IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2023, accepted 18 April 2023, date of publication 26 April 2023, date of current version 5 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3270627

METHODS

Decentralized Global Copyright System Based on Consortium Blockchain With Proof of Authority

MD. MAINUL ISLAM[1], (Graduate Student Member, IEEE), AND HOH PETER IN[2], (Member, IEEE)
[1]Department of Computer Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia
[2]Department of Computer Science and Engineering, Korea University, Seoul 02841, South Korea

Corresponding author: Hoh Peter In (hoh_in@korea.ac.kr)

This work was supported by National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. NRF-2021R1A2C2012476).

ABSTRACT Conventional copyright systems are governed nationally, and there is no global ledger for storing copyright data. Due to the lack of a global copyright monitoring system, it is difficult to provide cross-border copyright protection. In this paper, we propose a novel decentralized copyright system based on a consortium blockchain, which ensures cross-border copyright protection of individuals' digital content and solves existing challenges in international copyright management. The proposed system enables a synchronized platform to register and trade copyright globally without using a global cloud. Individual countries receive membership from a copyright federation and participate in block creation by executing the energy-efficient proof of authority consensus algorithm. These countries are regarded as the authorities of the platform. They validate transactions conducted by users and store them in the blockchain. Anyone, either registered or unregistered, can investigate a copyrighted work, but only registered users can make transactions. A token-based payment method is also proposed for paying copyright charges (i.e., transaction fees) to authorities through the federation. A prototype of the system was implemented, and its performance was evaluated. This paper provides direction and guidance towards international copyright management.

INDEX TERMS Decentralized copyright, international copyright, cross-border copyright, copyright protection, consortium blockchain, proof of authority.

I. INTRODUCTION

With the advancement of the internet and social media, the reproduction of digital content is increasing rapidly worldwide. At the same time, the protection of copyrighted content has become a crucial requirement. Content creators must be assured that they can protect their content from piracy when making it available online. Copyright infringement causes economic losses to producers, developers, and policymakers. It discourages creative works and hinders the development of science and innovation. There is no such term as international copyright laws; instead, there are some international treaties on copyright protection [2]. The first acceptable treaty on cross-border copyright protection was the treaty proposed by the Berne Convention of 1886 to set an international agreement for protecting literary and artistic works beyond the borders of individual countries [3]. Today, more than 179 countries worldwide have signed this treaty. The basic principle of the treaty is regarding legitimate rights for the protection of literary and artistic creations that originated in a contracting state. According to the treaty, if a copyright is infringed by someone in a foreign territory and the convention is available, the copyright owner is treated as a citizen of the foreign territory and must be prepared to proceed with the national copyright laws of that territory, which is termed as "national treatment" [2]. The Bern Convention has not defined a specific court for conducting national treatments; instead, the treatments are practiced in the local jurisdictions of the foreign territories where the copyright infringements occurred. The minimum protection period is 50 years after the owner's death for an artistic work and 25 years for a photograph. The minimum

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

protection standards regarding copyrighted works are the rights to reproduce or make copies, distribute, translate, make adaptations, recite or perform publicly, display, and broadcast.

A. MOTIVATION

There also exist several global organizations, such as the World Intellectual Property Organization (WIPO), International Federation of Reproduction Rights Organizations (IFRRO), European Commission - Copyright International, Copyright Clearance Center (CCC), and RightsDirect, who aim to protect copyright from a global perspective. However, they are not still able to achieve complete harmonization is not in cross-border copyright protection [1], [2]. The absence of a central cloud that can store copyrighted content worldwide and the lack of transparency raise challenges for these organizations in international copyright management [3], [4], [5], [6]. Copyright information is dispersed across various centralized databases, which are not incentivized to share. It is impossible to unite all centralized servers, which store billions of copyright data, into a synchronized platform. Due to the lack of a global copyright monitoring system through which people can investigate whether a work is copyrighted by searching copyright catalogs and records, copyrighted works are often unconsciously downloaded from Google or social media and freely shared with others, which may result in huge revenue losses to copyright holders. Therefore, a global platform for copyright management is urgently needed in this era of globalization. To meet the requirements for international copyright management, blockchains can play an important role by providing a distributed ledger, unlike traditional centralized databases, in which worldwide copyright data can be recorded in a transparent and tamper-proof manner [6]. Copyrighted content throughout the world can easily be recognized and publicly be verified.

A blockchain is a tamper-proof shared ledger that records timestamped transactions on chronological blocks identified by unique hashes. These blocks are cryptographically connected to each other across a distributed network, and anyone in the network can verify their correctness. It can eliminate the dependency on trusted third parties for resource management and establish mutual trust among unknown entities. Resources can be tangible (e.g. money, lands, cars, houses, and energy) or intangible (e.g. copyrights, intellectual properties, educational certificates, birth certificates, medical reports, and other digital documents) [7]. Literally, a blockchain network is able to track anything that has a numerical value and can mitigate the overall security risks and monitoring costs involved [8]. It reduces manpower by using a consensus algorithm that provides a provably secure method for machines to collaborate with each other [9], [10]. These advantages meet the necessary requirements, but several issues must be addressed such as how to register digital content for copyright, where to store the copyrighted content, and what type of blockchain should be used.

B. PROBLEM DEFINITION

While reviewing a copyright application in a national copyright system, there is no ability to search or compare the claimed work with other works to justify whether similar works were previously registered in the system's database or elsewhere. A copyright office does not knowingly register a duplicate or ambiguous claim that is available in the public domain. If a registered claim is discovered to be an exact duplicate of another claim, the copyright office refuses the registration [11]. By contrast, a blockchain can immediately detect and reject a duplicate claim at the time of registration if all copyrighted content is available on-chain. Each transaction in blockchains has a collision-resistant, unique Merkle hash. In the case of copyright registration, if the hash value of a file transacted by user A matches the hash value of a file already registered by user B, miners (validators) can easily identify that A's file is a duplicate copy of B's file.

The problem with blockchain is that the storage of massive files on-chain is impossible due to high memory consumption and network overhead. For example, the size of the Bitcoin blockchain doubles each year even though it contains only metadata about transactions and limits the block size to 1 MB [6]. Therefore, it is unwise to store copyrighted content in a blockchain. Instead, the content can be stored in a separate server, and their locations can be included as metadata in transactions. However, location addressing poses another problem when changing the URL of a piece of content or when the server is unavailable. To solve this issue, the interplanetary file system (IPFS), a peer-to-peer (P2P) file sharing protocol, has introduced content addressing in which each content is identified by a unique address called a content identifier (CID) [12]. Each CID is derived from each piece of content using cryptographic hash functions called multi-hashes. Content is permanently stored in a distributed network and accessed by its CID (i.e., https://ipfs.io/ipfs/CID). The problem with IPFS is that multi-hashes do not provide proof of ownership because mathematical operations cannot be performed on hash values. Therefore, the owner of a digital content uploaded to IPFS network cannot be verified unless the network implements an additional centralized layer for access control, compromising decentralization.

C. OUR SOLUTION

This study aims to solve the data synchronization and copyright verification issues of international copyright management.

- To address the data synchronization issue faced by conventional centralized copyright offices, individual countries under a federation maintain a consortium blockchain. To support copyright verifiers free of charge, the federation keeps a copy of the entire blockchain containing worldwide copyright information and displays them on its official website.
- To overcome the copyright verification issue faced by IPFS, elliptic curve cryptography (ECC) is used

in which a CID represents a public key or point on an elliptic curve [13]. The content ownership can be publicly verified using the owners' digital signatures. If the signatures are verified with the signers' public keys, their ownership claims are true as they know the corresponding private keys of the public keys. Thus, worldwide copyrighted content can be easily identified and verified. Before sharing any media file, one can check whether it has copyright or not to avoid copyright infringements.

Our proposed solution includes copyright registration, verification, and trading facilities. The steps for copyright registration of a media file are as follows:

1) Owner stores the file in any local device.
2) Owner computes its CID.
3) Owner buys a token from the copyright federation.
4) Owner generates a transaction including the CID and ownership details.
5) Owner signs the transaction and broadcasts it to the network.
6) Authorities validate the transaction and store it in the blockchain.

The steps for copyright trading of a media file are as follows:

1) Buyer purchases the copyright or license from the current owner.
2) Buyer generates a transaction including proofs of purchase.
3) Buyer signs the transaction and broadcasts it to the network.
4) Authorities validate the transaction and store it in the blockchain.

The steps for copyright investigation of a media file are as follows:

1) Investigator uploads the file to the federation's website.
2) The website computes the file's CID.
3) The website searches the blockchain for the transaction that contains the CID.
4) The website shows its copyright information (if available).

D. RELATED WORK

Several papers regarding copyright protection exist in the literature, and they are primarily focused on centralized copyright management. In [14], [15], and [16], the authors proposed watermarking-based image copyright protection without considering a blockchain and various media files from a global perspective. The study [17] revealed that there is not yet a comprehensive and organized taxonomy devoted to blockchain-based copyright protection solutions. Furthermore, there are very few blockchain-based content protection systems that have been developed successfully. This research gap raises open challenges to digital right management (DRM). In order to promote the deployment of blockchain-based copyright protection system, the authors presented a taxonomy that integrates the technological components of blockchain and digital content protection applications. Table 1 presents a qualitative comparison of blockchain-based copyright management systems. Ma et al. [18] developed an Ethereum-based DRM system called DRMchain, which ensures the appropriate usage of digital content by authorized users and provides flexible external storage of digital content through IPFS. DRMchain makes use of two separate blockchain application interfaces (BAI): the plain BAI, which stores the original content along with its cipher summary, and the cipher BAI, which houses the copyright protection services, including content watermarking, encryption, license, and violation tracing.Â DRMchain offers effective and secure authentication, privacy protection, conditional traceability based on multiple signatures, and reliable content protection. However, it cannot prevent the offline distribution of the revealed copies. In addition, the system lacks a reward mechanism and functionality for diverse copyright management such as assignment.

The Blockchain as a Service model was proposed in [19] to create a DRM platform that offers content creators, customers, and service providers high-level credit and security. The DRM platform provides copyright data storage in the blockchain to avoid copyright infringement or abuse. Users can make payments for content consumption using digital currencies based on blockchain technology. As a payment method on the platform, a multi-signature-based cryptocurrency digital rights coin is proposed. Secure connections and data transfer are made possible through the use of dynamic key agreement and session data encryption. However, this technique makes extensive use of modulo operations, considerably reducing the efficiency of creating a temporary shared key. In addition, the alliance chain foundation of the platform employs a central authority for the prevention of direct dealings between content owners and consumers. Wang et al. [20] proposed an image copyright protection framework with the combination of zero-watermarking algorithm, Ethereum blockchain, and IPFS. Images are uploaded to IPFS in the form of ciphertext (encrypted plaintext) so that no IPFS node can recover the original information. Only authorized individuals have access to the unique password, which is stored and distributed using smart contracts. However, no details about the consensus have been provided.

To protect music copyright effectively, Zhao and O'Mahony [21] introduced a prototype called BMCProtector that is built on Ethereum. BMCProtector encrypts music files using the AES algorithm. To trace ownership of the files off-chain, it adopts an off-chain access control mechanism and vector quantization (a method of watermarking). Only users who paid for the files can get the decrypting keys. The deployed smart contract is in charge of automatically sending royalties to the copyright owners' wallet addresses and sharing the music owners' copyright details. However, the prototype supports copyright management of audio files only. Bhowmik and Feng [22] proposed a multimedia blockchain system to provide the distributed image security and integrity. The system operates on a self-embedding watermarking algorithm that uses compressive sensing to

TABLE 1. **Comparison of Blockchain-Based Copyright Management Systems.**

References	Types of content	Types of blockchain	Content storage	Methods of trading	Consensus algorithms	Content protection techniques
[18]	Video	Ethereum	IPFS	Ether	PoW/PoS/ PBFT	Watermarking
[19]	Video	Ethereum, hyperledger fabric	Centralized server	Ethereum token	PoW/PoS/PBFT	Encryption
[20]	Video	Hyperledger fabric	On-chain	–	–	Watermarking
[21]	Audio	Ethereum	IPFS	Ether	PoW	Watermarking
[22]	Image	Ethereum	Centralized server	Ether	–	Watermarking
[23]	Image	Ethereum	IPFS	Ether	–	Encryption + watermarking
[24]	Image	Ethereum	IPFS	Ether	–	Encryption + watermarking
[25]	eBook	Self-developed	Centralized server	Customized coin	PoW	ECC-based encryption
[26]	Any media files	Ethereum	On-chain	Ether	PBFT	Watermarking
[27]	Any digital files	Hyperledger fabric	IPFS	–	PBFT	–
[28]	Any digital files	Hyperledger fabric	–	–	–	Digital fingerprints
[29]	Any digital files	Self-developed	Centralized server	–	WBFT	Encryption
This work	Any digital files	Self-developed	Flexible	Federated token	PoA	ECC-based verifiable CIDs

detect any tampering and restore the original content. The watermark on an image isÂ composed of two hashes: an image hash that is used to identify the tampered regions of the image, and a cryptographic hash that is used to retrieve the metadata (transaction logs) of the image from the blockchain. The image is distributed when the validating nodes have given their approval to the corresponding transaction, and it is then kept in a media server. Although it is significant to store image verification data in the blockchain, the availability of image management is impacted when the server is down.

An Ethereum-based digital copyright management system was proposed by Peng et al. [23], allowing direct business transactions between content providers and consumers without involving a central authority. It uses the ElGamal cryptosystem, IPFS, and smart contracts along with digital watermarking. The usage of ElGamal encryption to encrypt the entire multimedia content causes the scheme to have a significant overhead (memory and CPU time). Thus, the system trades memory and speed for traceability and security. Wu et al. [24] a blockchain-based novel zero-watermarking system for video copyright protection. On-chain data storage and on-chain data traceability are the primary purposes of the blockchain used in this system. However, this is an unwise approach because it increases the block size and node response time. A single transaction containing 18 min short video in the system can be as large as 71 MB while most real-world blockchains keep their transaction size in KB range to avoid network overhead. Although the authors experimentally showed a satisfactory performance with 8 GB on-chain data and 45 users, the performance of the system may severely degrade when billions of video will be stored on-chain by a growing number of users.

Chi et al. [25] introduced a trustworthy and secure real-time eBook marketing system that enables users to independently publish creative content and receive payments from readers. Blockchain is used to provide P2P payments and protect eBook content that has been paid for. A book repository houses both the published, encrypted eBook content and the eBook key. A digital watermark management system called Y-DWMS was presented in [26] as a way to guard against the infringement of digital rights. The system is built on a public smart contract platform. The smart contract is intended to carry out tasks like authenticating the informer's report, tracking down infringers, punishing infringers, recovering losses sustained by the copyright owners, and rewarding informers. It also performs the verification of watermarks on the disclosed copies. However, Y-DWMS has some security flaws, including account security and privacy, as it is still in its early stages of development.

Ouyang et al. [27] proposed a copyright management platform with the integration of IPFS and hyperledger fabric. To registrar a copyright, the content owner uploads the literary file to IPFS and collects the CID. Then, the copyright type, number, price, author name, author ID, CID, text summary, and additional relevant information are posted to the blockchain network. A vulnerability of this approach is that a dishonest party can register the content in their name before the content owner once the content is uploaded to IPFS. Liu et al. [28] proposed digital copyright protection based on hyperledger fabric. In their system, registration is also prerequisite for copyright protection. Users must upload media files to the network during registration, whereas content creators can keep their content confidential until the content has been registered in our system. Uploading media files prior to the registration raises concern if the network comprises any malicious nodes. Another concern about their approach is that the hash values of digital files are stored in the blockchain instead of storing any public keys that could prove ownership of the files in terms of digital signatures.

Heo et al. [29] introduced an efficient and secure approach for digital content trading that uses a combination of off-chain and on-chain network components. User authentication and copyright registration are processed on a centralized platform, which employs a server to store encrypted content. To address the storage capacity limitation, copyright consumers create some secret blocks that contain encrypted content and transaction details. The headers of the secret blocks are stored in a public blockchain as transactions. Both authenticated and unauthenticated users maintain the public blockchain network by executing a weighted Byzantine fault tolerance (WBFT) consensus algorithm, where authenticated users have higher weights and an increased chance to be elected as validators. A major drawback of their system is that the consensus time increases as the number of users increases. Thus, their approach is only applicable for a platform that has a limited number of users.

There are several limitations and drawbacks of these works. Firstly, watermarking protocols can only protect the copyright of photographic, textual, and visual works. Therefore, their use cases are limited from a global copyright perspective, which should cover all types of digital content. In addition, most watermarks fail to prove ownership and are vulnerable in this era of advanced editing technology [30], [31]. Secondly, storing worldwide media files in a centralized server requires a huge storage capacity, which is challenging and expensive. On the other hand, the systems that use IPFS for content storage cannot provide proof of ownership. Thirdly, the proof of work (PoW) consensus algorithm requires high energy consumption, proof of stake (PoS) favors wealthy validators, and practical BFT (PBFT) has high computational complexity. They also suffer from low transaction throughput, whereas proof of authority (PoA) provides energy-efficient, high-speed consensus. Fourthly, most of the papers do not provide the blockchain implementation details (e.g., transaction validation and consensus processes) and sufficient experimental results (e.g., transaction and block sizes, transaction verification time, and throughput).

E. OUR CONTRIBUTIONS

The key contributions of this research study can be summarized as follows:

- We design a global copyright monitoring system based on blockchain technology in which worldwide digital content can be registered with a low registration fee and verified free of charge. Copyright data are recorded on a distributed ledger so that people can easily inquire about digital content to ascertain whether it is copyrighted or not. Thus, they can become conscious of copyrighted works and will not unknowingly violate copyright laws.
- Unlike IPFS, ECC is used for content addressing so that the real owners can prove their content creations in terms of digital signatures.
- Copyright owners are free to keep their content anywhere, even on their local storage devices. The verifiable public keys, which are linked to the content, are recorded on the blockchain as metadata. This method is safe and does not require a centralized server to store copyrighted content globally.
- The proposed system supports off-chain ownership transfer with fiat currencies in which a seller delivers the private key of an auxiliary public key regarding a copyrighted work to a buyer as proof of the ownership transfer. The buyer can record the purchase on the blockchain by making an on-chain transaction with a digital signature.
- The proposed blockchain network can also be applied when deploying a decentralized cross-border or interbank payment system.

The remainder of this paper is organized as follows: The background related to this research is provided in Section II. The proposed system and algorithms are presented in Section III. The system's security is analyzed in Section IV. The implementation results and performance analyses are presented in Section V. The limitations of the system and potential areas for future research are discussed in Section VI. Finally, this research study is concluded in Section VII.

II. BACKGROUND

A. COPYRIGHT

Copyright is an IP that protects its owner's original work of authorship as soon as the work is fixed in a tangible medium of expression [32]. The work can be any type of digital content, including articles, books, newsletters, emails, images, sound recordings, videos, architectural designs, software, and computer programs. The term "original work of authorship" refers to a work that is created by an author independently with a minimal degree of creativity, excluding its title, name, symbols, short phrases, and colors. Copyright does not protect the concept, idea, principle, and method of content creation; rather, it protects the way in which these things are expressed. Minor variations in datasets may be sufficient to abolish copyright protection. If a copyright owner transfers all of the rights associated with their copyrighted work to a person without any condition, it is termed as an assignment. When the owner transfers some specific rights to someone, it is called a license.

B. CONSORTIUM BLOCKCHAIN

A consortium blockchain is a permissioned blockchain that combines the features of both public and private blockchains [33]. It is governed by a group of miners, where each miner refers to an authority or organization. In a consortium blockchain, a 51% or Sybil attack [34] is impractical since validators come from multiple authorized organizations and must be identified. Consortium blockchains provide many advantages over public blockchains, including low-latency transactions, lightweight consensus, low power consumption, proper monitoring, and no illicit operation. PBFT, Istanbul BFT (IBFT), delegated BFT (DBFT), and PoA are several consensus algorithms compatible with consortium blockchains. PoA outperforms the BFT variants in terms of speed [35]. In contrast to PBFT and IBFT, which can tolerate up to f faulty nodes among $3f + 1$ consensus nodes, PoA can tolerate a maximum of f faulty nodes in a network consisting of $2f + 1$ consensus nodes. Thus, PoA has greater fault tolerance than PBFT and IBFT. The computational complexity of the BFT and PoA-based algorithms is $\mathcal{O}(n^2)$ and $\mathcal{O}(n)$, respectively. Because of their greater computational complexity, BFT-based algorithms are slower than PoA. The consensus processes of DBFT and PoA are very similar except the leader selection methods. DBFT arranges an election for reputation-based leader selection [36], whereas PoA sets a round-robin schedule for time-based leader selection in which block proposers are

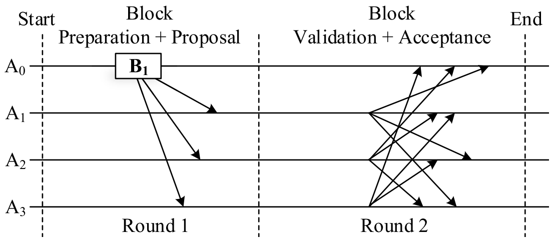

FIGURE 1. Message patterns of the PoA (Aura) consensus algorithm [37].

selected in turns, and all consensus nodes get equal rights and revenue.

C. PROOF OF AUTHORITY

PoA is a lightweight consensus algorithm that provides an efficient solution for permissioned blockchains [33]. Unlike PoW, it does not include the longest chain or confirmation rule. New blocks are directly added to the blockchain with the unanimous approval of a limited number of trusted validators. There is no competition for cryptographic puzzle solving. Therefore, low computing power is required to execute this algorithm. As the name suggests, PoA depends on a set of N trusted nodes called authorities, each of which is identified by a unique ID or public key [37]. To ensure transparency in the network, a majority (at least $N/2+1$) of authorities must be honest, and as such, all authorities closely monitor each other's actions. Consensus in PoA is reached through a round-robin schedule in which the responsibility of block creation is assigned to each authority in turn. PoA has two different implementations, called Aura and Clique [38]. Although both the implementation have their own advantages and disadvantages, we adopt the Aura implementation for our system to avoid forks and achieve better synchronization with higher security. Fig. 1 illustrates the message patterns of Aura. The first round covers block proposal by the current leader, and the second round is required to obtain block acceptance from the non-leader authorities. The block is added to the blockchain after receiving acceptance from the majority of authorities.

Evaluation of Aura Based on The CAP Theorem:

Consistency: A blockchain ensures ledger consistency by avoiding *forks*. If *forks* occur, synchronizing the ledgers of individual consensus nodes becomes difficult. As only a single node can propose block at a step, no *fork* appears in Aura.

Availability: If users' broadcast transactions are continuously processed, a blockchain network is then available. Because of allowing only the leader node to propose block at each step, transaction throughout of the Aura network decreases if the leader is offline for some moments. Thus, Aura trades availability for consistency.

Partition tolerance: When a blockchain network splits up, disjoint groupings of consensus nodes are formed, where each group holds a different blockchain. It requires a majority of Byzantine nodes for identifying the right blockchain. As a result, Aura can tolerate up to $N/2-1$ faulty nodes for even N or $(N-1)/2$ faulty nodes for odd N.

D. ELLIPTIC CURVE CRYPTOGRAPHY

ECC is used to make a transaction chain of copyrighted content. The elliptic curve digital signature algorithm (ECDSA) [39] is employed to sign and verify transactions. An elliptic curve is expressed by the following equation [13], [41]:

$$E_{a,b}: y^2 = x^3 + ax + b \tag{1}$$

where $x, y, a, b \in \mathbb{F}_p$, p is a curve-specific prime number, and

$$4a^3 + 27b^2 \neq 0 \tag{2}$$

Secp256k1 is a version of $E_{a,b}$ widely used in blockchains for which $a = 0$, $b = 7$, and $p = 2^{256} - 2^{32} - 977$ [40], [41]. Therefore, this curve can be provided as follows:

$$E: y^2 = x^3 + 7 \tag{3}$$

The group of points $E(\mathbb{F}_p)$ is a finite, cyclic, additive group of order q and base point P.

A public key Q is an arbitrary point on E, which is generated through the elliptic curve point multiplication (ECPM) operation [41]. The underlying theory of ECPM is to multiply the generator point $P \in E(\mathbb{F}_p)$ with a private key or integer $k \in \mathbb{F}_p$, i.e., $Q = kP \in E(\mathbb{F}_p)$. The elliptic curve group operations (i.e., point addition and point doubling) are part of ECPM [42].

Let $P(x_1, y_1) \in E(\mathbb{F}_p)$ and $Q(x_2, y_2) \in E(\mathbb{F}_p)$ be two points in affine coordinates. The point addition $P + Q$ generates another point $R(x_3, y_3) \in E(\mathbb{F}_p)$ as follows [13]:

$$\begin{aligned} R(x_3, y_3) &= \{P(x_1, y_1) + Q(x_2, y_2)\} \in E(\mathbb{F}_p) \\ x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \tag{4}$$

where $\lambda = (y_2 - y_1)(x_2 - x_1)^{-1}$ and $P \neq Q$.

The point doubling of $P(x_1, y_1)$ on E is performed as follows [13]:

$$\begin{aligned} R(x_3, y_3) &= 2P(x_1, y_1) \in E(\mathbb{F}_p) \\ x_3 &= \lambda^2 - 2x_1 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \tag{5}$$

where $\lambda = (3x_1^2)(2y_1)^{-1}$and $P = Q$.

In affine coordinates, adding and doubling points requires multiple division operations over $\mathbb{F}_p$ [43] that are considerably time consuming because the computation time of a single modular division is the same as 80 modular multiplications [13]. To reduce ECPM time by avoiding modular divisions, point addition and doubling are carried out in Jacobian coordinates by expressing the points P and Q as $(X_1 = x_1, Y_1 = y_1, Z_1 = 1)$ and $(X_2 = x_2, Y_2 = y_2, Z_2 = 1)$, respectively.

Algorithm 1 Binary Method for ECPM [13], [41]

```
Input: Base point $P$, private key $k$
Output: Public key $Q$
$Q = P$
for $i$ in range($len(k) - 2, 0$) do
    $Q = 2Q$
    if $k[i] = 1$ then
        $Q = P + Q$
    end if
end for
return $Q$
```

The formula for projective point addition is given by the equation [13]:

$$\begin{aligned} R(X_3, Y_3, Z_3) &= P(X_1, Y_1, Z_1) + Q(X_2, Y_2, Z_2) \in E(\mathbb{F}_p) \\ X_3 &= \alpha^2 - \beta^3 - 2X_1Z_2^2\beta^2 \\ Y_3 &= \alpha(X_1Z_2^2\beta^2 - X_3) - Y_1Z_2^3\beta^3 \\ Z_3 &= Z_1Z_2\beta \end{aligned} \tag{6}$$

where $\alpha = Y_2Z_1^3 - Y_1Z_2^3$ and $\beta = X_2Z_1^2 - X_1Z_2^2$.

The formula for projective point doubling is given by the equation [13]:

$$\begin{aligned} R(X_3, Y_3, Z_3) &= 2P(X_1, Y_1, Z_1) \in E(\mathbb{F}_p) \\ X_3 &= 9X_1^4 - 8X_1Y_1^2 \\ Y_3 &= 3X_1^2(4X_1Y_1^2 - X_3) - 8Y_1^4 \\ Z_3 &= 2Y_1Z_1 \end{aligned} \tag{7}$$

ECPM can be carried out by $k - 1$ point additions of P as follows [41]:

$$Q = P + \underbrace{P + \ldots\ldots + P}_{k-1 \ times} \tag{8}$$

If k represents a power of 2, Q can be computed by log_2k times doubling of P as follows [41]:

$$Q = \underbrace{\ldots 2(2(2(P)))}_{log_2k \ \ times} \tag{9}$$

Algorithm 1 presents the binary method for public key Q generation from private key k. The binary bit pattern of k is used to calculate Q through the sequential operations of point addition and point doubling. Point doubling is operated in every iteration. However, point addition is carried out when k's current bit is 1.

Definition 1: If an elliptic curve E is defined over a prime field $\mathbb{F}_p$, base point $P \in E(\mathbb{F}_p)$ of order q, and arbitrary point $Q \in \langle P \rangle$. Determine the integer $k \in [0, q-1]$ so that $Q = kP$ holds. Here, k is called the discrete logarithm of Q to P, which is denoted as $k = log_PQ$. It is difficult to find k when it is sufficiently large. This difficulty is termed as the elliptic curve discrete logarithm problem (ECDLP) [13], [44].

III. PROPOSED COPYRIGHT SYSTEM

A. METHODOLOGY SUMMARY

The proposed global copyright system is illustrated in Fig. 2 in which a copyright federation provides membership to individual countries and tokens to users. Member countries are referred to as authorities, and they maintain a consortium blockchain network by executing the Aura PoA consensus algorithm. To support copyright investigators free of charge, the federation keeps track of the blockchain containing worldwide copyright data and displays them on its official website. Copyright holders must be registered in the network by their government and must collect a token from the federation for conducting transactions. However, copyright verification does not require registration or a token. Anyone can verify the copyright of a piece of digital content (i.e., a file) simply by uploading the file or submitting its CID to the federation's website. When the file is uploaded to the website, the website computes its CID by a given formula and shows its copyright information (if available) by searching through the blockchain for the transaction that contains the CID.

The mechanisms of copyright registration, ownership transfer, and consensus are described separately in the following subsections.

B. TRANSACTION MANAGEMENT

To make a transaction, a token must be collected from the federation and attached to the transaction as a transaction fee. Fig. 3 illustrates the proposed token collection scheme in which a user collects a token from the federation in several consecutive steps. The user first sends their wallet public key Q_p to the federation. The federation checks whether the key is registered or not. If the key is found to be registered, a token code τ_c is sent to the user. The user signs the code using their wallet private key k_p and delivers the signature S_p with a token fee to the federation. The federation verifies S_p using Q_p, signs it using their private key k_F, and delivers the signature to the user. Upon receiving S_F from the federation, the user finally creates a token including τ_c, S_p, and S_F.

Algorithm 2 demonstrates the consortium network's copyright registration process. To register a copyright, a user first reads the input file content C and converts its hexadecimal value to a 256-bit private key integer k_c using the SHA256 hash function. A public key Q_c is generated by performing ECPM with k_c, which is the file's CID. A secondary private key k_s is computed by combining the primary (wallet) private key k_p and k_c, and the corresponding public key Q_s is generated. The secondary public key is a proof of content creation by the user since it is linked to the user's wallet. A tertiary private key k_{t_1} and the corresponding public key Q_{t_1} are created so that the owner can transfer ownership in the future. Because the content was not registered previously, the reference R is zero. The system defines specific rights on a creative work using a set of characters. Since the content creator has all rights on the content, they specify the transacting rights $\mathcal{R}$ as "A", where "A" means that the transaction owner can exercise all rights reserved for the content. Representing rights in a short form

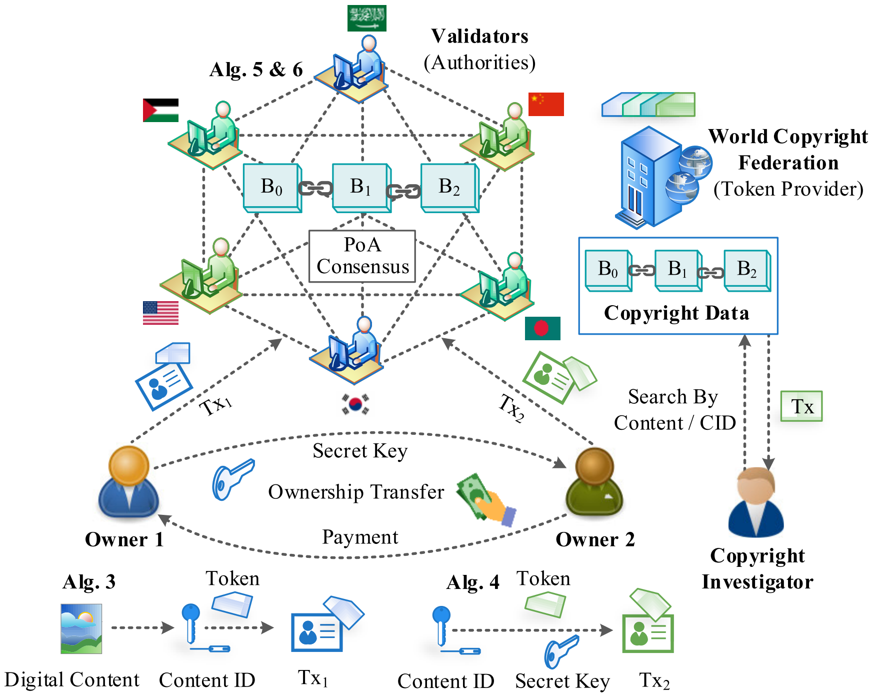

FIGURE 2. Overview of the proposed global copyright system.

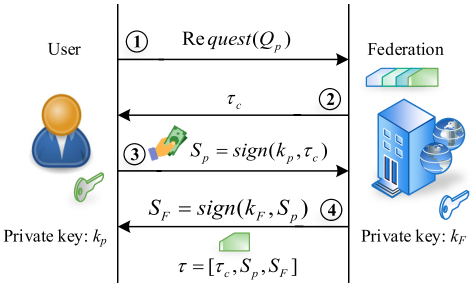

FIGURE 3. Proposed token collection scheme.

reduces the transaction size. Thus, a transaction T is formed, and it includes R, Q_p, Q_s, Q_{t_1}, the current timestamp t, a token τ, and $\mathcal{R}$. T is signed with k_s, and the signature S is inserted into the transaction. Finally, T is broadcast to the consortium network, validated by the authorities, and recorded on the blockchain.

In the case of joint authorship, the same CID is shared among all joint authors of a work. For n joint authors, n primary keys, n secondary keys, n tertiary keys, n signatures, and an optional agreement are included in a registering transaction. If the authors provide a mutual agreement, all rights and revenues are distributed among them accordingly; otherwise, they own the content equally and can exercise equal rights by default. Each joint author can transfer their ownership to a person without the consent of the other joint authors by delivering their own tertiary private key.

To sell the copyright of registered content, a seller delivers the corresponding CID Q_c, transaction hash T_h, and tertiary private key k_{t_1} to a buyer. T_h is collected from the blockchain in which the transaction regarding Q_c is stored. As shown in Algorithm 3, the buyer first checks whether T_h is a valid transaction hash or not. The buyer also confirms that the seller is the last owner of Q_c by ensuring T_h is the last transaction hash regarding Q_c. If these conditions are satisfied, the buyer conducts a transaction in a similar manner to Algorithm 2 but according to different operations regarding R and k_s. In this case, R is set to T_h, and k_s is computed by adding k_{t_1} to

```
Algorithm 2 Copyright Registration
Input: File content $C$, token $\tau$, primary private–public key pair $(k_p, Q_p)$
Read the file content $C$.
Compute the hexadecimal value of $C$: $h = hex(C)$.
Compute the hash value of $h$: $H = SHA256(h)$.
Convert $H$ to a private key integer: $k_c = int(H)$.
Generate the corresponding public key (CID): $Q_c = k_cG$.
Compute the secondary private key: $k_s = k_p + k_c$.
Generate the secondary public key: $Q_s = k_sG$.
Choose the tertiary private key $k_{t_1}$ for ownership transfer.
Generate the tertiary public key: $Q_{t_1} = k_{t_1}G$.
Set the reference: $R = 0$.
Specify the transacting rights: $\mathcal{R} = $ “A”.
Set the timestamp: $t = datetime.now()$.
Make the transaction: $T = [R, Q_p, Q_s, Q_{t_1}, Q_c, t, \tau, \mathcal{R}]$.
Sign the transaction with $k_s$: $S = sign(k_s, T)$.
Include the signature:
    $T = [R, Q_p, Q_s, Q_{t_1}, Q_c, t, \tau, \mathcal{R}, S]$.
Broadcast $T$ to the consortium network.
Store $\{k_{t_1}, T\}$ in a safe place.
```

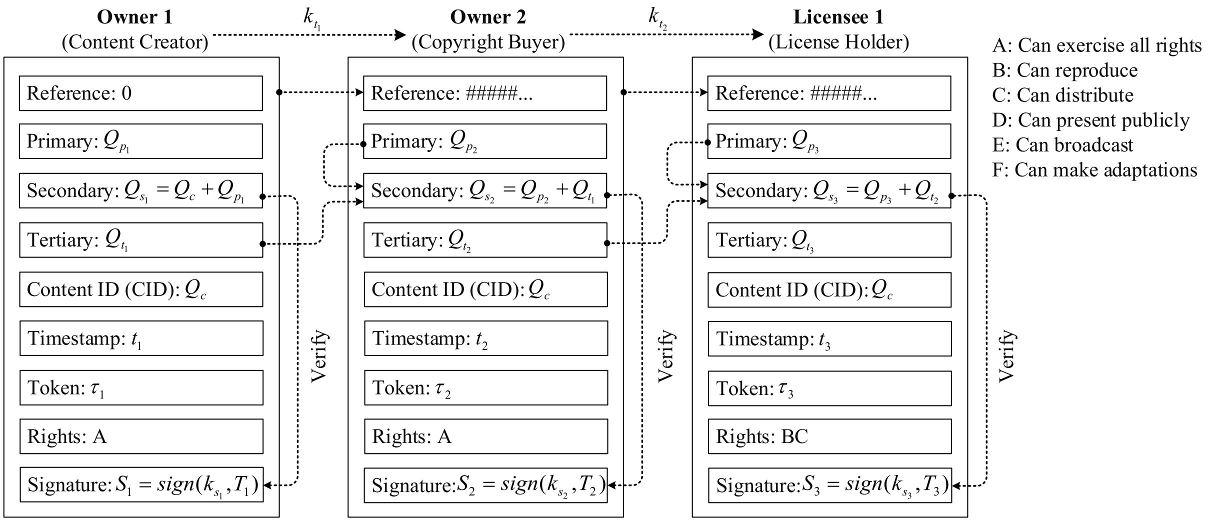

FIGURE 4. Transaction chain of copyright registration and trading.

Algorithm 3 Copyright Transfer

```
Input: CID $Q_c$, token $\tau$, transaction hash $T_h$, seller's transfer key $k_{t_1}$, primary private–public key pair $(k_p, Q_p)$
Confirm that the reference hash $T_h$ exists in the blockchain.
Confirm that $T_h$ is the last transaction hash regarding $Q_c$.
Compute the secondary private key: $k_s = k_p + k_{t_1}$.
Generate the secondary public key: $Q_s = k_s G$.
Choose the tertiary private key $k_{t_2}$ for ownership transfer.
Generate the tertiary public key: $Q_{t_2} = k_{t_2} G$.
Set the reference: $R = T_h$.
Specify the transacting rights: $\mathcal{R} =$ "A".
Set the timestamp: $t = datetime.now()$.
Make the transaction: $T = [R, Q_p, Q_s, Q_{t_2}, Q_c, t, \tau, \mathcal{R}]$.
Sign the transaction with $k_s$: $S = sign(k_s, T)$.
Include the signature:
    $T = [R, Q_p, Q_s, Q_{t_2}, Q_c, t, \tau, \mathcal{R}, S]$.
Broadcast $T$ to the consortium network.
Store $\{k_{t_2}, T\}$ in a safe place.
```

the buyer's primary private key k_p, which is proof of the ownership transfer. Since the seller assigns the content to the buyer, the buyer can exercise all the rights on the content and hence $\mathcal{R} =$ "A".

Fig. 4 depicts the transaction chain of assignment and license for a digital item. Each party holds three public keys for the item, which are referred to as primary, secondary, and tertiary keys. The primary public key is the owner's wallet address, the secondary public key is the proof of content creation or purchase, and the tertiary public key is required for ownership transfer. A token is required to provide the service charge in terms of a transaction fee to the authorities who maintain the network transparency by validating transactions and managing the blockchain. It can be collected from the federation. Since the proof of copyright purchase is verified using the current owner's secondary public key, the key is generated by adding the primary public key of the current owner to the tertiary public key of the previous owner from whom the copyright is transferred. Thus, a chain of ownership transfers is established. To protect a transaction from being tampered with, it is signed with the corresponding private key of the secondary public key that the transaction contains. Thus, the signature can be verified using the secondary public key, which proves that the sender truly has the private key with which the transaction was signed.

C. CONSENSUS ALGORITHM

A consensus among the authorities is required to record incoming transactions on the blockchain. The authorities are assumed to be synchronous within one epoch time t_e where the creation of the first or genesis block is considered an epoch. For a certain time, t_e is calculated as follows:

$$t_e = t - t_0 \tag{10}$$

where t is the *Unix* timestamp of that particular time, and t_0 is the *Unix* timestamp of the genesis block.

If N is the number of authorities and δ is the length of each step, the step number s is obtained as $s = t_e/\delta$, and the current leader is identified as A_i where $i = s\%N$. The step duration depends on N, the authorities' average mining power or validation speed, and the number of transactions stored per block (block volume) [37]. The value of δ is chosen such that no authority lags behind during transaction validation, block creation, and acceptance reception because of slow speed or poor network connectivity. In addition, minimum requirements for processor configuration and network connectivity can be set for the authorities. Each authority has two separate local queues: a transaction queue q_T (mempool) and a pending block queue q_B. The current leader at step s creates a block that includes valid transactions

Algorithm 4 Transaction Verification

```
Input: Transaction $T = [R, Q_p, Q_s, Q_t, Q_c, t, \tau, \mathcal{R}, S]$, federation's public key $Q_F$
Output: True/False
Validity=False
if $T[1]$ is registered and verify($T[2], T[0:8], T[8]$)=True then
    if verify($T[1], T[6], [0], T[6], [1]$)=True then
        if verify($Q_F, T[6], [1], T[6], [2]$)=True then
            if $T[6]$ is not already spent then
                if $T[0] = 0$ then
                    if $T[1] + T[4] = T[2]$ then
                        Validity=True
                    end if
                else
                    Find the transaction chain of the CID $T[4]$
                    Find the last transaction hash $T_{h_l}$.
                    Find the tertiary key of the last owner $Q_{t_l}$.
                    if ($T_{h_l} = T[0]$) and ($Q_{t_l} + T[1] = T[2]$) then
                        Validity=True
                    end if
                end if
            end if
        end if
    end if
end if
return Validity
```

from q_T and sends it to the non-leader authorities. If no transaction is available at this step, an empty block is sent.

The transaction validation mechanism is described by Algorithm 4 in which an authority accepts a transaction T and the federation's public key Q_F as inputs. First, the authority inquires whether the primary public key $T[1]$ is registered as well as whether the transaction signature $T[8]$ is valid or not. If the transaction requester is found to be a valid user of the system and $T[8]$ is verified with the secondary public key $T[2]$, the validation process continues. Second, the authority validates the token $T[6]$ by verifying the signatures $T[6]$, [1] and $T[6]$, [2] using $T[1]$ and Q_F, respectively. If the token is found to be valid and not previously spent, the authority checks whether the reference $T[0]$ is zero or not. If $T[0]$ is zero, T is considered a copyright registration request. For a valid transaction, the point addition of $T[1]$ and the CID $T[4]$ must be equal to $T[2]$. If $T[0]$ is nonzero, T is considered a copyright transfer request. In such a case, the owner of the referenced transaction must be the last owner of the content whose ownership is to be transferred, and the point addition of the tertiary public key of the last owner Q_{t_l} and $T[1]$ must be equal to $T[2]$. If all validation criteria are satisfied, the transaction is confirmed for storage in a block.

Algorithm 5 demonstrates how an authority creates and broadcasts a block using valid transactions. A number of valid transactions V_T, the previous block hash B_h^-, and the private–public key pair (k_A, Q_A) of the authority are accepted as inputs. First, a Merkle tree M_t is generated for V_T where M_r and M_h denote the Merkle root and hash, respectively. Each transaction of V_T is identified by a transaction index T_i and transaction hash T_h, which are inserted at the start of the transaction. After indexing all transactions, a block B is created including block index B_i, hash B_h, previous hash B_h^-, timestamp t, nonce B_n, M_r, V_T, miner B_m (Q_A), and size B_s. To broadcast the block to the network, the authority must sign it with k_A and provide a signature S_A so that other authorities can authenticate the block by verifying S_A using B_m.

Algorithm 5 Block Generation [37]

```
Input: Valid transactions $V_T$, authority's private–public key pair $(k_A, Q_A)$, previous block hash $B_h^-$
Output: Block with miner's signature
Make a Merkle tree $M_t = \{M_r, M_h\}$ for $V_T$.
for $i$ in range(0, $len(V_T)$) do
    Define the transaction number: $T_i \leftarrow i$.
    Define the transaction hash: $T_h \leftarrow M_h[i]$.
    Add the transaction ID: $V_T[i] = [T_i, T_h] + V_T[i]$
end for
Pick the block nonce: $B_n = os.urandom(8).hex()$.
Get the block timestamp: $t = datetime.now()$.
Calculate the block hash: $B_h = SHA256(B_h^- || t || B_n || M_r)$.
Identify the block miner: $B_m = Q_A$.
Generate the block: $B = \{B_i, B_h, B_h^-, t, B_n, M_r, V_T, B_m\}$.
Measure the block size: $B_s = getsizeof(B)$.
Mention $B_s$ in $B$:
    $B = \{B_i, B_h, B_h^-, t, B_n, M_r, V_T, B_m, B_s\}$.
Sign the block for authentication: $S_A = sign(k_A, B)$.
return $\{B, S_A\}$
```

After receiving and authenticating B, each authority checks the correctness of the block hash and verifies all transactions contained in the block. Then, they store it in q_B and send their acceptance feedback to all other authorities. As soon as the block receives $N/2+1$ acceptances, it is moved from q_B to the blockchain, and the common transactions between the block and q_T are removed from q_T.

D. LATENCY ANALYSIS

If n, t_v, T_s, and D_r are the block volume (the number of transactions per block), transaction verification time, transaction size, and data rate, respectively, the minimum latency of a block being proposed and validated by the network authorities, block time B_t, can be estimated as follows:

$$B_t = nt_v + \frac{nT_s}{D_r} + nt_v + \frac{nT_s}{D_r} \tag{11}$$

For simplicity, it is assumed that the block generation time and the verification time of n transactions are equal. If $n \geq 10$, computing the block hash takes negligible time in comparison with the total transaction verification time nt_v.

The step duration must be sufficiently long such that $\delta > B_t$; otherwise, slow step leaders miss their turns. The number of transactions per second (TPS) the system can confirm (i.e., transaction throughput) is obtained as follows:

$$TPS = \frac{n}{\delta} \tag{12}$$

All the proposed algorithms have a computational complexity of $\mathcal{O}(n)$.

IV. SECURITY ANALYSES

Lemma 1: No one can retrieve other's private key.

Solving the ECDLP is the sole means for an adversary $\mathcal{A}$ to compute the private key k of a participant. The most basic formula to solve the ECDLP is extensive search in which $\mathcal{A}$ calculates the series of points $P, 2P, 3P, 4P, \ldots$ until Q is obtained as shown in Algorithm 6. The algorithm continuously adds P to a temporary variable Z and checks whether Z is equal to Q until the iteration number i reaches q. If Z is equal to Q for any value of i, the private key is revealed as i; otherwise, $\mathcal{A}$ fails to retrieve the private key. It needs q and $q/2$ point additions in the worst and average cases, respectively. So far, the best general-purpose algorithm for this computation is the Pollard's rho algorithm [13], which has an expected running time of $\sqrt{\pi q}/2$ point additions. The computation speed can further be improved by involving m processors in parallel, where the speed linearly increases with the number of processors utilized. For this arrangement, the computation time is reduced to $\sqrt{\pi q}/(2m)$ point additions.

It is not supported by mathematics that the ECDLP is impossible to solve. Theoretically, ECDLP is solvable but practically, it is difficult over a large prime field. To prevent this attack, carefully choosing the elliptic curve parameters with a sizable curve order (e.g., $q \geq 2^{160}$) is recommended so that $\mathcal{A}$ needs to perform an impractical number of iterations [41]. On the experimental computer (CPU: Intel Core i5-10400 @ 2.9 GHz, RAM: 48 GB), a point addition needs 8 μs. Therefore, even if $\mathcal{A}$ combined 20000 computers of such configuration to solve the ECDLP over 256-bit $\mathbb{F}_q$, it would require approximately 1.5×10^{34} point additions $\approx$ 3.8×10^{21} years, which is more than the attacker's lifetime.

Lemma 2: Proof of content creation.

During the registration process, a CID is broadcast to the network rather than revealing the content itself. The CID is a public key Q_c produced by converting the content into a private key integer k_c. Only the content owner can prove ownership of the CID by signing the registering transaction with their secondary private key k_s that is a combination of their primary private key k_p and k_c:

$$k_s = k_p + k_c \tag{13}$$

The corresponding secondary public key Q_s is computed by performing ECPM with k_s as follows:

$$Q_s = k_s G \tag{14}$$

Algorithm 6 Exhaustive Search for Solving ECDLP

Input: Base point P, target public key Q, curve order q
Output: Retrieved private key k

```
$Z = 0$
**for** $i$ in range(0, $q$) **do**
    $Z \leftarrow Z + P$
    **if** $Z = Q$ **then**
        **return** $i$
    **end if**
**end for**
**print** "Private key has not been found"
```

According to ECC:

$$k_s G = (k_p + k_c)G \tag{15}$$

$$k_s G = k_p G + k_c G \tag{16}$$

$$Q_s = Q_p + Q_c \tag{17}$$

The transaction signature is verified using Q_s. By combining k_p and k_c, a relation between the owner's wallet and the unregistered content is established.

Let $\mathcal{A}$ be an adversary whose primary private–public key pair is $\{k_p^{\mathcal{A}}, Q_p^{\mathcal{A}}\}$. $\mathcal{A}$ can generate a duplicate secondary public key $Q_s^{\mathcal{A}}$ simply by adding the CID Q_c to $Q_p^{\mathcal{A}}$, such that:

$$Q_s^{\mathcal{A}} = Q_p^{\mathcal{A}} + Q_c \tag{18}$$

Equation (18) can be represented as follows:

$$k_s^{\mathcal{A}} G = k_p^{\mathcal{A}} G + k_c^{\mathcal{A}} G \tag{19}$$

Here, the adversary knows the keys Q_c, $Q_p^{\mathcal{A}}$, $Q_s^{\mathcal{A}}$, and $k_p^{\mathcal{A}}$ but does not know the unrevealed content private key k_c and the secondary private key $k_s^{\mathcal{A}}$. However, $k_s^{\mathcal{A}}$ is required to sign the transaction and generate a valid signature. If $\mathcal{A}$ knew k_c, they could generate $k_s^{\mathcal{A}}$ as follows:

$$k_s^{\mathcal{A}} = k_p^{\mathcal{A}} + k_c \tag{20}$$

It is evident from (20) that $k_s^{\mathcal{A}}$ cannot be generated without k_c, whereas $\mathcal{A}$ cannot retrieve k_c even though they know Q_c. This constraint occurs because a public key can be computed from a private key, but the reverse operation is impossible due to the ECDLP. As a result, $\mathcal{A}$ cannot provide a valid signature, and thus their false transaction will be rejected by the network's authorities.

For security, the content creator should not reveal the content or k_c before the transaction has been confirmed and recorded on the blockchain.

Lemma 3: Proof of ownership transfer.

Suppose a copyright buyer $\{k_{p_b}, Q_{p_b}\}$ intends to buy a copyright $\{T_h, Q_c, Q_{t_s}\}$ from a seller $\{k_{p_s}, Q_{p_s}, k_{t_s}\}$. To sell the copyright, the seller receives payment from the buyer and delivers the tertiary private key k_{t_s}. Upon receiving k_{t_s} from the seller, the buyer generates a secondary private–public key pair (k_{s_b}, Q_{s_b}) as follows:

$$k_{s_b} = k_{p_b} + k_{t_s} \tag{21}$$

$$Q_{s_b} = k_{s_b} G \tag{22}$$

According to the properties of an elliptic curve:

$$k_{s_b}G = (k_{p_b} + k_{t_s})G \tag{23}$$

$$k_{s_b}G = k_{p_b}G + k_{t_s}G \tag{24}$$

$$Q_{s_b} = Q_{p_b} + Q_{t_s} \tag{25}$$

Similar to the registering transaction, the trading transaction is signed with k_{s_b}, which can be verified using Q_{s_b}. Adversary $\mathcal{A}$ having the private–public key pair $\{k_p^{\mathcal{A}}, Q_p^{\mathcal{A}}\}$ can compute $Q_{s_b}^{\mathcal{A}}$ simply by adding Q_{t_s} to $Q_{p_b}^{\mathcal{A}}$, such that:

$$Q_{s_b}^{\mathcal{A}} = Q_{p_b}^{\mathcal{A}} + Q_{t_s} \tag{26}$$

Equation (26) can be represented as follows:

$$k_{s_b}^{\mathcal{A}}G = k_{p_b}^{\mathcal{A}}G + k_{t_s}G \tag{27}$$

Here, $\mathcal{A}$ knows the keys Q_{t_s}, $Q_{p_b}^{\mathcal{A}}$, $Q_{s_b}^{\mathcal{A}}$, and $k_{p_s}^{\mathcal{A}}$ but does not know the tertiary private key k_{t_s} and the secondary private key $k_{s_b}^{\mathcal{A}}$. However, $k_{s_b}^{\mathcal{A}}$ is required to sign the transaction and generate a valid signature. If the adversary knew k_{t_s}, they could generate $k_{s_b}^{\mathcal{A}}$ as follows:

$$k_{s_b}^{\mathcal{A}} = k_{p_b}^{\mathcal{A}} + k_{t_s} \tag{28}$$

Since $\mathcal{A}$ does not know k_{t_s} and cannot retrieve it by solving the ECDLP with Q_{t_s}, they cannot provide a valid signature. Thus, they cannot claim the ownership of Q_c.

Lemma 4: Man-in-the-middle (MitM) attacks cannot succeed against the proposed system

A MitM attack is a form of hijacking wherein a transaction's destination address is substituted with the attacker's address by putting malware on the sender's device. In our transaction model, a transaction's destination address is the sender's primary or wallet public key, which links the sender's secondary public key. The secondary public key generates the transaction signature. Therefore, with the intention of becoming the copyright owner, an attacker cannot replace the sender's primary public key during the transition period (i.e., before the transaction has been recorded on the blockchain). This constraint makes the system resistant to MitM attacks. In addition, the signature is generated by signing the transaction itself, which prevents an attacker from tampering with any data in the transaction and ensures the data integrity.

Lemma 5: Although a distributed denial of service (DDoS) attack can lengthen block time and lower the transaction throughput, it cannot stop the copyright service from operating.

Similar to other BFT and crash fault tolerance consensus algorithms, PoA is vulnerable to DDoS attacks [37], [41]. Aura allows only the leader node at each step to propose block. As a result, block time can be lengthen, resulting in a reduction of the transaction throughput, if step leaders are repeatedly targeted by DDoS attacks. The slow internet connection of some nodes can complicate this problem. Any consensus node can be the target of a DDoS attack. However, the majority of consensus nodes must fail to challenge the fault tolerance property of PoA. Consequently, the likelihood of such an attack being successful is low unless the network is fragile and contains a significant number of unprotected nodes.

V. IMPLEMENTATION RESULTS

A. EXPERIMENT

The proposed system was implemented in Python with five virtual machines on the Oracle VM VirtualBox 6.1 installed in a host computer (CPU: Intel Core i5-10400 @2.9 GHz, RAM: 48 GB). Considering a real-time environment, we made a variation between the computational power of the virtual machines used to conduct the experiment. To vary the computational power, we distributed the host memory and cores among the virtual machines as follows: VM0 (RAM: 4 GB, Core: 1), VM1 (RAM: 16 GB, Cores: 4), VM2 (RAM: 12 GB, Cores: 3), VM3 (RAM: 8 GB, Cores: 2), and VM4 (RAM: 4 GB, Cores: 1), each with different IP addresses. The four nodes VM1, VM2, VM3, and VM4 were considered as four authorities, here referred to as authorities A, B, C, and D, respectively. A full node for the copyright federation, which provides tokens and holds a copy of the complete blockchain, as well as a light node for a user who makes transactions without downloading the blockchain were installed in VM0. A round-robin schedule was set as A→B→C→D→A and used to select the step leaders of the PoA consensus. The system's sensitivity was tested by conducting separate experiments with variable step duration. The user continuously generated transactions and broadcast them to the consortium network using the user datagram protocol that allows a maximum packet size of 65 KB per transmission. In the propose system, 70 transactions roughly made a block size of less than 65 KB. Therefore, we defined the block volume as 70. We expected that no consensus node in the network would miss a turn due to a slow processor or poor internet connection. In the network, the slowest computer took 24.28 ms to verify per transaction. The average block size and data rate were 53 KB and 261 Kbps, respectively. As a result, the minimum block time was 4.6 s according to (11).

In PoA, a block must receive acceptance from the majority of authorities to be added to the blockchain. Therefore, for four authorities, the proposed block at a given step was added to the blockchain as soon as it received acceptance from any two non-leader authorities of that step. At each step, the step leader created a block with valid transactions (if available), broadcast the block to the network, and received at least two block acceptances. All these tasks should be completed within time period δ. If the total latency exceeded δ and the next block appeared, the current block was not accepted. However, if a step leader did not receive sufficient acceptances of the previous step leader's block before creating the new block, two consecutive blocks would contain the same block index and common transactions. In such a case, the former was accepted if it received sufficient acceptances before the appearance of the latter block, and

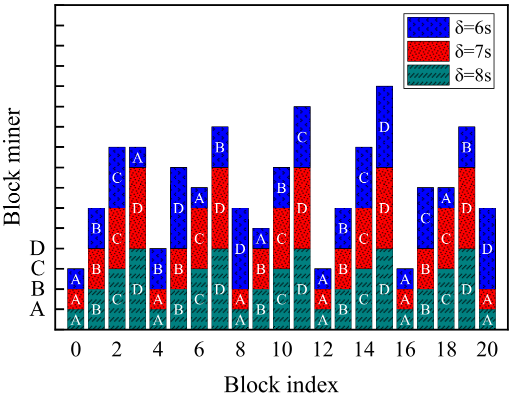

FIGURE 5. Miners of the first 20 blocks for variable step duration.

the latter was rejected when it was proposed. Rejection of a block or missing a turn means increasing the block time by an interval of δ, which is unexpected. Setting the step duration to δ means the network expects to add blocks to the blockchain after each δ time interval. Therefore, the value of δ should be sufficient for a block to be proposed by the current step leader and added by the next step leader to the blockchain with required acceptances within this time frame.

B. PERFORMANCE ANALYSES

Since $\delta > B_t$, the step duration must be greater than 4.6 s. Therefore, we increased the step duration from 5 s to determine the optimum value. It was observed that no authority missed a turn when the step duration was 7 s or 8 s, whereas several turns were missed when the step duration was 5 or 6 s. As the step duration decreased, the probability that authorities missed turns increased, and vice versa. To overcome the block failure problem, the minimum step duration of the network can be set to 7 s. Fig. 5 illustrates the miners of blocks 0 to 20 for different values of δ. It was observed that no authority missed a turn when δ was 7 s or 8 s, whereas several turns were missed when δ was 6 s. As δ decreased, the probability that authorities missed turns increased, and vice versa. To overcome the block failure problem, the minimum step duration of the network can be set to 7 s. However, this setting varies from network to network depending on network's performance benchmarks, such as the incoming transaction rate, block volume, average computing power, authority response time, and data rate.

Fig. 6 depicts the epoch times of the first 20 blocks for variable step duration. The epoch time of a block is the difference between the block timestamp and the genesis block timestamp. A step duration of 7 s took the lowest time to add 20 blocks to the blockchain; hence, it is the optimum value of δ for this network. When δ was 6 s, some turns were missed in a round, increasing the total time required to mine 20 blocks. Although no turn was missed when δ was 8 s, it took 160 s to mine 20 blocks, whereas the setting $\delta = 7$ s required only 140 s to mine the same number of blocks.

Fig. 7 shows the PoA consensus times for the first 20 blocks when the step duration was set to the optimum value. The consensus process took an average of 3.98 s to finish the second round after taking 1.56 s to complete the first round. The average block time was 5.54 s, which is a total sum of the first and second round time. The average difference between the step duration and block time was 1.46 s, which acted as the safe guard for slower consensus nodes. However, this latency is considered as the system loss in terms of time [37].

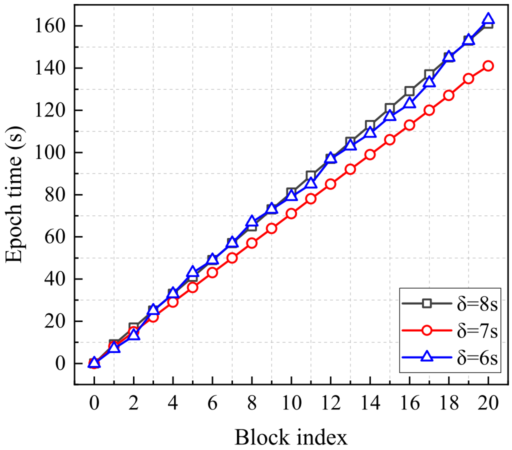

FIGURE 6. Epoch times of the first 20 blocks for variable step duration.

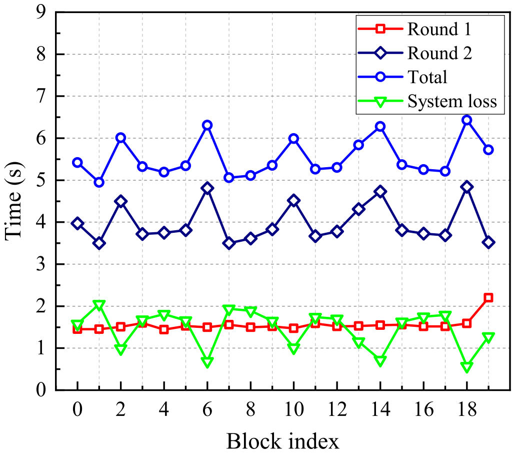

FIGURE 7. PoA consensus times for the first 20 blocks when $\delta = 7s$.

Table 2 presents the overall performance of the proposed system. The data rate of individual authorities was measured using the Wireshark 3.4.6 network traffic monitoring tool. Authority A provided the highest computing power and data rate because it had the best hardware configuration. The performance of the authorities decreased in order from A to D. The ECPM time required by the authorities was 3.6 ms. The signature generation and verification time was 3.78 and 7.4 ms, respectively, on average. The transaction size was 770 B only, and the block size for 70 TPB was 52.95 KB. The average data rate, transaction verification time, and offset delay over the network were 261.43 Kbps, 22.21 ms, and 2.42 s, respectively. The system processed 10 TPS in the testing environment.

TABLE 2. Overall Performance of The Proposed System.

Authorities	A	B	C	D
ECPM time (ms)	3.6	3.6	3.6	3.6
ECDSA signing time (ms)	3.7	3.7	3.8	3.9
ECDSA verification time (ms)	7.3	7.4	7.4	7.6
Transaction verification time (ms)	21.28	21.57	21.71	24.28
Block preparation time (s)	1.49	1.51	1.52	1.70
Data rate (Kbps)	261.25	261.24	261.28	261.86
Average data rate (Kbps)	261.43			
Maximum data rate (Mbps)	1.11			
Transaction size (B)	770			
Block volume (TPB)	70			
Block size (KB)	52.95			
Average verification time (ms)	22.21			
Step duration (s)	7			
Average block time (s)	5.54			
Transaction throughput (TPS)	10			

VI. LIMITATIONS AND SCOPES

The system's limited throughput (10 TPS) is a drawback. However, to use blockchain for copyright management, transaction speed has to be compromised. The Bitcoin and Ethereum cryptocurrencies maintain a throughput of approximately 3 and 13 TPS, respectively, although the speed of a financial transaction is more important than that of a copyrighting transaction. In addition, ECPM speed can be increased by using the field programmable gate array [33]. Since the experimental computer operated with a RAM of 48 GB only and the five virtual machines occupied a total memory of 44 GB, we could not install additional virtual machines in the host computer. As a result, we had to test the system using only four consensus nodes.

Because the round-robin schedule is fixed in our Aura implementation, transaction throughout of the network decreases if a certain authority is offline. To improve the throughput, we can use a dynamic leader selection approach based on the computing power and network connectivity of individual authorities using federated learning [45], which is left for the future work.

Our system is not only suitable for cross-border copyright protection but also applicable for auditable cryptocurrencies. Governments around the world can launch a global digital currency to provide cross-border payments and reduce the costs and transaction delays associated with the traditional society for worldwide interbank financial telecommunication-based payments [37]. They can manage a consortium blockchain, as proposed in this paper, with equal rights to make profits from block rewards and transaction fees. An auditable cryptocurrency can help them to prevent their citizens from engaging in money laundering and other criminal activities through existing non-auditable cryptocurrencies while providing citizens with the opportunity of using digital currency in legal ways. However, the privacy and double-spending issues must be solved. While copyright data does not require privacy, and no owner double-registers their work, a financial transaction requires privacy and can be double-spent.

VII. CONCLUSION AND FUTURE WORK

In this paper, a decentralized copyright system has been proposed for cross-border copyright protection based on a consortium blockchain with the PoA consensus algorithm. Unlike conventional copyright systems, the proposed system does not require any centralized server to store copyrighted content. Instead, blockchain is used to store the content metadata. This model creates a synchronized platform for copyright investigation of worldwide digital content. Copyright is provided to registered users only so that punishment can be applied to copyright infringements. Before use and share, people can investigate copyrighted works to avoid violation of copyright laws. Thus, the system can reduce copyright infringements. As part of future work, we plan to improve the transaction throughput of the system by sharding the blockchain. It would split the network into separate shards and enable load balancing by parallel processing of transactions.

REFERENCES

[1] M. Trimble, "Undetected conflict-of-laws problems in cross-border online copyright infringement cases," *NCJL & Tech.*, vol. 18, no. 1, pp. 119–159, 2016.

[2] E. Fiordalisi, "The tangled web: Cross-border conflicts of copyright law in the age of internet sharing," *Loy. U. Chi. Int. L. Rev.*, vol. 12, no. 2, pp. 197–213, 2015.

[3] L. Ruth Okediji, "The limits of international copyright exceptions for developing countries," *Vanderbilt JETLaw*, vol. 21, no. 3, p. 689, 2019.

[4] S. Garfield. (2022). *5 Challenges for Global Copyright*. Copyright Clearance Center. Accessed: Aug. 7, 2022. [Online]. Available: https://www.copyright.com/blog/5-challenges-global-copyright

[5] F. Cantatore, *Publishing and the Law: Copyright and Globalisation*, D. Baker, D. Brien, and J. Webb, Eds. Newcastle Upon Tyne, U.K.: Cambridge Scholars Publishing, 2019, pp. 41–64.

[6] A. Savelyev, "Copyright in the blockchain era: Promises and challenges," *Comput. Law Secur. Rev.*, vol. 34, no. 3, pp. 550–561, Jun. 2018.

[7] B. Bhushan, A. Khamparia, K. M. Sagayam, S. K. Sharma, M. A. Ahad, and N. C. Debnath, "Blockchain for smart cities: A review of architectures, integration trends and future research directions," *Sustain. Cities Soc.*, vol. 61, Oct. 2020, Art. no. 102360.

[8] B. Bhushan, P. Sinha, K. M. Sagayam, and J. Andrew, "Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions," *Comput. Electr. Eng.*, vol. 90, Mar. 2021, Art. no. 106897.

[9] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1432–1465, 2nd Quart., 2020.

[10] S. Kaur, S. Chaturvedi, A. Sharma, and J. Kar, "Research survey on applications of consensus protocols in blockchain," *Secur. Commun. Netw.*, vol. 2021, pp. 1–22, Jan. 2021.

[11] *Compendium of U.S. Copyright Office Practices*, 3rd Ed. Washington, DC, USA: U.S. Copyright Office, Dec. 2014, pp. 1–1301.

[12] J. Benet, "IPFS–content addressed, versioned, P2P file system," 2014, *arXiv:1407.3561*.

[13] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2004.

[14] R. Sinhal and I. A. Ansari, "A multiple transform based approach for robust and blind image copyright protection," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 28, pp. 1–16, Oct. 2022.

[15] Z. Xia, X. Wang, C. Wang, C. Wang, B. Ma, Q. Li, M. Wang, and T. Zhao, "A robust zero-watermarking algorithm for lossless copyright protection of medical images," *Int. J. Speech Technol.*, vol. 52, no. 1, pp. 607–621, May 2021.

[16] H. M. Al-Otum, "Dual image watermarking using a multi-level thresholding and selective zone-quantization for copyright protection, authentication and recovery applications," *Multimedia Tools Appl.*, vol. 81, no. 18, pp. 25787–25828, Mar. 2022.

[17] A. Qureshi and D. M. Jiménez, ''Blockchain-based multimedia content protection: Review and open challenges,'' *Appl. Sci.*, vol. 11, no. 1, p. 1, Dec. 2020.

[18] Z. Ma, M. Jiang, H. Gao, and Z. Wang, ''Blockchain for digital rights management,'' *Future Gener. Comput. Syst.*, vol. 89, pp. 746–764, Dec. 2018.

[19] Z. Ma, W. Huang, and H. Gao, ''Secure DRM scheme based on blockchain with high credibility,'' *Chin. J. Electron.*, vol. 27, no. 5, pp. 1025–1036, Sep. 2018.

[20] B. Wang, S. Jiawei, W. Wang, and P. Zhao, ''Image copyright protection based on blockchain and zero-watermark,'' *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 4, pp. 2188–2199, Jul. 2022.

[21] S. Zhao and D. O'Mahony, ''BMCProtector: A blockchain and smart contract based application for music copyright protection,'' in *Proc. Int. Conf. Blockchain Technol. Appl.*, Dec. 2018, pp. 1–6.

[22] D. Bhowmik and T. Feng, ''The multimedia blockchain: A distributed and tamper-proof media transaction framework,'' in *Proc. 22nd Int. Conf. Digit. Signal Process. (DSP)*, Aug. 2017, pp. 1–5.

[23] W. Peng, L. Yi, L. Fang, D. XinHua, and C. Ping, ''Secure and traceable copyright management system based on blockchain,'' in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2019, pp. 1243–1247.

[24] X. Wu, P. Ma, Z. Jin, Y. Wu, W. Han, and W. Ou, ''A novel zero-watermarking scheme based on NSCT-SVD and blockchain for video copyright,'' *EURASIP J. Wireless Commun. Netw.*, vol. 2022, no. 1, pp. 1–21, Mar. 2022.

[25] J. Chi, J. Lee, N. Kim, J. Choi, and S. Park, ''Secure and reliable blockchain-based eBook transaction system for self-published eBook trading,'' *PLoS ONE*, vol. 15, no. 2, Feb. 2020, Art. no. e0228418.

[26] B. Zhao, L. Fang, H. Zhang, C. Ge, W. Meng, L. Liu, and C. Su, ''Y-DWMS: A digital watermark management system based on smart contracts,'' *Sensors*, vol. 19, no. 14, p. 3091, Jul. 2019.

[27] Y. Ouyang, X. Zheng, X. Lu, L. Xiaowei, and S. Zhang, ''Copyright protection application based on blockchain technology,'' in *Proc. IEEE Int. Conf. Parallel Distrib. Process. With Appl., Big Data Cloud Comput., Sustain. Comput. Commun., Social Comput. Netw. (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2019, pp. 1271–1274.

[28] Y. Liu, J. Zhang, S. Wu, and M. S. Pathan, ''Research on digital copyright protection based on the hyperledger fabric blockchain network technology,'' *PeerJ Comput. Sci.*, vol. 7, no. e709, pp. 1–26, Sep. 2021.

[29] G. Heo, D. Yang, I. Doh, and K. Chae, ''Efficient and secure blockchain system for digital content trading,'' *IEEE Access*, vol. 9, pp. 77438–77450, 2021.

[30] A. A. Elrowayati, M. A. Alrshah, M. F. L. Abdullah, and R. Latip, ''HEVC watermarking techniques for authentication and copyright applications: Challenges and opportunities,'' *IEEE Access*, vol. 8, pp. 114172–114189, 2020.

[31] J.-U. Hou, D. Kim, W.-H. Ahn, and H.-K. Lee, ''Copyright protections of digital content in the age of 3D printer: Emerging issues and survey,'' *IEEE Access*, vol. 6, pp. 44082–44093, 2018.

[32] *Understanding Copyright and Related Rights*, 2nd ed. Geneva, Switzerland: WIPO, 2016, pp. 1–40.

[33] T. Meng, Y. Zhao, K. Wolter, and C.-Z. Xu, ''On consortium blockchain consistency: A queueing network model approach,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 6, pp. 1369–1382, Jun. 2021.

[34] M. Saad, J. Spaulding, L. Njilla, C. Kamhoua, S. Shetty, D. Nyang, and D. Mohaisen, ''Exploring the attack surface of blockchain: A comprehensive survey,'' *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1977–2008, 3rd Quart., 2020.

[35] M. Mazzoni, A. Corradi, and V. Di Nicola, ''Performance evaluation of permissioned blockchains for financial applications: The ConsenSys quorum case study,'' *Blockchain: Res. Appl.*, vol. 3, no. 1, Mar. 2022, Art. no. 100026.

[36] Z. Su, Y. Wang, Q. Xu, M. Fei, Y.-C. Tian, and N. Zhang, ''A secure charging scheme for electric vehicles with smart communities in energy blockchain,'' *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4601–4613, Jun. 2019.

[37] M. M. Islam, M. K. Islam, M. Shahjalal, M. Z. Chowdhury, and Y. M. Jang, ''A low-cost cross-border payment system based on auditable cryptocurrency with consortium blockchain: Joint digital currency,'' *IEEE Trans. Services Comput.*, early access, Sep. 16, 2023, doi: 10.1109/TSC.2022.3207224.

[38] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, ''PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain,'' in *Proc. Italian Conf. Cyber Secur.*, Milan, Italy, Jun. 2018, pp. 1–11.

[39] D. Johnson, A. Menezes, and S. Vanstone, ''The elliptic curve digital signature algorithm (ECDSA),'' *Int. J. Inf. Secur.*, vol. 1, no. 1, pp. 36–63, Aug. 2001.

[40] M. Qu, *SEC2: Recommended Elliptic Curve Domain Parameters*, document SEC2-Ver-1.0, Certicom Res., Mississauga, ON, Canada, 1999.

[41] M. M. Islam and H. P. In, ''A privacy-preserving transparent central bank digital currency system based on consortium blockchain and unspent transaction outputs,'' *IEEE Trans. Services Comput.*, early access, Dec. 1, 2022, doi: 10.1109/TSC.2022.3226120.

[42] M. M. Islam, M. S. Hossain, M. K. Hasan, M. Shahjalal, and Y. M. Jang, ''FPGA implementation of high-speed area-efficient processor for elliptic curve point multiplication over prime field,'' *IEEE Access*, vol. 7, pp. 178811–178826, 2019.

[43] M. S. Hossain and Y. Kong, ''High-performance FPGA implementation of modular inversion over F_256 for elliptic curve cryptography,'' in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 169–174.

[44] A. Menezes, *Evaluation of Security Level of Cryptography: The Elliptic Curve Discrete Logarithm Problem (ECDLP)*. Waterloo, ON, Canada: Univ. Waterloo, 2001.

[45] Y. Wang, H. Peng, Z. Su, T. H. Luan, A. Benslimane, and Y. Wu, ''A platform-free proof of federated learning consensus mechanism for sustainable blockchains,'' *IEEE J. Sel. Areas Commun.*, vol. 40, no. 12, pp. 3305–3324, Dec. 2022.

MD. MAINUL ISLAM (Graduate Student Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Bangladesh, in April 2018, and the M.Sc. degree in electronics engineering from Kookmin University, South Korea, in February 2021. He is currently pursuing the Ph.D. degree in computer engineering with the King Fahd University of Petroleum and Minerals (KFUPM), Saudi Arabia. After completing the M.Sc. degree, he joined the Intelligent Blockchain Engineering Laboratory (IBEL), Department of Computer Science and Engineering, Korea University, South Korea, as a full-time Researcher. He collaborates with IBEL as a Remote Researcher. His research interests include blockchain, cross-border payment, central bank digital currency, cryptography, and self-adaptive cybersecurity. For the M.Sc. degree, he received the Academic Excellence Award. He has served as a reviewer for IEEE, Elsevier, Wiley, and MDPI publishing journals.

HOH PETER IN (Member, IEEE) received the B.Sc. degree in computer engineering and the M.Sc. degree in computer science from Korea University, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Southern California, USA, in 1998. In 1999, he was an Assistant Professor with Texas A&M University, USA. He joined the Department of Computer Science, Korea University, as an Assistant Professor, in 2003, where he is currently a Professor. He is also the Founder and the Emeritus President of the Korea Society of Blockchain and the Director of the Blockchain Research Institute, South Korea. He has published more than 120 research articles. His main research interests include blockchain, smart contracts, and software engineering. He received the ICRE Ten Years Most Influential Paper Award, in 2006.