Article

Blockchain-Based Smart Propertization of Digital Content for Intellectual Rights Protection

Nam-Yong Lee ¹ , Jinhong Yang ^{2,*} and Chul-Soo Kim ³

¹ Department of Applied Mathematics and College of Artificial Intelligence, Inje University, Gimhae 50834, Korea; nylee@inje.ac.kr

² Department of Healthcare and Information Technology, Inje University, Gimhae 50834, Korea

³ Department of Computer Engineering, Inje University, Gimhae 50834, Korea; cskim@inje.ac.kr

* Correspondence: jinhong@inje.ac.kr; Tel.: +82-10-8518-4555

Abstract: Several platform companies have been successful in competing with digital piracy by producing consumer-friendly services. Nowadays, however, the digital content service market has become more monopolized than ever, which forms barriers to the entry of new platform companies. The platform monopoly would cause considerable problems both to content providers and consumers as it limits the choice of consumers. To remove the platform monopoly in digital content markets, we propose a public blockchain-based digital content service method. The proposed method encrypts the digital content to a self-decryptable form, which we call Smart Propertized Digital Content (SPDC), and utilizes the decentralization and traceability of the public blockchain to provide a non-monopolistic ecosystem for the management and distribution of the SPDC license. The proposed method can be more beneficial both to content creators and users than the current winner-takes-all platform model. For instance, users can download SPDC once and play many times without requesting the decryption key, and SPDC owners can make SPDC licenses to be time-limited, device-limited, resellable, or terminated without resorting to help from other intermediaries. We conducted the threat analysis on the proposed method by examining possible attacks in various scenarios. Based on threat analysis, we conclude that the proposed method can provide a new type of digital content service ecosystem that can be operated in a completely decentralized way and neutrally beneficial to all participants.

Keywords: Smart Propertized Digital Content (SPDC); license; self-decryption; immobilizer; multi-structured blockchain system

Citation: Lee, N.-Y.; Yang, J.; Kim, C.-S. Blockchain-Based Smart Propertization of Digital Content for Intellectual Rights Protection. *Electronics* **2021**, *10*, 1387. <https://doi.org/10.3390/electronics10121387>

Academic Editor: Paulo Ferreira

Received: 8 May 2021

Accepted: 7 June 2021

Published: 9 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The rapid development of information communication technologies (ICTs) in storing, managing, searching, and transmitting digital data has changed how people consume content such as book, music, video, and continues doing so. As of the year 2021, a great portion of content is now being stored in digital format and distributed and sold to users via the Internet, and digital data-specific services such as subscription-based streaming have become ever more popular. For instance, the International Federation of the Phonographic Industry (IFPI) 2019 annual report showed that global recorded music revenues posted growth of 8.2%, which was the fifth consecutive year of global growth, and the digital streaming revenue alone accounted for 56.1% of the global recorded music market after increasing by 22.9% in 2019 [1]. The report from Grand View Report also showed that the global digital video streaming market size was valued at USD 42.6 billion in 2019 and is projected to grow at a compound annual growth rate of 20.4% from 2020 to 2027 [2].

1.1. Problem Statement

As compared with analog content, digital content is easy to manage and suitable for massive distribution. Such advantages of digital content, however, become disadvantages

in the presence of digital piracy. Digital content can be copied identically to original versions and massively distributed to potential content consumers freely through the Internet. Most people are unlikely to pay for digital content that can be downloaded from the Internet or copied from one device to another. This has been a major problem for content industries such as the music, movie, and publishing industries. It has been a common belief that without legal and technological protection against digital piracy, content sales revenues will plummet, content creators will consequently not receive royalties and hesitate to make new content. Consequently, the content industries can be in jeopardy, and consumers will eventually suffer from the lack of high-quality content. Some research works [3–5] challenged this by claiming that the elimination of digital piracy would not transform illegal users to the content buyers substantially. They also claimed that digital piracy gives some ‘positive effects’ (e.g., an effective promotion and publicity vehicle, et cetera) to consumers and artists. The majority of research works, however, support the common belief [6–13].

To protect digital content against digital piracy, content industries have adopted various forms of the ‘digital rights management’ system (DRM). The DRM technology allows the content provider to restrict consumers to playing digital content under specific rules (such as on a streaming service only, a limited number of plays, or specified devices, et cetera). The DRM technology has contributed to the development of digital content services by alleviating widespread fear of digital piracy. Some consumers, however, criticize DRM for restricting their content usage under consumer-unfriendly rules.

Several platform companies have been successful in competing with digital piracy. Examples include the ebook service employed by Amazon and Barnes and Nobles, the Apple iTunes’ music download service, the music streaming by Spotify, and the video streaming by Netflix, et cetera. Those platform companies do not impose powerful DRM technologies to fight against digital piracy (but their DRM methods are strong enough to keep most consumers from attempting to make illegal content copies) but give consumer-friendly platform services to honest consumers. They vigorously use consumer data and the scale of economies to attract more consumers. Such efforts mean that digital content markets are dominated by few platforms nowadays. This phenomenon is often called the natural monopoly in digital content markets.

1.2. Key Contributions

To remove the natural monopoly in digital content markets, we propose a public blockchain-based digital content service method. To be specific, we use the decentralization and traceability of the public blockchain to provide a non-monopolistic content service ecosystem. In particular, contributions of the proposed digital content service model are as follows:

- The license management and payment collection of digital content can be performed in decentralized public blockchain networks.
- Digital content owners can have detailed control of the use condition. For example, they can make licenses which are time-limited, device-restricted, playing number-limited, resellable, or, if necessary, terminated, et cetera.
- Users can have more freedom in playing digital content. For example, they can download digital content once and play it many times or pay only for what they want to have or rent.

2. Background

2.1. Natural Monopoly in Digital Content Markets

There have been two opposite opinions on the platform monopoly in digital content markets. One opinion is that the market dominance by few platforms is a temporary phenomenon, and the seeming monopolies in current digital content markets will disappear once markets are mature. The other opinion is that the platform monopoly is a present danger in digital content markets and will cause serious problems not only to

content providers but also to consumers, unless proper antitrust regulations and technical improvements are introduced. This study is based on the latter opinion. Specific issues that motivate this study are as follows:

- **Platform monopoly:** Platform companies massively collect user data in the process of distributing digital content and use collected data vigorously in personalized content marketing. While the user data-driven personalized marketing can benefit users, it can also introduce switching costs for users. The more the platform company learns about its users' preferences, the more accurately it can market content based on their specific preferences. It makes a formidable barrier to entry for new platform companies since they do not have the user data necessary to compete with the dominant platform company. Platform companies also limit purchased content to their ecosystem by employing the DRM system. In such a case, users would prefer to have all their digital content on a single platform rather than spread it out on many platforms. It makes another formidable barrier to entry for new platform companies. It also makes it difficult for users to switch to other platforms since they want to continue to use purchased digital content. This is called the 'platform lock-in' effect [14]. Based on these arguments, there has been the warning (e.g., see [15]) that digital content platform markets tend to be naturally monopolized.
- **Dilemmas of content creators:** Digital content platforms have benefited content creators by providing new kinds of distribution channels. With these channels, creators can distribute and advertise their content to customers without using the channels provided by content companies, which often demand a high price. The natural monopoly in content platform markets, however, has introduced significant risk to content creators. Nowadays, it is not unusual to hear complaints of content creators about platforms, for taking too large a portion of revenues or not giving clear log information which determines the revenue share of content creators. As individual sellers, content creators are more vulnerable than content companies to predatory business practices performed by platforms.
- **Dilemmas of content users:** In the initial days, platforms rapidly gained user acceptance by providing digital content, with low prices and data-driven personalized services. Such benevolence given by platforms is brought to users not only by technological improvement but also by sacrifices in content companies and creators. Those sacrifices will make content companies and creators hesitate to make high-quality digital content and hence will eventually harm users.

2.2. Related Works

There have been several approaches that use blockchain technology in protecting intellectual rights management or providing new ecosystems for digital content services. Examples are as follows:

- Ujo music [16] uses the Ethereum blockchain technology [17] to keep track records of digital content owners and assign a unique identity to each uploaded music. The payment process in Ujo music is carried out by Ethereum smart contracts. In Ujo music, however, digital content is not protected against illegal copying.
- Resonate music [18] provides subscription-based music streaming services and uses blockchain technology to make digital music owners take participate in publishing their music and managing royalties on them.
- The platform proposed by SingularDTV [19], which describes itself to be a decentralized 'Netflix' on Ethereum, is somewhat similar to the proposed method of this paper. The difference is in the fact the SingularDTV platform is designed to charge for each streaming, while the proposed method of this paper allows digital content owners to charge as they wish. For instance, they can demand the payment for each watch, a fixed rate for a limited number of views or a limited period, et cetera. It is also true that the proposed platform of this paper is not restricted to streaming services, unlike SingularDTV.

- Methods proposed in [20,21] are also quite similar to the proposed method of this paper. The method in [20] is designed to identify malicious users, with the hope that this kind of identification might discourage attempts at malicious activity from users. On the other hand, the method in [21], however, allows the user to access the digital content decryption key. In other words, the user can perform the duplication of decrypted digital content. This is not good. This kind of scenario is not allowed in our method.
- Methods in [22–25] proposed to store the digital watermark of contents in blockchain to implement copyright registration. For instance, the method in [22] proposed a watermarking-based multimedia blockchain framework to provide a trusted mechanism to retrieve the transaction trails or the modification histories on digital content. The study in [23] proposed an artwork image DRM for Internet misusing detection. The method in [24] combined digital watermarking, blockchain, perceptual hash function, quick response code, and InterPlanetary File System (IPFS) for copyright management. Among them, blockchain was used to securely store watermark information and provide timestamp authentication for multiple watermarks to confirm the creation order. The method in [25] used digital fingerprint technology, an Inter-planetary File System (IPFS), and blockchain technology to create a digital system to optimize traditional processes and improve the efficiency of digital media copyright solidification.
- Methods in [26–31] used the blockchain for secure data sharing. For instance, the method in [26] proposed a privacy-preserving DRM system. The method in [27] proposed a data sharing protocol over different DRM systems. The study in [28] proposed a blockchain-based system to implement a superdistribution-based ideal content distribution system. The methods in [29] used blockchain technology to enable the user to have full control over their multimedia files such as storing, querying, sharing, and auditing data. The study in [30] proposed a blockchain-enabled DRM system to share and manage multimedia resources of online education, while the method in [31] used blockchain technology to share and reuse scholars' datasets securely.
- Methods in [32–35] proposed blockchain-based DRM systems for various content types for secure data sharing. To be specific, the study in [32] used digital watermarking and a scalable blockchain model. The study in [33] proposed a blockchain-based solution for the identification, attribution, and payment for digital work. The study in [34] proposed an integrated trading system, to provide a transparent privacy-preserving, and tamper-proof transaction history for registration, provenance, and traceability of art assets. The method in [35] proposed a blockchain-based scheme for digital rights management, to provide trusted and high-level credible content protection and conditional traceability of violation content service.
- There were several hybrid blockchain-based approaches for the copyright protection of digital content. The method in [36] proposed a new zero watermarking construction method based on the angular features of vector data that store the zero watermarking and copyright information on the blockchain. The study in [37] explained blockchain and tangle technologies, along with an extended application to the copyright protection of digital content. The study in [38] proposed multi-authority attribute-based encryption (MA-ABE) schemes to make a relevant user obtain the final decryption attribute key only after all attribute authorities have issued their keys to the blockchain publicly.

One of the main differences between the described examples and our method is in the way how the proposed blockchain is used. To be specific, those described examples use blockchain as a database to record ownership or identity of digital content or as a payment collecting system. Their blockchains, however, are not the main reason for the prevention the illegal copying of digital content. For example, in [19] the illegal copying of digital content can be suppressed not by blockchain but by the streaming service. Similarly, in [22–25] blockchains do not prevent illegal copying, but the possibility of legal accusation with a watermark as proof of illegal use does. On the other hand, our method uses the

blockchain as a virtual machine to determine which digital content can be used by whom and how. In other words, in our method, the blockchain itself makes illegal copying of digital content useless.

3. Proposed Method

The proposed method uses a blockchain system to allow digital content creators to directly control the whole procedure of content distribution. To do so, the proposed method encrypts digital content to a self-decryptable form and makes the content player follows the use condition recorded in the system of blockchains.

The proposed ecosystem consists of the following three components:

- Smart Propertized Digital Contents (SPDC);
- Manager Program (MP);
- Hierarchical Multi-Blockchain System (HMBS).

The proposed method of this paper transforms the digital content to SPDC which can be self-decryptable and playable only by the MP with the HMBS. To be specific, the HMBS stores the license information of SPDC, the MP sends a signal to the SPDC according to license information of the SPDC in the HMBS (here the term ‘license’ is used to specifically mean ‘the right to play a specific SPDC’), and SPDC continues to play or stops the self-decryption based on the received signal from the MP.

The HMBS itself consists of AC (accountchain), MC (mainchain), and multiple pairs of SC (sidechain) and dSC (dual sidechain). Roughly speaking, the proposed method encapsulates the digital content in SPDC, and defines the MP to follow the SPDC license rule recorded in the AC, and provides the integrity of AC by using SCs, dSCs, and MC.

This section is outlined as follows. In Section 3.1 we shall explain how SPDC is formed. In Section 3.2 we shall explain the structure of the HMBS, which consists of MC, AC, multiple SCs and dSCs. Finally, in Section 3.3 we shall explain how the SPDC license is bought, transferred, or terminated through the HMBS.

In this study, we shall use the following abbreviations in Table 1.

Table 1. A list of abbreviations.

Abbreviation	Description
SPDC	Smart Propertized Digital Content
MP	Manager Program
HMBS	Hierarchical Multi-Blockchain System
MC	Mainchain
AC	Accountchain
SC	Sidechain
dSC	Dual Sidechain

In this study, we shall assume that entities listed in Table 2 participate in the HMBS:

Table 2. A list of notations that represent participants.

Notation	Description
U, V, W, Y	Users in the HMBS
P	Content owner in the HMBS
D	SPDC storage provider

In Table 2, the SPDC storage provider D is not confined to rules defined by the HMBS and can be any ordinary data storage provider on the Internet. In other words, we assume

that SPDC (the self-decryptable encrypted version of digital content) itself is freely available to anyone through the Internet.

3.1. Smart Propertized Digital Content

Let m be a digital content to be protected under the HMBS, and P be the owner of m . To protect m from the illegal copying, the proposed method makes P take the following step:

$$P : m, K \rightarrow c = E(K, m), \quad (1)$$

where c is the encrypted data by a symmetric encryption function E (for example, Advanced Encryption Standard (AES) [39]) with a secret key K . In (1), P is included to indicate that P performs the encryption process of computing c from m and K . The secret key K is randomly selected by P and used as input in the encryption process.

A standard form of many DRM systems sells digital content in an encrypted format (for, e.g., c in (1)) and controls secret keys (for, e.g., K in (1)) to be used only by content users without disclosing them to anyone, including content users. A weak point of this approach is the fact that it is not easy to hide secret keys from content users who need them for their legitimate use. Many approaches have been proposed to solve this dilemma. One of them is the streaming service, which allows only a small portion of digital content to be available to users and hence does not need to provide whole encrypted digital content and corresponding secret keys to users. The streaming technology is, however, not perfect in removing digital piracy, just as in any DRM technology ever introduced so far.

To deal with the weak point in secret key transfer from the content owner P to the user U , the proposed method makes P compute a secret key embedded decryption function D_K by replacing many computation parts of the decryption with transformation by randomized lookup tables (it is an example of white-box cryptography technologies [40,41]) and by implementing reverse engineering resilient obfuscated codes [42]. The process of making D_K is described as follows:

$$P : K, r_1 \rightarrow D_K, \quad (2)$$

where r_1 is a random number chosen by P to secretly hide K in D_K .

The proposed method puts the encrypted digital content c , the decryption function D_K , and a control program which initiates the use of D_K on c only under predetermined conditions in one package. This control program is called the ‘immobilizer’ and denoted by IM from now on. The package formed by c , D_K , and IM is called ‘smart propertized digital content’ (SPDC) of m . The process of making SPDC s is described as follows:

$$P : c, D_K, \text{IM}, r_2 \rightarrow s, \quad (3)$$

where r_2 is a random number chosen by P to make it infeasible to distinguish c , D_K , and IM in s .

The computation of SPDC s of a digital content m can be carried out consecutively:

$$P : m, K, r_1, r_2 \rightarrow s = \text{SP}(K, r_1, r_2, m), \quad (4)$$

where SP is the program that performs computations in (1)–(3) consecutively. The content owner P of the digital content m will use SP to make SPDC s of m by selecting K, r_1, r_2 randomly. The proposed method does not require P to store K, r_1, r_2 . Those random numbers are disposable once they are used.

Figure 1 illustrates a SPDC s (the one that has c , IM, and D_K), where m_i is a decrypted bitstream by D_K . It illustrates that IM controls the decryption of s by communicating only with MP.

Figure 1. The structure of SPDC s , where c is the decrypted content (1), IM is the immobilizer, D_K is the key hidden decryption function (2), and m_i is a decrypted bitstream by D_K .

3.2. Hierarchical Multi-Blockchain System

As mentioned earlier, the HMBS itself consists of AC (accountchain), MC (mainchain), and multiple pairs of SC (sidechain) and dSC (dual sidechain). The main role of each blockchain is summarized as follows:

- MC records underlying cryptocurrency (we shall call it ‘HMBS coin’) transfer transactions and controls the initiation and connection of ‘segment’ (we shall explain it in Section 3.2.3) of multiple SCs and dSCs.
- AC records the SPDC license use condition to determine who can play SPDC and how.
- SC and dSC validate the creation, transfer, and termination of the SPDC license.

The proposed method uses HMBS coins recorded in MC only. In other words, the proposed method regards transactions in SC or dSC segments as valid only after they are connected to MC.

Figure 2 illustrates components of the HMBS. The block chaining process in MC controls the initiation and connection of multiple SC and dSC segments. AC expands its blocks independently from MC.

Figure 2. Components of the HMBS: MC, AC, SCs and dSCs.

3.2.1. Mainchain

We assume that the MC network appends a new block to MC every certain time on average. We call that certain time the ‘MC block period’ and the corresponding time duration ‘MC epoch’. We now denote the MC epoch by $\mathbb{R}_{MC}^{[n]}$ with the order index n ($n = 0, 1, \dots$), the MC block by $B^{[n]}$, and the updated MC by $MC^{[n]}$. This process can be described as follows: when the MC epoch $\mathbb{R}_{MC}^{[n]}$ starts, MC in each node is $MC^{[n-1]}$ and miners (nodes that try to find new blocks) compete in computing the new block. Afterwards, the first miner who successfully computes a new block $B^{[n]}$ announces it to the MC network. Other nodes show their consensus on the announced block $B^{[n]}$ by updating their MCs (which are all assumed to be equal to $MC^{[n-1]}$) to $MC^{[n]}$ by appending $B^{[n]}$ to $MC^{[n-1]}$ after checking its validity (For details, e.g., see Section 5 in [43]):

$$MC^{[n]} = MC^{[n-1]} || B^{[n]}, \quad (5)$$

where the notation ‘||’ is the concatenation.

We shall explain the mining process of MC in the following setting: The current MC epoch is $\mathbb{R}_{MC}^{[n]}$, a miner collects transactions $t_1^{[n]}, \dots, t_{c^{[n]}}^{[n]}$ to record them in MC, and $h^{[n-1]}$ is the hash value of the latest block $B^{[n-1]}$. The mining of the next block $B^{[n]}$ needs to find a nonce r that makes

$$H(h^{[n-1]} || t_1^{[n]} || \dots || t_{c^{[n]}}^{[n]} || r) \in \mathcal{C}_{MC}, \quad (6)$$

where H is the cryptographic hash function SHA-256 [44], $c^{[n]}$ is the number transactions to be collected in $B^{[n]}$, and $\mathcal{C}_{MC}$ represents ‘a condition’. Throughout this paper, we shall use the notation ‘ $\in$ ’ to indicate that the left term ‘satisfies’ the condition written in the right hand side. The condition $\mathcal{C}_{MC}$ is imposed by the consensus algorithm of MC. For example, if the ‘Proof of Work’ consensus algorithm is used, then the condition $\mathcal{C}_{MC}$ demands the

hash value in (6) be smaller than a fixed small number (for details, e.g., see Section 2.4 in [45]).

The hash value of the block $B^{[n]}$ is defined by

$$h^{[n]} = H(h^{[n-1]} || t_1^{[n]} || \dots || t_{c[n]}^{[n]} || r^{[n]}) \quad (7)$$

with a nonce $r^{[n]}$ which satisfies (6).

3.2.2. Accountchain

After the content owner P transforms their digital content m to SPDC s as described in Section 3.1, they request the creation of the account of s to the AC network. Table 3 shows a list of information stored in the account of SPDC s .

Table 3. A list of information stored in the account of SPDC s owned by P .

Notation	Description
$H(s)$	hash value of SPDC s
$\text{pk}(P)$	public key of P (owner of s)
$\mathcal{R}_s$	license usage rule of s
w	target value
q	input value
$\mathcal{Z}_s$	registration key set

In Table 3, $H(s)$, the hash value of SPDC s , can be used as an identification number of s . The public key $\text{pk}(P)$ of P defines the owner of s and the license fee receiving address in MC. In this work, we assume that the user must play s under a certain rule. The notation $\mathcal{R}_s$ in Table 3 represents that certain rule. The owner P of s must determine what rules are to be imposed on the use of s and write them in $\mathcal{R}_s$ as readable codes by smart contracts of the HMBS. Examples of usage rules include the maximum number of plays, the maximum number of devices, the time duration to play, the transferability of license (a user can resell his license to other users), et cetera. The set $\mathcal{Z}_s$ in Table 3 contains so-called ‘registration keys’. These keys are used as the user identity in purchasing the license of SPDC s . Detailed explanation will be given in Section 3.3.1.

When the account of s is created for the first time, the target value w in Table 3 is computed by

$$w = H_\lambda(H(s) || \text{pk}(P) || 0 \cdots 0), \quad (8)$$

where H_λ is the first λ -bit truncation of the SHA-256 hash function H , where $1 \leq \lambda \leq 256$, and $0 \cdots 0$ is the all zero bitstream of length 512 bits. The input value q in Table 3 is a bitstream of length 256 bits designed to make

$$w = H_\lambda(H(s) || \text{pk}(P) || q || H(\mathcal{Z}_s)) \quad (9)$$

holds for each non-empty registration key set $\mathcal{Z}_s$. In other words, q is said to be the input value of $\mathcal{Z}_s$ for s if and only if q satisfies (9). The parameter λ in (9) determines how difficult the computation of the input value q is. For some λ , say $\lambda \approx 70$, finding an input value q for a given non-empty set $\mathcal{Z}_s$ would be a very difficult task, but it can be done with collective computation, as shown in block mining in Bitcoin.

Each element of the registration key set has the following form:

$$\tau = \text{the registration key : hash value } h, \quad (10)$$

where h is the hash value of the SC block containing the transaction that requests the inclusion of τ in (10) to $\mathcal{Z}_s$. The inclusion of h in (10) is designed not to have head start in computing the input value q . More detailed explanation will be given in Section 3.3.3.

The proposed method uses AC, which has a modifiable blockchain structure [46], to record SPDC account information in Table 3. The modifiability of AC is used to efficiently record frequently changing account information, for example, $\mathcal{Z}_s$. Just as in MC, we assume that the AC network appends a new block to AC every certain time on average, and define terms ‘AC block period’, ‘AC epoch’, and ‘AC block’ accordingly. We also denote AC epoch by $\mathbb{R}_{AC}^{[n]}$ with the order index n ($n = 0, 1, \dots$), AC block by $A^{[n]}$, and AC at the moment when $A^{[n]}$ is connected to AC by $\mathbb{AC}^{[n]}$.

The AC block records the information of multiple accounts. The process of block mining in AC can be similar to that in MC. Suppose that the current AC epoch is $\mathbb{R}_{AC}^{[n]}$, a miner collects an account representing bitstreams $\mu_1^{[n]}, \dots, \mu_{\tilde{c}^{[n]}}^{[n]}$ (each $\mu_i^{[n]}$ represents the account of one SPDC), where

$$\mu_i^{[n]} = H(s) \parallel \text{pk}(P) \parallel \mathcal{R}_s \parallel w \quad (11)$$

and $\tilde{c}^{[n]}$ is the number of accounts representing bitstreams to be recorded in $A^{[n]}$, and $g^{[n-1]}$ is the hash value of the latest block $A^{[n-1]}$ in AC. The mining of the next block $A^{[n]}$ needs to find a nonce ρ that makes

$$H(g^{[n-1]} \parallel \mu_1^{[n]} \parallel \dots \parallel \mu_{\tilde{c}^{[n]}}^{[n]} \parallel \rho) \in \mathcal{C}_{AC}, \quad (12)$$

where $\mathcal{C}_{AC}$ represents the condition determined by the consensus algorithm of AC. The hash value of the block $A^{[n]}$ is defined by

$$g^{[n]} = H(g^{[n-1]} \parallel \mu_1^{[n]} \parallel \dots \parallel \mu_{\tilde{c}^{[n]}}^{[n]} \parallel \rho^{[n]}) \quad (13)$$

with a nonce $\rho^{[n]}$ which satisfies (12).

The modifiability of AC comes from (11). Suppose that we want to modify $\mathcal{Z}_s$ to $\tilde{\mathcal{Z}}_s$ to respond to a new license purchase of SPDC s . If we can find q that satisfies (9) with $\tilde{\mathcal{Z}}_s$ in the place of $\mathcal{Z}_s$, then we can replace $\mathcal{Z}_s$ with $\tilde{\mathcal{Z}}_s$ in the account of SPDC s without destroying the chaining structure (imposed by (13)) in AC.

When an AC block is mined, all accounts in it are assumed to be created for the first time. The proposed method uses a separate cryptocurrency for AC. The AC-specific cryptocurrency will be used for the AC block reward and SPDC account creation request fee. It is also possible that the AC-specific cryptocurrency can be designed to be exchangeable with the HMBS coin at a fixed rate.

3.2.3. Sidechain and Dual Sidechain

The proposed method uses multiple pairs of SC (sidechain) and dSC (dual sidechain) to validate the creation, transfer, and termination of the SPDC license. We assume that there is a predetermined rule assigning each SPDC license to be validated through only one SC and dSC pair.

Let us use the notation $\mathbb{S}_\sigma$ to indicate one of the SCs with σ as an SC identifier. In the proposed method, each SC $\mathbb{S}_\sigma$ consists of a series of ‘segments’, and each segment consists of ‘SC blocks’. We denote the n -th segment of $\mathbb{S}_\sigma$ by $\mathbb{S}_\sigma^{[n]}$, and the j -th SC block of the segment $\mathbb{S}_\sigma^{[n]}$ by $b_\sigma^{[n,j]}$.

The proposed method uses a predetermined rule based on tail bits of MC block hash value $h^{[n]}$ to select one SC in which the old segment (being mined in the SC network) is to be terminated and a new segment is to be initiated. To explain this, let us assume that the hash value $h^{[n_i]}$ of the MC block $B^{[n_i]}$, based on the predetermined rule, initiates the n -th

segment $\mathbb{S}_\sigma^{[n]}$ of $\mathbb{S}_\sigma$. Let us also assume that $B^{[n_{i+1}]}$ is the next MC block that initiates the next segment $\mathbb{S}_\sigma^{[n+1]}$ of $\mathbb{S}_\sigma$. Let

$$\mathbb{S}_\sigma^{[n]} = b_\sigma^{[n,0]}, b_\sigma^{[n,1]}, \dots, b_\sigma^{[n,k]} \quad (14)$$

be SC blocks mined in the $\mathbb{S}_\sigma$ SC network during the time interval that $h^{[n_i]}$ to $h^{[n_{i+1}]}$ are computed in the MC network.

In the proposed method, each SC must have its dual dSC. Let us use the notation $\hat{\mathbb{S}}_\sigma$ to indicate the dSC corresponding to $\mathbb{S}_\sigma$. Just like $\mathbb{S}_\sigma$, each dSC consists of a series of ‘segments’, and each segment consists of ‘dSC blocks’. We denote the n -th segment of $\hat{\mathbb{S}}_\sigma$ by $\hat{\mathbb{S}}_\sigma^{[n]}$, and the j -th dSC block of the segment $\hat{\mathbb{S}}_\sigma^{[n]}$ by $\hat{b}_\sigma^{[n,j]}$. Let

$$\hat{\mathbb{S}}_\sigma^{[n]} = \hat{b}_\sigma^{[n,0]}, \hat{b}_\sigma^{[n,1]}, \dots, \hat{b}_\sigma^{[n,\hat{k}]} \quad (15)$$

be dSC blocks mined in the $\hat{\mathbb{S}}_\sigma$ -SC network during the time interval that $h^{[n_i]}$ to $h^{[n_{i+1}]}$ are computed in the MC network.

Let us assume that the block chaining in the $\mathbb{S}_\sigma$ SC segment has the same structure in MC. To be specific, the mining of the $b_\sigma^{[n,i]}$ SC block in the $\mathbb{S}_\sigma^{[n]}$ SC segment needs to find a nonce that makes

$$H(h_\sigma^{[n,i-1]} || t_1^{[\sigma,n,i]} || \dots || t_{c^{[\sigma,n,i]}}^{[\sigma,n,i]} || r) \in \mathcal{C}_{\mathbb{S}_\sigma}, \quad (16)$$

where $c^{[\sigma,n,i]}$ is the number transactions to be included in $b_\sigma^{[n,i]}$, $h_\sigma^{[n,i-1]}$ is the hash value of the SC block $b_\sigma^{[n,i-1]}$, and $\mathcal{C}_{\mathbb{S}_\sigma}$ represents the condition determined by the consensus algorithm of $\mathbb{S}_\sigma$ SC. The hash value of the SC block $b_\sigma^{[n,i]}$ is defined by

$$h_\sigma^{[n,i]} = H(h_\sigma^{[n,i-1]} || t_1^{[\sigma,n,i]} || \dots || t_{c^{[\sigma,n,i]}}^{[\sigma,n,i]} || r^{[\sigma,n,i]}) \quad (17)$$

with a nonce $r^{[\sigma,n,i]}$ which satisfies (16).

The block chaining structure of the dSC segment is almost identical to that of the SC segment. One difference is that the dSC block records only one ‘smart contract’, while the SC block records multiple transactions. Detailed explanation will be given in Section 3.3.3.

The SC segment $\mathbb{S}_\sigma^{[n]}$ and dSC segment $\hat{\mathbb{S}}_\sigma^{[n]}$ are connected to MC starting from $B^{[n_i]}$ and ending at $B^{[n_{i+1}]}$ as follows:

- When $h^{[n_i]}$ is announced, miners in $\mathbb{S}_\sigma$ and $\hat{\mathbb{S}}_\sigma$ networks stop block mining for $\mathbb{S}_\sigma^{[n-1]}$ and $\hat{\mathbb{S}}_\sigma^{[n-1]}$, and initiate the block mining for the new SC segment $\mathbb{S}_\sigma^{[n]}$ and dSC segment $\hat{\mathbb{S}}_\sigma^{[n]}$ starting from $B^{[n_i]}$, by making $B^{[n_i]}$ be the first SC and dSC blocks in $\mathbb{S}_\sigma^{[n]}$ and $\hat{\mathbb{S}}_\sigma^{[n]}$:

$$b_\sigma^{[n,0]} = \hat{b}_\sigma^{[n,0]} = B^{[n_i]}. \quad (18)$$

- When $h^{[n_{i+1}]}$ is announced, the miner of $B^{[n_{i+1}]}$ connects SC segment $\mathbb{S}_\sigma^{[n]}$ and dSC segment $\hat{\mathbb{S}}_\sigma^{[n]}$ to MC by updating the hash value $h^{[n_{i+1}]}$ as

$$h^{[n_{i+1}]} \leftarrow h^{[n_{i+1}]} || h_\sigma^{[n,k]} || \hat{h}_\sigma^{[n,\hat{k}]}, \quad (19)$$

where $h_\sigma^{[n,k]}$ and $\hat{h}_\sigma^{[n,\hat{k}]}$ are hash values of the last blocks in $\mathbb{S}_\sigma^{[n]}$ and $\hat{\mathbb{S}}_\sigma^{[n]}$, respectively.

3.3. Creation, Transfer, and Termination of SPDC License

In this section, we shall explain how the SPDC license is created, transferred, and terminated through the HMBS.

3.3.1. Registration Key

We use the term ‘registration key’ to refer to a public key associated with a fixed amount of HMBS coins. The proposed method uses the registration key as the SPDC license user identity. The fixed amount of HMBS coins assigned to the registration key is called ‘registration fee’. For simplicity of presentation, we assume that the registration key itself is ‘registered’ to make users not transfer HMBS coins to it. We also assume that the registration fee can be spent just like any other HMBS coins.

The proposed method follows the UTXO (Unspent Transaction Output) model of Bitcoin in recording transactions related to HMBS coin transfer. In such a case, the ‘address’ (the hash value of the public key), instead of the public key itself, is used as the identity of the payer or the receiver. In this paper, however, we shall ignore the described difference in the public key and the address for simplicity of presentation.

3.3.2. SPDC License

We shall use following notations to represent descriptions written on the right hand sides: (U : a user)

$$\text{rk}(U) = \text{a registration key of } U, \quad (20)$$

$$\widehat{\text{rk}}(U) = \text{the private key paired with } \text{rk}(U), \quad (21)$$

$$\mathcal{L}_s = \text{the license of SPDC } s, \quad (22)$$

$$\Lambda_{\mathcal{L}_s, \text{rk}} = \text{the creation of } \mathcal{L}_s \text{ assigned to } \text{rk}, \quad (23)$$

$$\Lambda_{\mathcal{L}_s, \text{rk}, \text{rk}'}^* = \text{the transfer of } \mathcal{L}_s \text{ to } \text{rk}' \text{ from } \text{rk}, \quad (24)$$

and

$$\Lambda_{\mathcal{L}_s, \text{rk}}^\# = \text{the termination of } \mathcal{L}_s \text{ assigned to } \text{rk}. \quad (25)$$

Let us consider the case when a user U tries to buy $\mathcal{L}_s$ from an owner P of s by using his registration key $\text{rk}(U)$. To fulfill this, the proposed method makes U to request the following transaction t is to be recorded in one of the SCs, say, $\mathbb{S}_\sigma$ SC:

$$t: \text{pk}(U) \text{ pays } C \text{ to } \text{pk}(P) \text{ for } \Lambda_{\mathcal{L}_s, \text{rk}(U)}, \quad (26)$$

signed by $\text{sk}(U)$,

where C is the price for $\mathcal{L}_s$. The proposed method uses the recording of t in $\mathbb{S}_\sigma$ SC as a necessary condition for the change in the SPDC account that allows the user U to play s .

Let us consider another case when a user V tries to resell the SPDC license $\mathcal{L}_s$ assigned to his registration key $\text{rk}(V)$ to another user W by reassigning $\mathcal{L}_s$ to $\text{rk}(W)$, a registration key of W . Here we assume that P , the owner of s , made $\mathcal{L}_s$ transferable when they made s from the beginning. They can specify rules for when to create, transfer, or terminate SPDC license of s in the license rule $\mathcal{R}_s$ in Table 3.

To fulfill the transfer of $\mathcal{L}_s$ between users V and W , the proposed method makes V and W request the following transaction t^* to be recorded in $\mathbb{S}_\sigma$ SC:

$$t^*: \text{pk}(W) \text{ pays } C^* \text{ to } \text{pk}(V) \text{ for } \Lambda_{\mathcal{L}_s, \text{rk}(V), \text{rk}(W)}^*, \quad (27)$$

signed by $\text{sk}(W)$ and $\text{sk}(V)$,

where C^* is the price to be paid by W to V as the license transfer fee. Here we note that $\mathbb{S}_\sigma$ SC must be the same SC where the transaction that generates $\Lambda_{\mathcal{L}_s, \text{rk}(V)}$ was recorded. The proposed method uses the recording of t^* in $\mathbb{S}_\sigma$ -SC as a necessary condition to change the SPDC account of s so that W is allowed to play s , while V cannot play s any longer.

Finally, we consider the case where an HMBS participant, including the owner P of s , tries to terminate the SPDC license $\mathcal{L}_s$ assigned to a registration key $\text{rk}(Y)$ of a user Y , due to license misuse. Various misuses can cause the termination of the SPDC license. Among them, we shall consider the registration key sharing case here. In the proposed method, to play s , Y needs the private key $\widehat{\text{rk}}(Y)$ paired with the registration key $\text{rk}(Y)$

(see Section 3.3.4). Therefore, if Y discloses $\widehat{\text{rk}}(Y)$ to another user, say, Y' , then Y' can play s without paying the license fee to P , the owner of s . At the same time, Y' can spend the registration fee assigned to $\text{rk}(Y)$. Such a spending is easily detectable in the proposed HMBS. To punish this type of license misuse, anyone who finds the registration key sharing can request the termination of $\text{rk}(Y)$ by submitting the following transaction $t^\sharp$ to the $\mathbb{S}_\sigma$ SC network:

$$t^\sharp: \text{terminate } \text{rk}(Y) \text{ in } \mathcal{L}_s \parallel \text{'evidence'}, \quad (28)$$

where the 'evidence' is the location of the transaction that spent the registration fee assigned to $\text{rk}(Y)$. Here we also note that $\mathbb{S}_\sigma$ SC must be the same SC where the transaction that generates $\Lambda_{\mathcal{L}_s, \text{rk}(Y)}$ was recorded.

Notice that the user Y' , by knowing $\widehat{\text{rk}}(Y)$, can play not only s but also any SPDCs whose licenses are assigned to $\text{rk}(Y)$. Thus, the termination of $\text{rk}(Y)$ must be requested for all SPDCs whose licenses are assigned to $\text{rk}(Y)$. However, it might require a large number of registration key sets to be changed. As explained in Section 3.2.2, it would be extremely difficult, if not impossible, to find inputs values q 's in (9) for a large number of registration key set changes. Considering this fact, the proposed method makes one transaction produce one request of terminating $\text{rk}(Y)$ in each SPDC license $\mathcal{L}_s$. The termination of $\text{rk}(Y)$ in other SPDCs whose licenses are assigned to $\text{rk}(Y)$ can be completed as time progresses.

3.3.3. Smart Contracts in Dual Sidechains

Without loss of generality, we now assume that transactions t in (26), t^* in (27), and $t^\sharp$ in (28) are recorded in the i -th $\mathbb{S}_\sigma$ block $b_\sigma^{[n,i]}$ of the n -th segment $\mathbb{S}_\sigma^{[n]}$ of $\mathbb{S}_\sigma$ SC, and $\mathbb{S}_\sigma^{[n]}$ itself was started from MC-block $B^{[n_i]}$. The miner who connects $b_\sigma^{[n,i]}$ to $\mathbb{S}_\sigma^{[n]}$ will receive the $\mathbb{S}_\sigma$ block reward and a part of the transaction fees for recording t and t^* transactions (after $\mathbb{S}_\sigma^{[n]}$ is successfully connected to MC). The remaining part of transaction fees will be given to the miner who connects $\mathbb{S}_\sigma^{[n]}$ to MC.

When the n -th segment $\mathbb{S}_\sigma^{[n]}$ ends at the MC block $B^{[n_{i+1}]}$, all transactions recorded in $\mathbb{S}_\sigma$ blocks of $\mathbb{S}_\sigma^{[n]}$ segment are almost recorded in MC. The realization of those transactions is firmly guaranteed as more blocks are appended to MC after $B^{[n_{i+1}]}$. Notice that the recording of HMBS coin transfers in transactions t and t^* can be almost completed at the moment when the n -th segment $\mathbb{S}_\sigma^{[n]}$ is connected to $B^{[n_{i+1}]}$ in MC, but requests of changing ownership of $\mathcal{L}_s$ as in t , t^* , and $t^\sharp$ are not completed yet. To be specific, the transaction t in (26) demands the inclusion of $\text{rk}(U)$ in $\mathcal{Z}_s$ in Table 3 so that MP of U gives the 'continue-to-play' signal to IM (immobilizer) of s after verifying that U has $\widehat{\text{rk}}(U)$, the private key paired with the registration key $\text{rk}(U)$. In other word, $\text{rk}(U)$ is not yet included in $\mathcal{Z}_s$ which determines who can play s . Similarly, the transaction t^* in (27) demands the inclusion of $\text{rk}(W)$ and the exclusion of $\text{rk}(V)$ in $\mathcal{Z}_s$, and the transaction $t^\sharp$ in (28) demands the exclusion of $\text{rk}(Y)$ from $\mathcal{Z}_s$. Those requests are not completed yet.

Considering frequent license change requests, the immediate response to each request is not efficient. To efficiently satisfy license change (e.g., creation, transfer, and termination) requests in $\mathbb{S}_\sigma^{[n]}$, the proposed method uses the dSC-segment $\mathbb{S}_\sigma^{[n+1]}$ that started from $B^{[n_{i+1}]}$ immediately after the SC segment $\mathbb{S}_\sigma^{[n]}$ ended at $B^{[n_{i+1}]}$. When the dSC segment $\mathbb{S}_\sigma^{[n+1]}$ starts being formed, a miner in $\mathbb{S}_\sigma$ network reviews license change requests in $\mathbb{S}_\sigma^{[n]}$ for an SPDC s and makes a dSC block, say, $\hat{b}_\sigma^{[n+1,i]}$ which consists of one 'smart contract'

$$\text{sc} : \text{change } \mathcal{Z}_s \text{ to } \mathcal{Z}'_s \text{ with } q', \quad (29)$$

where $\mathcal{Z}_s$ and $\mathcal{Z}'_s$ are registration key sets for s before and after license change requests in $\mathbb{S}_\sigma^{[n]}$ are applied, respectively, and q' is the bitstream that satisfies (9) with $\mathcal{Z}'_s$, i.e.,

$$w = H_\lambda(H(s) \parallel \text{pk}(P) \parallel q' \parallel H(\mathcal{Z}'_s)).$$

The miner who finds q' first tries to connect the dSC block $\hat{b}_{\sigma}^{[n+1,i]}$ to the dSC segment $\hat{\mathbb{S}}_{\sigma}^{[n+1]}$ by following the block chaining rule of $\hat{\mathbb{S}}_{\sigma}$ SC.

The smart contract sc (29) is executed when the dSC segment $\hat{\mathbb{S}}_{\sigma}^{[n+1]}$ is connected to MC. To be specific, in the account of SPDC s in AC, the registration key set Z_s and the input value q are replaced with Z'_s and q' , respectively.

Figure 3 illustrates how the SPDC license change request is executed. In Step 1, SPDC license change requests are recorded in the SC segment $\mathbb{S}_{\sigma}^{[n]}$. In Step 2, SPDC license change requests recorded in $\mathbb{S}_{\sigma}^{[n]}$ are sorted according to SPDC and form smart contracts in $\hat{\mathbb{S}}_{\sigma}^{[n+1]}$. Finally, in Step 3, smart contracts in $\hat{\mathbb{S}}_{\sigma}^{[n+1]}$ are executed to change licenses sets in AC immediately after the dSC segment $\hat{\mathbb{S}}_{\sigma}^{[n+1]}$ to MC.

Figure 3. The process of changing an SPDC license in the HMBS.

3.3.4. Manager Program

The proposed method makes the MP (manager program) communicate with the IM (immobilizer) of SPDCs and use SPDC accounts in AC to determine who can play SPDCs. To explain this, let us consider the case when a user U uses the MP to play SPDC s . The user U does not need to pay the license fee for downloading s , since the downloading itself does not make s playable. At the moment when U plays s , The MP gives the ‘continue-to-play’ command to the IM inside the SPDC s if U can provide a correct response which can be made only by using the private key paired with one of the registration keys Z_s of s , and the ‘stop’ command to the IM of s , otherwise. Thus, to play s , U needs to pay the license fee by

requesting the transaction t (26) be recorded in SC. It leads to the inclusion of a registration key $\text{rk}(U)$ of U in $\mathcal{Z}_s$ of s . After this, U can play s by proving the ownership of the private key paired with $\text{rk}(U)$ to the MP, without revealing the private key itself.

4. Discussion

In this section, we shall discuss the results of thought experiments on possible attacks on the proposed method. In this thought experiment, we assume that the license $\mathcal{L}_s$ of SPDC s is assigned to a registration key $\text{rk}(U)$ of a user U .

Unauthorized SPDC license sharing: Suppose that U wants to share $\mathcal{L}_s$ with many anonymous Internet users. As mentioned in Section 3.3.2, the sharing of the SPDC license needs the sharing of the private key $\widehat{\text{rk}}(U)$ paired with $\text{rk}(U)$. In this case, any users who know $\widehat{\text{rk}}(U)$ can spend the registration fee assigned to $\text{rk}(U)$. Such action is easily detectable in the HMBS. It would lead to the termination of all SPDC licenses assigned to $\text{rk}(U)$. This scenario indicates that the attempt to perform Internet-wide license sharing would be strongly suppressed.

Attempt to use a terminated SPDC license: Suppose that the license $\mathcal{L}_s$ is terminated and U attempts to play s by using a local HMBS copy at which $\text{rk}(U)$ is still in the registration set $\mathcal{Z}_s$. The user U , however, cannot provide the up-to-dateness of $\mathcal{Z}_s$ to the manager program (MP), since the MP would refuse to play s by checking that $\mathcal{Z}_s$ is not from the latest HMBS.

Attempt to use the SPDC license without ever paying fee: Suppose that U attempts to use the license $\mathcal{L}_s$ without ever paying the license fee to the owner P of s . In other words, U never has requested a transaction like (26) to be recorded in one of SC segments. We note that U can play s if his registration key $\text{rk}(U)$ is in the registration key set $\mathcal{Z}_s$ of s . Therefore, if U could find the solution q of (9) with $\mathcal{Z}_s$ containing $\text{rk}(U)$, then this attempt might work at first glance. This approach is, however, worthless for following reasons:

- As a single miner, U would have extreme difficulty in finding the solution q of (9).
- Even if U finds the solution q of (9), he can only modify $\mathcal{Z}_s$ in his local copy of the HMBS. He cannot verify the validity and the up-to-dateness of the modified $\mathcal{Z}_s$ to the MP, and hence the MP refuses to play s .

It is also conceivable that U could make the smart contract sc request $\text{rk}(U)$ to be in $\mathcal{Z}_s$ in some dSC segments, say, $\mathbb{S}_\sigma^{[n+1]}$. In order for sc to be in the $\mathbb{S}_\sigma^{[n+1]}$, however, the immediately preceding SC segment $\mathbb{S}_\sigma^{[n]}$ must have a transaction which shows the purchase of the $\mathcal{L}_s$ by U from P . This contradicts our early assumption that U never has requested a transaction like (26) for the purchase of $\mathcal{L}_s$. Thus, it is impossible to use an SPDC license without ever paying the fee.

Attempt to use the SPDC license after reselling it: Suppose that U attempts to play s after reselling the SPDC license $\mathcal{L}_s$ to a user V . The transfer of the SPDC license $\mathcal{L}_s$ from U to V indicates that the registration key $\text{rk}(U)$ used in playing s is removed from $\mathcal{Z}_s$. Therefore, U cannot play s anymore.

Double HMBS coin spending: Suppose that U makes two transactions t_1 and t_2 to spend the same HMBS coin. Here we can exclude the case when t_1 or t_2 are in dSC segments with the reason that dSCs record smart contracts only, and the dSC block reward must be given in MC after smart contracts are executed in AC (we can include this kind of rule in smart contracts in dSCs from the beginning).

We begin with the case when both t_1 and t_2 are requested for approval in MC. In this case, it is obvious that only one of t_1 and t_2 can be approved in the sequential chain of MC blocks.

Next, we consider the case when $t_1 \in B^{[k]}$ in MC and $t_2 \in b_\sigma^{[n,j]}$ in $\mathbb{S}_\sigma^{[n]}$ segment that starts from $B^{[k']}$ and ends at $B^{[k']}$ in MC. If $k \leq k'$, then t_2 cannot be recorded in $b_\sigma^{[n,j]}$ from the beginning. If $k' < k$, then t_1 cannot be recorded in $B^{[k]}$ from the beginning. Finally, if $k' < k \leq k''$, then t_2 is recorded in $b_\sigma^{[n,j]}$, but regarded as invalid in MC. Therefore, only one of t_1 and t_2 can be used for the coin spending in the case when $t_1 \in B^{[k]}$ and $t_2 \in b_\sigma^{[n,j]}$.

We now consider the case when $t_1 \in b_{\sigma}^{[n,j]}$ in the $\mathbb{S}_{\sigma}^{[n]}$ segment that starts from $B^{[k']}$ and ends at $B^{[k'']}$ and $t_2 \in b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ in the $\mathbb{S}_{\tilde{\sigma}}^{[\tilde{n}]}$ segment that starts from $B^{[\tilde{k}]}$ and ends at $B^{[\tilde{k}']}$. We first consider the case when t_1 and t_2 are in different SCs, i.e., $\sigma \neq \tilde{\sigma}$. In this case, all k' , k'' , $\tilde{k}'$, and $\tilde{k}''$ are distinct, and hence it suffices to consider the following four cases. Each of them prevents double HMBS coin spending as follows:

- If $\tilde{k}'' < k'$, then t_1 cannot be recorded in $b_{\sigma}^{[n,j]}$ from the beginning, while t_2 is recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ and regarded as valid in MC.
- If $k'' < \tilde{k}'$, then t_1 is recorded in $b_{\sigma}^{[n,j]}$ and regarded as valid in MC, while t_2 cannot be recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ from the beginning.
- If $k' < \tilde{k}'' < k''$, then t_1 is recorded in $b_{\sigma}^{[n,j]}$, but regarded as invalid in MC, while t_2 is recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ and regarded as valid in MC.
- If $\tilde{k}' < k'' < \tilde{k}''$, then t_1 is recorded in $b_{\sigma}^{[n,j]}$ and regarded as valid in MC, while t_2 is recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$, but regarded as invalid in MC.

On the other hand, in the case when t_1 and t_2 are in the same kind of SC, i.e., $\sigma = \tilde{\sigma}$, it suffices to consider following three cases. Each of them prevents double HMBS coin spending as follows:

- If $\tilde{k}'' \leq k'$, then t_1 cannot be recorded in $b_{\sigma}^{[n,j]}$ from the beginning, while t_2 is recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ and regarded as valid in MC.
- If $k'' \leq \tilde{k}'$, then t_1 is recorded in $b_{\sigma}^{[n,j]}$ and regarded as valid in MC, while t_2 cannot be recorded in $b_{\tilde{\sigma}}^{[\tilde{n},\tilde{j}]}$ from the beginning.
- If $k' = \tilde{k}'$ and $k'' = \tilde{k}''$, i.e., t_1 and t_2 are in the same SC segment, then only one of t_1 and t_2 is recorded in that segment, and eventually regarded as valid in MC.

These arguments so far show that the proposed method allows only one of t_1 and t_2 to be regarded as valid in MC, and hence prevents double HMBS coin spending.

Double license reselling: Suppose that U attempts to resell $\mathcal{L}_s$ to both users V and W by requesting two transactions t_1 ($\mathcal{L}_s$ to V) and t_2 ($\mathcal{L}_s$ to W) to be recorded in SCs. Here we note that t_1 and t_2 must be recorded in the same SC, since t_1 and t_2 are transactions related to the same SPDC s . Thus, only one of t_1 and t_2 can be recorded in the sequential chain of blocks in single SC.

Malicious coin transferring to another user's registration key: Suppose U attempts to harm a user V by performing the following:

- U transfers a small amount of HMBS coin to a registration key $\text{rk}(V)$ of V , without V knowing, to make the HMBS coin assigned to $\text{rk}(V)$ not equal to the registration fee; and
- U requests the termination of $\text{rk}(V)$ with the evidence that the amount of coin assigned to $\text{rk}(V)$ is not equal to the registration fee.

At this point, $\text{rk}(V)$ is used at least twice; once in representing the coin of the registration fee for V and the other in representing the coin received from U . This could produce confusion; some miners might consider $\text{rk}(V)$ as invalid by judging that the latter one represents the registration fee.

To avoid this type of the wrong accusation, the proposed method lets the accused user V have an 'appeal chance'. When V is informed that their registration key $\text{rk}(V)$ is to be unregistered, V will spend all coins assigned to $\text{rk}(V)$ except the original registration-fee-representing coin and request the nullification of the transaction that showed $\text{rk}(V)$ as invalid. Even in the case when $\text{rk}(V)$ is already unregistered, V can still reverse the termination of all SPDC licenses of s assigned to $\text{rk}(V)$ by using the described method.

Notice that this approach does not work for the case when the private key $\widehat{\text{rk}}(V)$ paired with the registration key $\text{rk}(V)$ is disclosed; whenever V fills the HMBS coin assigned to $\text{rk}(V)$ to be equal to the registration fee, someone who knows $\widehat{\text{rk}}(V)$ would spend the

coin assigned to $\text{rk}(V)$, and hence the registration key $\text{rk}(V)$ would be requested for the termination, again.

Head start block mining: The block mining in the proposed method, whether it is in MC, AC, SC or dSC, needs the hash value of the immediately preceding block. Therefore, the mining of the next block is worthless, in the case when the hash value of the current block is not computed yet.

Head start computation of the input value: In the proposed method, miners in dSC network must make smart contracts in the dSC segment, say, $\mathbb{S}_\sigma^{[n+1]}$, by reviewing all transactions in the immediately preceding SC segment $\mathbb{S}_\sigma^{[n]}$. For instance, to make the smart contract (29), miners in dSC network must check whether the request of changing $\mathcal{Z}_s$ to $\mathcal{Z}'_s$ is correct, including the validity of hash values of SC blocks (in $\mathbb{S}_\sigma^{[n]}$) containing registration key change requesting transactions (recall the form (10) of the element of the registration key set). Therefore, miners cannot start the computation of the input value q' in $\mathbb{S}_\sigma^{[n+1]}$ at least before $\mathbb{S}_\sigma^{[n]}$ starts.

5. Conclusions

In this study, we propose a digital content service model based on a hierarchical multi-blockchain system (HMBS). The proposed method uses the HMBS as a tool to use the cryptocurrency for license fee collecting and manage content accounts defining license usage rules in the accountchain (AC). In addition, the proposed method uses several sidechains (SCs) and dual sidechains (dSCs) to efficiently record transactions and smart contracts of the HMBS by parallel processing. The transparent traceability and the tamper-proof nature of the proposed HMBS allow content owners to hire several promoters to increase content sale revenues, without worrying about the possibility of revenue hiding by some promoters.

The proposed method distributes digital content in the self decryptable encrypted format and makes the self decryption controlled only by a program called the indistinguishably embedded immobilizer (IM). The IM controls the decryption of content based on the HMBS received from the manager program (MP). The role of MP is to provide the latest HMBS to the IM. In this sense, the proposed method uses the HMBS as a secret key in decrypting digital content. By doing so, the proposed method can easily revoke the service to bad users.

The proposed method uses white-box cryptographic methods to allow SPDC to be stored safely in a hostile environment, such as users' hard drives. Furthermore, by making SPDC self decryptable, decryption keys can be disposable immediately after they are used in the creation of SPDC.

The HMBS controls the whole procedure of SPDC license distribution automatically. Thus, the SPDC owner does not need to respond to the SPDC license buying request; the smart contract of the HMBS gives the SPDC license to the buyer and collects the payment for the SPDC owner automatically.

The proposed digital content service model can give more benefit to users and content owners than current monopolistic platform services. The proposed method allows content owners to distribute their digital content and collect license fees without resorting to intermediaries which would demand a significant portion of revenues. The proposed method also provides various options to owners in promoting their content. For instance, their digital content can be sold in time-limited, device-limited, or license-transfer-number-limited versions, et cetera. Users will be also beneficial by not having intermediaries. Nowadays, intermediary platform companies set almost identical fixed prices for similar kinds of digital content, even though some content owners want to lower the price to promote their content more. In the proposed method, however, the price will be determined by the owner, and hence users will enjoy reduced prices.

The self decryptable encrypted format in the proposed method would generate extra overheads in distributing and storing digital content. The future research program includes

a study on reducing those overheads along with an empirical evaluation of the proposed smart propertization method for the protection of intellectual rights on digital content.

Author Contributions: Conceptualization, Writing—original draft, N.-Y.L., J.Y. and C.-S.K. All authors have read and agreed to the submitted version of the manuscript.

Funding: This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2019R1F1A1059357).

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

References

1. IFPI Global Music Report 2020—The Industry in 2019. Available online: <https://www.ifpi.org/ifpi-issues-annual-global-music-report> (accessed on 10 October 2020).
2. Grand View Research Report—Video Streaming Market Size, Share & Trends Analysis Report by Streaming Type, by Solution, by Platform, by Service, by Revenue Model, by Deployment Type, By User, by Region, and Segment Forecasts, 2020–2027. Available online: <https://www.grandviewresearch.com/industry-analysis/video-streaming-market> (accessed on 15 April 2021).
3. Oberholzer-Gee, F.; Strumpf, K. The effect of file sharing on Record Sales: An Empirical Analysis. *J. Political Econ.* **2007**, *115*, 1–42. [CrossRef]
4. Smith, M.D.; Telang, R. Competing with Free: The Impact of Movie Broadcasts on DVD Sales and Internet Piracy. *Manag. Inf. Syst. Q.* **2009**, *33*, 321–338. [CrossRef]
5. Andersen, B.; Frenz, M. Don't blame the P2P file-sharers: The impact of free music downloads on the purchase of music CDs in Canada. *J. Evol. Econ.* **2010**, *20*, 715–740. [CrossRef]
6. Hui, K.-L.; Png, I. Piracy and the Legitimate Demand for Recorded Music. *Econ. Anal. Policy* **2003**, *2*, 1–24. [CrossRef]
7. Peitz, M.; Waelbroeck, P. The Effect of Internet Piracy on Music Sales: Cross-Section Evidence. *Rev. Econ. Res. Copyr. Issues* **2004**, *1*, 71–79.
8. Zentner, A. File Sharing and International Sales of Copyrighted Music: An Empirical Analysis with a Panel of Countries. *Top. Econ. Anal. Policy* **2005**, *5*, 1–17.
9. Stevans, L.K.; Sessions, D.N. An Empirical Investigation Into the Effect of Music Downloading on the Consumer Expenditure of Recorded Music: A Time Series Approach. *J. Consum. Policy* **2005**, *28*, 311–324. [CrossRef]
10. Bounie, D.; Bourreau, M.; Waelbroeck, P. Piracy and the Demand for Films: Analysis of Piracy Behavior in French Universities. *Rev. Econ. Res. Copyr. Issues* **2006**, *3*, 15–27. [CrossRef]
11. Michel, N.J. The impact of digital file sharing on the music industry: An empirical analysis. *Top. Econ. Anal. Policy* **2006**, *6*, 1549. [CrossRef]
12. Rob, R.; Waldfogel, J. Piracy on the High C's: Music Downloading, Sales Displacement, and Social Welfare in a Sample of College Students. *J. Law Econ.* **2006**, *49*, 29–62. [CrossRef]
13. Adermon, A.; Liang, C.-Y. Piracy and music sales: The effects of an anti-piracy law. *J. Econ. Behav. Organ.* **2014**, *105*, 90–106. [CrossRef]
14. Sharpe, N.F.; Arewa, O.B. Is Apple Playing Fair? Navigating the iPod FairPlay DRM Controversy. *Northwest. J. Technol. Intellect. Prop.* **2007**, *5*, 331–349.
15. Ducci, F. *Natural Monopolies in Digital Platform Markets*; Cambridge University Press: Cambridge, UK, 2020.
16. Ujo Music. Available online: <https://ujomusic.com> (accessed on 15 April 2021).
17. Ethereum. Available online: <https://ethereum.org> (accessed on 15 April 2021).
18. Resonate Music. Available online: <https://resonate.is> (accessed on 15 April 2021).
19. SingularDTV. Available online: <https://singulardtv.com> (accessed on 15 April 2021).
20. Mishra, D. An accountable privacy architecture for digital rights management system. In Proceedings of the Sixth International Conference on Computer and Communication Technology, Allahabad, India, 25–27 September 2015.
21. Zhang, Z.; Zhao, L. A Design of Digital Rights Management Mechanism Based on Blockchain Technology. In *Blockchain—ICBC 2018*; Chen, S., Wang, H., Zhang, L.J., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 10974.
22. Bhowmik, D.; Feng, T. The multimedia blockchain: A distributed and tamper-proof media transaction framework. In Proceedings of the 22nd International Conference on Digital Signal Processing (DSP 2017), London, UK, 23–25 August 2017.
23. Ma, Z.; Huang, W.; Gao, H. A new blockchain-based trusted DRM scheme for built-in content protection. *EURASIP J. Image Video Process.* **2018**, *2018*, 91. [CrossRef]
24. 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.
25. Opuni-Boachie, K.; Agyekum, O.; Xia, Q.; Liu, Y.; Pu, H.; Cobblah, C.N.A.; Kusi, G.A.; Yang, H.; Gao, J. Digital Media Copyright and Content Protection Using IPFS and Blockchain. In *Image and Graphics, ICIG 2019*; Zhao, Y., Barnes, N., Chen, B., Westermann, R., Kong, X., Lin, C., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11903.

26. Gaber, T.; Ahmed, A.; Mostafa, A. Privdrm: A privacy-preserving secure digital right management system. In Proceedings of the Evaluation and Assessment in Software Engineering, Trondheim, Norway, 15–17 April 2020.
27. Keoh, S.L. Marlin: Toward seamless content sharing and rights management. *IEEE Commun. Mag.* **2011**, *49*, 174–180. [CrossRef]
28. Kishigami, J.; Fujimura, S.; Watanabe, H.; Nakadaira, A.; Akutsu, A. The blockchain-based digital content distribution system. In Proceedings of the 2015 IEEE Fifth International Conference on Big Data and Cloud Computing, Dalian, China, 26–28 August 2015.
29. Vishwa, A.; Hussain, F.K. A blockchain based approach for multimedia privacy protection and provenance. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018.
30. Guo, J.; Li, C.; Zhang, G.; Sun, Y.; Bie, R. Blockchain-enabled digital rights management for multimedia resources of online education. *Multimed. Tools Appl.* **2019**, *79*, 9735–9755. [CrossRef]
31. Ghaffar, A.; Azeem, M.; Abubaker, Z.; Gurmani, M.U.; Sultana, T.; Shehzad, F.; Javaid, N. Smart Contracts for Research Lab Sharing Scholars Data Rights Management over the Ethereum Blockchain Network. In *Advances on P2P, Parallel, Grid, Cloud and Internet Computing. 3PGCIC 2019*; Barolli, L., Hellinckx, P., Natwichei, J., Eds.; Lecture Notes in Networks and Systems; Springer: Cham, Switzerland, 2019.
32. Garba, A.; Dwivedi, A.D.; Kamal, M.; Srivastava, G.; Tariq, M.; Hasan, M.A.; Chen, Z. A digital rights management system based on a scalable blockchain. *Peer-to-Peer Netw. Appl.* **2020**. [CrossRef]
33. McConaghy, M.; McMullen, G.; Parry, G.; McConaghy, T.; Holtzman, D. Visibility and digital art: Blockchain as an ownership layer on the Internet. *Strateg. Chang.* **2017**, *26*, 461–470. [CrossRef]
34. Wang, Z.; Yang, L.; Wang, Q.; Liu, D.; Xu, Z.; Liu, S. ArtChain: Blockchain-Enabled Platform for Art Marketplace. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019.
35. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for digital rights management. *Future Gener. Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
36. 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]
37. Suciu, G.; Nădrag, C.; Istrate, C.; Vulpe, A.; Ditu, M.; Subea, O. Comparative Analysis of Distributed Ledger Technologies. In Proceedings of the 2018 Global Wireless Summit (GWS), Chiang Rai, Thailand, 25–28 November 2018.
38. Hei, Y.; Liu, J.; Feng, H.; Li, D.; Liu, Y.; Wu, Q. Making MA-ABE fully accountable: A blockchain-based approach for secure digital right management. *Comput. Netw.* **2021**, *191*, 108029.
39. Federal Information Processing Standards, U.S. National Institute of Standards and Technology. Announcing the Advanced Encryption Standard (AES). Available online: <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf> (accessed on 15 April 2021).
40. Chow, S.; Eisen, P.A.; Johnson, H.; van Oorschot, P.C. A white-box DES implementation for DRM applications. In *Digital Rights Management, Proceedings of the ACM Workshop on Security and Privacy in Digital Rights Management (DRM 2002)*, Washington, DC, USA, 18 November 2002; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2002.
41. Billet, O.; Gilbert, H.; Ech-Chatbi, C. Cryptanalysis of a White Box AES Implementation. In *Selected Areas in Cryptography, Proceedings of the 11th International Workshop on Selected Areas in Cryptography (SAC 2004)*, Waterloo, ON, Canada, 9–10 August 2004; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2004.
42. DeMulder, Y.; Roelse, P.; Preneel, B. Cryptanalysis of the Xiao-Lai White-Box AES Implementation. In *Selected Areas in Cryptography, Proceedings of the SAC 2012, Trento, Italy, 26–30 March 2012*; Knudsen, L.R., Wu, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2013.
43. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 15 January 2020).
44. NIST. Descriptions of SHA-256, SHA-384, and SHA-512. Available online: <https://web.archive.org/web/20130526224224/http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf> (accessed on 15 April 2021).
45. Narayanan, A.; Bonneau, J.; Felten, E.; Miller, A.; Goldfeder, S. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*; Princeton University Press: Princeton, NJ, USA, 2016.
46. Lee, N.-Y.; Yang, J.; Onik, M.M.H.; Kim, C.-S. Modifiable Public Blockchains Using Truncated Hashing and Sidechains. *IEEE Access* **2019**, *7*, 173571–173582. [CrossRef]