secure for variant attacks in our another paper. And as for the video DRM protection, we used Http Living Streaming DRM for video content encryption and licensing service for playing times, exporting control, which is efficient and secure without memory cache and leakage in real application. In the proposed DRMChain the user's identity management and privacy protection, multi-signature-based

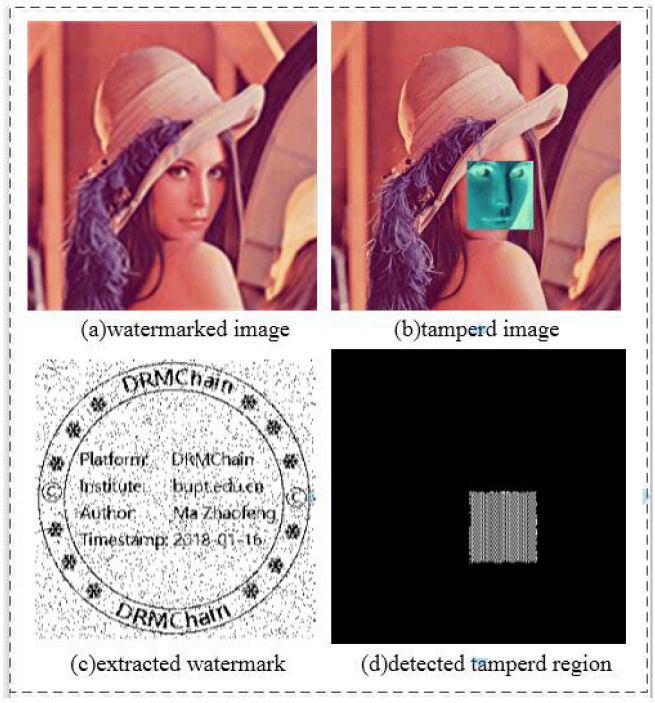


Fig. 13. Color reverse attack, NC = 0.9433.

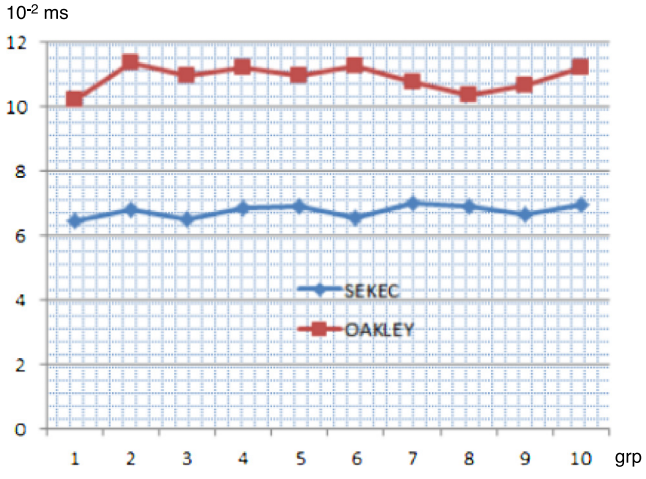


Fig. 14. The efficiency comparison of SEKEC and OAKLEY.

conditional traceability approaches, and thus the DRM license, usage control and constraint are ensured, which provided a new paradigm for the digital rights management for content consumption and protection.

7.4.4. Efficiency of DRMChain scheme

Upon the proposed DRMChain, external data storage in IPFS and blockchain internal block creation are evaluated in which we can find the common string, image, audio and video are easily and efficiently stored in IPFS network and blockchain nodes, the time consumed from string and video are all in acceptable range in DRMChain (in which the average time consumed in Ethereum is nearly about 12 s, and the other time consumed is mainly network traffic). And as for key agreement efficiency, plain/cipher video bitrate, and multi-signature efficiency are evaluated by large amounts experiments for performance evaluation, and we can see

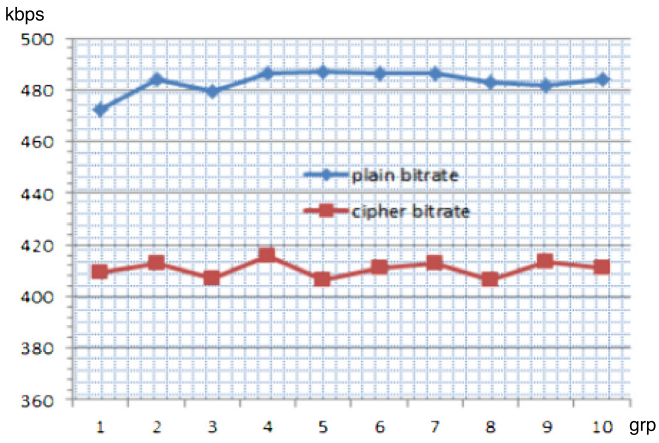


Fig. 15. Efficiency comparison of plain and cipher video playing.

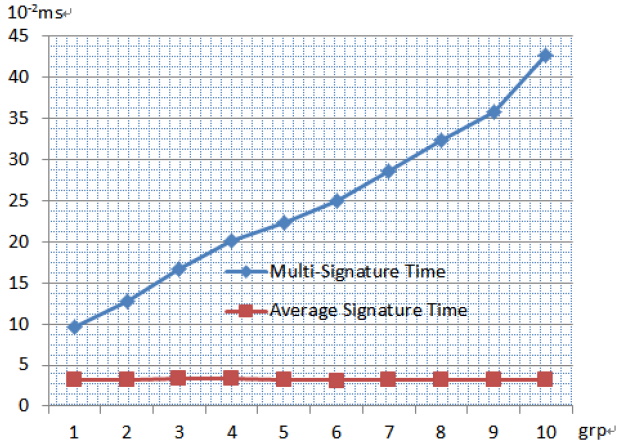


Fig. 16. Efficiency of multi-signature and its average signature.

the proposed SEKEC key agreement is efficient than the OAKLEY protocol, and the average time consumed in multi-signature is in the range of $3.21–3.35 * 10^{-2}$ ms, and average delay of ciphered video content is 17.42%, and the efficiency is high and can satisfied the speed of real-time application in blockchain platform.

7.5. Performance comparison with related work

Among the DRM research, the typical DRM scheme include [6–9], which proposed detailed and practical DRM solutions, although the work was based on mobile environments, the whole procedures and communications were still suitable for pervasive DRM protection. Thus we can compare the schemes with our proposed DRMChain scheme. in 2008, Chen [17] proposed a secure and traceable E-DRM system based on mobile device, in which the scheme applied symmetrical cryptosystem, asymmetrical cryptosystem, digital signature and one-way hash function mechanisms for persistent content protection, integrity, authentication, track usage of DRM work, changeable access right, however the in that time, the computing ability is limited by the mobile phone hardware, the scheme was not so efficient for multimedia content protection, and the efficiency is low. In 2010, CC Chang [18] found that Chen's scheme is insecure because the symmetric key can be easily computed by an attacker. In addition, tampering with the user's password cannot be discovered by the mobile user. Moreover, there are some redundant computations for user authentication in Chen's scheme, then proposed a new scheme, in CC

Table 9
DRMChain scheme comparison with related work [17–20].

No.	Scheme	Usage control	Dynamic key agreement	Phase	Client user side/P2P computation cost	Server(s) side/P2P computation cost	Total computation cost
1	Chen [17]	N/A	No	• Package • Registration • Authorization	– – $(5\|F(\cdot)\|+3)T_h + T_{sym}$	$T_{sym} + 5T_{pub}$ – $(3\|F(\cdot)\|+3)T_h + 2T_{pub}$	$(8\|F(\cdot)\|+6)T_h +$ $2T_{sym} + 7T_{pub}$
2	Chang et al. [18]	No	No	• Package • Registration • Authorization	– – $(3\|F(\cdot)\|+2)T_h + T_{sym}$	$T_{sym} + 5T_{pub}$ – $(3\|F(\cdot)\|+2)T_h + 2T_{pub}$	$(6\|F(\cdot)\|+4)T_h$ $+2T_{sym} + 7T_{pub}$
3	Chang et al. [19]	No	No	• Package • Registration • Authorization	– – $4T_h + T_{sym}$	$2T_h$ $8T_h$ $T_{sym} + 2T_{pub}$	$14T_h + 2T_{sym} + 6T_{pub}$
4	A. K. Das et al. [20]	Yes	No	• Package • Registration • Authorization	– $T_{fe} + 3T_h$ $T_{fe} + 7T_h + T_{sym}$	$T_{sym} + 2T_{pub}$ $T_h + T_{sym}$ $5T_h + T_{sym}$	$2T_{fe} + 16T_h + 4T_{sym} +$ $2T_{pub}$
5	DRMChain video content	Yes	Yes	• Package • Registration • Authorization	– $T_h + T_{pub}$ $T_h + T_{pub} + T_{ipfs} + T_{\mathrm{sym}}$	$T_{ipfs} + T_{sym}$ $T_h + T_{pub}$ $T_h + T_{pub} + T_{\mathrm{blk}}$	$4T_h + 4T_{pub} + 2T_{ipfs} +$ $2T_{sym} + T_{\mathrm{blk}}$
	DRMChain image content	Yes	Yes	• Package • Registration • Authorization	– $T_h + T_{pub}$ $T_h + T_{pub} + T_{ipfs} + T_{\mathrm{DCT}}$	$T_{ipfs} + T_{DCT}$ $T_h + T_{pub}$ $T_h + T_{pub} + T_{\mathrm{blk}}$	$4T_h + 4T_{pub} + 2T_{ipfs} +$ $2T_{DCT} + T_{\mathrm{blk}}$

Chang's scheme, the symmetric key was protected by a one-way hash function so it cannot be directly computed by an attacker. In addition, tampering with the transmitted message can be detected by the mobile users in the proposed scheme. Besides, the proposed scheme has no redundant computation for user authentication. Therefore, the proposed scheme is more efficient and reliable than Chen's scheme. Later in 2013, CC Chang [19] proposed an improved secure and efficient E-DRM mechanism based on a one-way hash function and exclusive or, which declared to overcome the weaknesses in the scheme of Chang et al., and also can reduces computation costs. However, In 2015, A. K. Das identified that Chang's scheme did not resist the insider attack and password-guessing attack [20]. In addition, Chang et al.'s scheme has some design flaws in the authorization phase, based on the analysis of Chang's scheme, A. K. Das proposed scheme supports the authorized content key distribution and satisfies the desirable security attributes. Additionally, Das' scheme offered low communication and computation overheads and user's anonymity as well.

The common attribute of the above schemes are centralized DRM solution, and provide fairly good security and performance of enterprise DRM solutions, however a most serious problems of the centralized DRM is once the DRM authorization server collapsed, the DRM system will not work again for large amount client users' request and cannot provide license and content services. While our proposed DRMChain scheme not only provided secure DRM services such as authorization and license in P2P mode which can overcome the above risk, but can provide blockchain-based rights proof and confirmation for each content, which can prevent content being violated or misused. And moreover, our proposed DRMChain scheme's total computation cost $4T_h + 4T_{pub} + 2T_{ipfs} + 2T_{sym} + T_{\mathrm{blk}}$ (video), $4T_h + 4T_{pub} + 2T_{ipfs} + 2T_{DCT} + T_{\mathrm{blk}}$ (image) are lower than the schemes [17–20], the detailed comparison with the related scheme are listed as Table 9.

Let T_{h}, T_{pub}, T_{sym}, T_{DCT} and T_{blk} denote the time complexity for computation of a one-way hash function H(.), a public key encryption/decryption/digital signature, a symmetric encryption/decryption, a watermark algorithm DCT transfer and a block creation respectively. From the comparison of variant DRM schemes with our proposed DRMChain scheme, we can see our proposed DRMChain supports usage control such as playing times, usage domain control, and our scheme is available for dynamic key agreement, which is efficient and secure and extendible for video and image content protection especially in P2P network environment.

8. Conclusion

Digital rights management is a traditional topic in network environment, in this paper we proposed a new paradigm based on blockchain for digital rights management, which supports the right digital rights-protected content serves the right users in a right way (thus we named DRMChain), the DRMchain can provide trusted and high-level credible content protection and conditional traceability of violation content service. In the proposed DRMChain, we use two isolated blockchain application interfaces (BAI) to respectively store plain and cipher summary information of original and DRM-protected digital content, and considering large capacity of digital content such as image, audio or video, we proposed external flexible storage of plain/cipher digital content and creates hashID of the content itself and links with the blockchain. In DRMChain scheme, we named the BAI plain interface as BAIP for summary metadata storage of original content, and the BAI cipher interface as BAIC for DRM-protected content service. In the DRMChain scheme we proposed efficient and secure authentication, privacy protection and multi-signature-based conditional traceability approaches, and thus the DRM license, usage control and constraint information can be easily retrieved form the blockchain, and customs can query all the consumption transaction lists of free or paid consumption history to prevent baleful fee-deduction. Analysis and performance evaluation manifest the DRMChain scheme provides a reliable, secure, efficient and tamper-resistance digital content service and DRM practice. In future, we will enhance the work that support Ethereum-based coin for digital rights management and trade that support the new promising vision: The right content serves the right users in a right way for the right value.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61272519, No. 61472258 the Research Funds of Blockchain Joint Lab between BUPT and BCT, China.

References

[1] A. Foroughi, M. Albin, S. Gillard, Digital rights management: A delicate balance between protection and accessibility, Inform. Sci. 28 (5) (2002) 389–395.
[2] D. Kundur, K. Karthik, Video fingerprinting and encryption principles for digital rights management, Proc. IEEE 92 (6) (2004) 918–932.

[3] D. Lindsay, S. Ricketson, Copyright, privacy, and digital rights management (DRM), in: New Dimensions in Privacy Law: International and Comparative Perspectives, Cambridge Univ. Press, New York, NY, USA, 2006, pp. 121–153 Eds.
[4] P. Koster, W. Jonker, Digital Rights Management, Vol. 25, Springer Berlin Heidelberg, 2007, pp. 225–235 No. 1.
[5] C.H. Huang, S.C. Chuang, Y.L. Huang, J.L. Wu, Unseen visible watermarking: a novel methodology for auxiliary information delivery via visual contents, IEEE Trans. Inf. Forensics Secur. 4 (2) (2009) 193–206.
[6] Alessandro Basso, Davide Cavagnino, et al., Blind watermarking of color images using Karhunen–Loève transform keying, Comput. J. 54 (7) (2011) 1076–1090.
[7] Deepayan Bhowmik, Charith Abhayaratne, Quality scalability aware watermarking for visual content, IEEE Trans. Image Process. 25 (11) (2016) 5158–5172.
[8] Javier Franco-Contreras, Gouenou Coatrieux, Robust watermarking of relational databases with ontology-guided distortion control, IEEE Trans. Inf. Forensics Secur. 10 (9) (2015) 1939–1952.
[9] Uhl Andreas, Andreas Pommer, Image and Video Encryption, Springer Press, 2005.
[10] Lini Abraham, Neenu Daniel, Secure image encryption algorithms: A review, Int. J. Sci. Technol. 2 (4) (2013) 186–189.
[11] N.K. Pareek, V. Patidar, K.K. Sud, Image encryption using chaotic logistic map, Image Vis. Comput. 24 (9) (2006) 926–934.
[12] S.J. Shyu, Image encryption by random grids, Pattern Recognit. 40 (3) (2007) 1014–1031.
[13] R. Lukac, K.N. Plataniotis, Bit-level based secret sharing for image encryption, Pattern Recognit. 38 (5) (2005) 767–772.
[14] Chang'e Dong, Color image encryption using one-time keys and coupled chaotic systems, Signal Process., Image Commun. 29 (5) (2014) 628–640.
[15] Osama Ahmed Khashan, Abdullah Mohd Zin, An efficient adaptive of transparent spatial digital image encryption, Procedia Technol. 11 (1) (2013) 288–297.
[16] Ferdinando Di Martino, Salvatore Sessa, Fragile watermarking tamper detection with images compressed by fuzzy transform, Inform. Sci. 195 (13) (2012) 62–90.
[17] C.L. Chen, A secure and traceable E-DRM system based on mobile device, Expert Syst. Appl. 35 (3) (2008) 878–886.
[18] C.C. Chang, J.H. Yang, D.W. Wang, An efficient and reliable E-DRM scheme for mobile environments, Expert Syst. Appl. 37 (9) (2008) 6176–6181.
[19] C.C. Chang, S.C. Chang, J.H. Yang, A practical secure and efficient enterprise digital rights management mechanism, Secur. Commun. Netw. 6 (8) (2013) 972–984.
[20] A.K. Das, D. Mishra, S. Mukhopadhyay, An anonymous and secure biometric-based enterprise digital rights management system, Secur. Commun. Netw. 8 (18) (2016) 3383–3404.
[21] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008. URL https://bitcoin.org/bitcoin.pdf.
[22] The Bitcoin Project. URL https://bitcoin.org.
[23] The Ethereum Project. URL https://www.ethereum.org.
[24] The Hyperledger Project. URL http://www.hyperledger.org.
[25] M.B. Taylor, The evolution of bitcoin hardware, Computer 50 (9) (2017) 58–66.
[26] S. Bag, S. Ruj, K. Sakurai, Bitcoin block withholding attack: Analysis and mitigation, IEEE Trans. Inf. Forensics Secur. 12 (8) (2017) 1967–1978.
[27] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2084–2123.
[28] Matevž Pustišek, Andrej Kos, Approaches to front-end IoT application development for the Ethereum Blockchain, Procedia Comput. Sci. 129 (2018) 410–419.
[29] K. O'Hara, Smart contracts - dumb idea, IEEE Internet Comput. 21 (2) (2017) 97–101.
[30] K. Alabi, Digital blockchain networks appear to be following metcalfe's law, Electron. Commer. Res. Appl. 24 (2017).
[31] E. Androulaki, A. Barger, V. Bortnikov, et al., Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains, 2018.
[32] V. Dhillon, D. Metcalf, M. Hooper, The Hyperledger Project, 2017.
[33] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharing protocol for open blockchains, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 17–30.
[34] A. Wright, P.D. Filippi, Decentralized Blockchain Technology and the Rise of Lex Cryptographia, Social Science Electronic Publishing, 2015.
[35] G. Zyskind, O. Nathan, A. Pentland, Decentralizing privacy: Using blockchain to protect personal data, in: IEEE symposium on Security and Privacy, 2015, pp. 180–184.
[36] A.E. Kosba, A.J. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The Blockchain model of cryptography and privacy-preserving smart contracts, in: IEEE symposium on security and privacy, 2016, pp. 839–858.
[37] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C.P.A. Ogah, Z. Sun, Blockchain-based dynamic key management for heterogeneous intelligent transportation systems, IEEE Internet Things J. 4 (6) (2017) 1832–1843.
[38] M. Vukolić, The quest for scalable Blockchain Fabric: Proof-of-Work vs. BFT replication, in: International Workshop on Open Problems in Network Security, 2015, pp. 112–125.
[39] A. Dorri, M. Steger, S.S. Kanhere, R. Jurdak, BlockChain: A distributed solution to automotive security and privacy, IEEE Commun. Mag. 55 (12) (2017) 119–125.
[40] R.M. Frey, P. Buhler, A. Gerdes, T. Hardjono, K.L. Fuchs, A. Ilic, The effect of a blockchain-supported, privacy-preserving system on disclosure of personal data, in: IEEE 16th International Symposium on Network Computing and Applications, NCA, 2017, pp. 1–5.
[41] E. Rescorla, Diffie-Hellman Key Agreement Method, Network Working Group, RFC2631.
[42] N. Kaur, R. Nagpal, Authenticated Diffie-Hellman key exchange algorithm, Int. J. Comput. Sci. Inf. Technol. 5 (4) (2014) 5404–5408.
[43] H. Orman, the OAKLEY Key Determination Protocol, Network Working, Group Request for Comments:2412.
[44] G. Ateniese, M. Steiner, G. Tsudikc, New multiparty authentication services and key agreement protocols, IEEE J. Commun. 18 (4) (2000) 628–639.
[45] O. Goldreich, Secure multi-party computation, Manuscript. Preliminary version, 1998.
[46] A. Boldyreva, Threshold signatures, multisignaturesand blind signatures based on the gap-diffiehellman-group signature scheme, in: Public Key Cryptography—PKC 2003, Springer, 2002, pp. 31–46.
[47] S.S.M. Chow, L.C.K. Hui, S.M. Yiu, K.P. Chow, Forward-secure multisignature and blind signature schemes, Appl. Math. Comput. 168 (2) (2005) 895–908.
[48] C. Claude, J. Stanisław, K. Jihye, T. Gene, Secure acknowledgment aggregation and multisignatures with limited robustness, Comput. Netw. 50 (10) (2006) 1639–1652.
[49] T.S. Wu, C.L. Hsu, ID-based multi-signatures with distinguished signing authorities for sequential and broadcasting architectures, Appl. Math. Comput. 131 (2) (2002) 349–356.
[50] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. 48 (177) (1987) 203–209.
[51] V.S. Miller, Use of elliptic curve in cryptography, in: Advances in Cryptology-CRYPTO'85, in: Lecture Notes in Computer Science, vol. 218, 1986, pp. 417–426.
[52] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ECDSA), Int. J. Inf. Secur. 1 (1) (2001) 36–63.
[53] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, second ed., John Wiley & Sons, Inc, 1995.
[54] OpenSSL. URL https://www.openssl.org.
[55] ANSI X9.62. Public Key Cryptography for the Finacial Service Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).1999.
[56] IEEE P1363. Standard Specifications for Public-Key Cryptography. IEEE. Standard.P1363, 2000.

Zhaofeng Ma, Ph.D. Degree, IEEE Member, CCF member. He engages in science research and education work in School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. He is now the director of Blockchain Joint Lab between BUPT–BCT. He received his Ph.D. degree from Xi'an Jiaotong University in 2004. He did his post-doctor research work in Tsinghua University during 2005–2007. Since 2007, he built up the research group and engaged in science research work in Beijing University of Posts and Telecommunications, His research interests include blockchain, mobile Internet innovation and security, digital rights management. He finished or presided over 12 research projects and built up 4 security-related Joint Labs (including BUPT–BCT Blockchain Joint Lab). He is now engaging in blockchain research and development work based on the popular blockchain platforms including Bitcoin, Ethereum and Hyperledger, and as the director, he guided and finished the 5 blockchain projects in BUPT–BCT Joint Lab. (Email: mzf@bupt.edu.cn).

Ming Jiang received the Ph.D. degree from Beijing University of Posts and Telecommunications in 2012. He is now an associate researcher in intelligent audio and video department and takes part in technological innovation in the Third Research Institute of China Electronics Technology Group Corporation. His research interests include digital watermarking, digital rights management. He finished more than 10 research projects of digital watermarking. (Email: jiangandming@aliyun.com).

Hong ming Gao is a Ph.D. candidate in School of Cyber Security, Beijing University of Posts and Telecommunications. His research interests include blockchain, applied cryptography and digital rights management. He finished the Blockchain platform of BUPT–BCT Joint Lab. (Email: gaohm@bupt.edu.cn).

Zheng Wang is a Ph.D. candidate in School of Cyber Security, Beijing University of Posts and Telecommunications. His research interests include blockchain, mobile Internet security and digital rights management. He participated and finished the Blockchain platform of BUPT–BCT Joint LAB, and mobile internet security projects of BUPT. (Email: wangzhen@bupt.edu.cn).