future internet

MDPI

Article

Blockchain and Smart Contracts for Digital Copyright Protection

Franco Frattolillo

Department of Engineering, University of Sannio, Corso Garibaldi 107, 82100 Benevento, Italy; frattolillo@unisannio.it; Tel.: +39-0824305806

Abstract: In a global context characterized by a pressing need to find a solution to the problem of digital copyright protection, buyer-seller watermarking protocols based on asymmetric fingerprinting and adopting a "buyer-friendly" approach have proven effective in addressing such a problem. They can ensure high levels of usability and security. However, they usually resort to trusted third parties (TTPs) to guarantee the protection process, and this is often perceived as a relevant drawback since TTPs may cause conspiracy or collusion problems, besides the fact that they are generally considered as some sort of "big brother". This paper presents a buyer-seller watermarking protocol that can achieve the right compromise between usability and security without employing a TTP. The protocol is built around previous experiences conducted in the field of protocols based on the buyer-friendly approach. Its peculiarity consists of exploiting smart contracts executed within a blockchain to implement preset and immutable rules that run automatically under specific conditions without control from some kind of central authority. The result is a simple, usable, and secure watermarking protocol able to do without TTPs.

Keywords: digital copyright protection; blockchain; smart contracts; digital rights management; watermarking protocols

check for updates

Citation: Frattolillo, F. Blockchain and Smart Contracts for Digital Copyright Protection. *Future Internet* **2024**, *16*, 169. https://doi.org/10.3390/fi16050169

Academic Editor: Massimo Cafaro

Received: 8 April 2024
Revised: 8 May 2024
Accepted: 12 May 2024
Published: 14 May 2024

1. Introduction and Motivations

The rapid and continuous spread of social networks and advances in multimedia technologies have turned ordinary web users into producers of multimedia digital content. Such users wish to control the dissemination of their digital content, which adds relevance to the problem of digital copyright protection. More precisely, problems arise because current multimedia and network technologies enable web users to obtain digital content legitimately and then illicitly exchange it through file-sharing applications. Duplication, modification, and re-distribution of contents do not reduce their perceived quality. Therefore, the inadequate digital copyright protection of content financially damages professional content providers and leads to serious privacy issues for ordinary internet users.

Even though the scientific community is actively studying the problem of digital copyright protection, the proposed solutions still fail to reconcile the conflicting interests of those who provide content and those who want to enjoy it. The former demand a high remuneration for their contents and low protection and distribution costs. The latter mainly want to preserve their privacy. Consequently, they are inclined to refuse protection mechanisms that could identify them. However, they also wish to achieve further goals, such as having complete control, according to the "fair use" doctrine, over the reproduction and distribution of the purchased contents and paying as little as possible for the contents available on the Internet.

Among the different solutions proposed over the years, "watermarking protocols" represent a convincing alternative [1]. Such protocols define the transaction schemes by which digital watermarked contents have to be traded through the Internet [2]. They exploit digital watermarking techniques to insert hidden information, called "watermarks", into the contents before they are commercialized. Watermarks represent actual "fingerprints" that can identify both the providers of the contents and their buyers. As a result, the

watermarks embedded into the contents released according to watermarking protocols make it possible to identify the copyright "infringers", that is, the legitimate buyers of contents who have illegally shared them on the Internet: it is sufficient to discover copies of such content and extract the watermarks from them. Therefore, the more content is illegally shared, the greater the chances of discovering and punishing copyright infringements.

The current watermarking protocols use different types of protection schemes based on a wide variety of security primitives. The result is that they achieve different results in terms of security and practicability for the web context. In particular, the protocols based on the well-known "buyer-seller" scheme [1,3] are characterized by high levels of security. However, they often fail to meet relevant design requirements, and this condition makes them impracticable in the current web context [1]. In this regard, a questionable issue of the implementation of buyer-seller watermarking protocols is represented by the involvement of "watermark certification authorities" (WCAs) acting as trusted third parties (TTPs) in the protection schemes [1,3,4]. In such schemes, the WCAs guarantee the transactions by which buyers get digitally watermarked contents from sellers. However, WCAs can be a source of collusion with buyers or sellers [2], and this condition represents a severe security problem for the protocols. In order to remedy such a problem, several watermarking protocols do not employ WCAs [5–8]. These protocols improve security, but the absence of WCAs makes the participation of buyers in the purchase transactions difficult because of complex security actions. Consequently, such protocols are inadequate for the current web context [1].

A compromise solution is obtained by adopting the "buyer-friendly" and "mediated" design approach described in [9]. It reintroduces the WCA but with a limited role. This means that the WCA has limited involvement in the protocol, just enough to support both a high-security level of the protocol and an easy involvement of buyers in the purchase transactions.

Although they represent a good balance between usability and security, the watermarking protocols based on a limited involvement of WCAs still need to be improved to meet the requirements of the current web context [1]. The first aspect to intervene in is the removal of WCAs without reducing the usability of the protocols. To this end, solutions can be found in the field of blockchains used to protect digital copyright [10–12].

A blockchain can be considered a distributed ledger that can record information on commercial or network transactions in blocks chained using several specific technologies. In particular, each block is digitally signed and chained after the insertion of two tokens: a timestamp and a cryptographic hash calculated on the previous block. As a result, the chained blocks cannot be modified. They are validated by running a decentralized "consensus algorithm" on computing web nodes to determine whether a transaction is valid.

Moreover, blockchains can take advantage of a specific facility represented by the so-called "smart contracts". Such contracts consist of code that is automatically run when preset conditions and requirements are fulfilled without the need for a central authority. The code unalterably implements both the conditions set out in an agreement reached by distinct web parties and the rules and events that can trigger the automatic execution of the contracts [13–15].

This paper describes a new watermarking protocol that exploits blockchain technology to avoid using WCA. The protocol is based on the "buyer-friendly" design approach [16,17], which has already proven effective in meeting the usability requirements documented in [1]. Moreover, it employs blockchain and smart contracts to implement and efficiently register verifiable transactions among the entities negotiating digital content on the web. The result is a protocol that is both usable and secure without resorting to TTPs, and this peculiarity makes it suited for the current web context.

2. Related Work

The first buyer-seller watermarking protocols exploit WCAs and public-key infrastructures (PKIs) based on "privacy-homomorphic" encryption schemes [18]. WCAs can thus

encrypt watermarks with the buyers' public keys and then directly embed them into the contents encrypted by sellers with the same buyers' public keys. Buyers can decrypt the requested contents by applying their private keys and thus obtain their copies protected by personalized watermarks, which are unknown to sellers [3].

The first protocols based on WCAs and privacy-homomorphic encryption schemes are characterized by several problems, the most important of which are the "unbinding problem" [3] and the collusion problems caused by malicious WCAs [2]. Their solution has led to protocols that do not employ WCAs [19] or that use off-line WCAs [4,20].

The different approach based on the "client-side embedding" protection scheme carries out a joint watermark insertion between seller and buyer by employing symmetric ciphers and "partial encryption" [21]. The seller adds a noise sequence to a set of selected transform coefficients of content in order to distort them. The buyer then employs a specific decryption key received by the seller to partially remove the added noise sequence and thus leave only the watermark. Such a scheme is characterized by a double advantage: it avoids WCAs and is scalable since it distributes the computing load generated by watermark insertion equally between buyer and seller. However, it is still affected by the "customer's rights problem" and collusion problems, which are caused by the fact that sellers have access to the decryption keys needed to partially remove the noise sequences so as to leave client-specific watermarks in the contents.

The promising scheme described in [21] is improved by the protocols proposed in [7,8], which mainly focus on the process by which fingerprints are left in the contents by decryption keys. In particular, the protocol in [7] generates decryption keys that can leave in the contents personalized binary fingerprints that are unknown to sellers. The protocol in [8], in addition, implements collusion resistance facilities within the scheme described in [7] by generating two types of fingerprints: near orthogonal independent Gaussian fingerprints and Tardos code-based fingerprints [22]. The results are represented by protocols characterized by scalability, collusion resistance, and absence of WCAs. The protocols, however, have a significant drawback: they require buyers to remove noise sequences from the purchased contents. Therefore, they force buyers to carry out complex security actions [1].

The protocols described in [5,6] represent an example of provably secure collusion-resistant interaction schemes. They do not use WCAs. However, they make the participation of buyers in the protocols rather difficult since buyers have to take charge of interactive zero-knowledge proofs, encryptions, group signatures, and watermark generations. Such a peculiarity makes these protocols unsuitable for the current Internet.

The buyer-seller watermarking protocols also inspire systems, known as DRM (Digital Rights Management) systems, specifically designed to protect the copyright of digital content on the Internet. DRMs are more complex than watermarking protocols since they can exploit advanced facilities, such as file-sharing peer-to-peer networks and blockchain technology, to implement specific services focused on the managed digital content and their modifications or copyright transfers [23,24].

DRMChain is a DRM system based on blockchain [25]. It uses the InterPlanetary File System (IPFS) to store protected content and supports content protection and traceability of copyright violations. It also provides specific security services, such as privacy-preserving user authentication, content encryption and watermarking, and a system that exploits conditional identity management to trace license violations. DRMChain proposes innovative ideas, but it cannot be considered a proof-of-concept. Its security is not proven for the whole system but only for single services. As a result, the ability to solve the classic problems of watermarking protocols is not guaranteed.

BMCProtector [26] uses a public blockchain, smart contracts, and the IPFS to protect music copyright. In particular, the smart contracts automatically implement the payment of royalties to copyright owners. However, BMCProtector adopts a Key Protection Center, in the form of TTP, to manage the keys that music owners and buyers use to encrypt/decrypt the distributed audio files. Furthermore, its security is not proven.

Another example of a complex copyright management system is described in [27]. It exploits different technologies to manage purchase transactions among buyers and sellers. Moreover, it does not employ WCAs. In particular, an improved ElGamal encryption algorithm is used to protect the buyers' and sellers' keys. A public blockchain is employed to record copyright and transaction data securely. The IPFS keeps watermarked contents, whereas smart contracts manage the purchase transactions and copyright information. The overall outcome is a system that can protect digital content. However, its internal structure is complex. So it is practically impossible to prove the security of the whole system, mainly with respect to the "unbinding problem" or "customer's rights problem".

Y-DWMS [28] is a DRM system that exploits blockchain and smart contracts to protect digital copyrights. In particular, smart contracts are exploited because they cannot be repudiated, whereas blockchain is employed because it cannot be tampered with. This way, Y-DWMS can implement the classic services of a DRM, such as watermark verification in the disclosed copies, authentication of the informers' reports, and traceability of infringements. In addition, Y-DWMS also implements specific services to reward informers, punish infringers, and recover losses suffered by copyright holders. However, Y-DWMS has not been proven secure since it is affected by some problems, mainly concerning account management and privacy.

SBBC (Secret Block-Based Blockchain) [29] is a system expressly developed to employ blockchain in the digital content trading environment. It consists of off-chain and on-chain network components. The off-chain components implement a preliminary authentication phase, after which users can trade digital content. Digital fingerprints are embedded into the traded contents to track illegal leaks. Furthermore, the traded contents are encrypted, and only the rightful users can access them, thus ensuring income for the legitimate content authors. The on-chain components implement the DRM services. The licensed users create secret blocks of their transactions and record them in the blockchain: through the verification and agreement of all blockchain participants, public blocks are added to the blockchain to finalize the transactions. SBBC represents a complete system for digital content trading based on blockchain. It specifically addresses the performance problems that affect systems based on blockchain. However, SBBC lacks a security analysis. So, it has yet to be proven secure against the classic problems of watermarking protocols [1,30].

FingerChain [31] is a copyrighted multi-owner media-sharing system based on a consortium blockchain network and asymmetric fingerprinting. The system enables authorized owners to share their media for a fee. It uses smart contracts and the "client-side embedding" watermarking protocol proposed in [7] to protect shared media. Its strengths are the absence of TTP to implement the protection protocol and the high owner-side efficiency due to client-side embedding. However, as reported above, the adopted watermarking scheme requires clients to remove noise sequences from the purchased contents and to generate secret fingerprints. These are security actions that are difficult for ordinary web users [1]. They make the system unsuited for the current Internet.

The DRM systems described above focus on two main objectives: how to prevent the use of copyrighted digital content without authorization or payment and how to improve blockchain performance. However, these systems cannot protect sellers when buyers legitimately purchase content and then illegally share it [32]. In particular, the DRM systems fail to disclose copyright infringements since they cannot prove the ownership of contents that are first downloaded and then tampered with by malicious users [30]. On the contrary, watermarking protocols just focus on the interaction schemes that make it possible to apply protection to digital content able to disclose copyright infringements [33]. As reported in Section 1, the deterrence action played by watermarking protocols appears to be the only way to support digital copyright protection in the current Internet.

3. Preliminary Considerations

The research activity conducted over the last two decades has made it possible to indicate a list of common usability and security requirements that watermarking protocols

must match in order to be suited for the current Internet [1]. The literature documents several examples of watermarking protocols that can match most of the requirements mentioned above. One example is the protocol described in [16], which is based on the "buyer-friendly" design approach [9]. Its peculiarity is the careful use of the TTP, whose role, due to the use of blockchain, can be confined to the initial phase of transactions between buyers and sellers. Such a solution makes the protocol provably secure without compromising its usability. However, TTPs are still considered some sort of "big brother", and watermarking protocols that exploit them are incompatible with privacy-oriented design specifics. Therefore, the only way to improve the protocol described in [16] is to eliminate the TTP.

A possible approach to achieving the above mentioned goal involves exploiting two main facilities characterizing blockchains.

The former is the capacity to record transactions in timestamped and cryptographically connected blocks saved in a public tamper-proof shared ledger so that anyone can verify their correctness. Such a facility makes TTPs superfluous for what concerns the need to establish mutual trust among unknown entities such as buyers and sellers.

The latter is represented by smart contracts, which can be used to automatically deploy business logic for copyright when the terms of an agreement are reached between buyers and sellers. Smart contracts can execute preset and unchangeable actions when given events are triggered under specific conditions without control from TTPs.

Both facilities can be used to implement the critical actions of watermarking protocols without resorting to TTPs, whereas the remaining actions can be implemented employing well-known primitives commonly used in previous and relevant examples of protocols [1]. Such an approach is just adopted to develop the watermarking protocol presented in this paper, which is primarily built on the primitives used in [16]. However, the protocol also exploits blockchain and smart contracts to avoid using a TTP and to match all the requirements reported in [1].

4. Basic Assumptions

The implementation of the watermarking protocol presented in this paper needs some basic security facilities: PKI, TLS (Transport Layer Security) secure communication, homomorphic cryptosystem with respect to watermark insertion [18], and blind and readable watermarking scheme able to embed fingerprinting codes [2,34].

The protocol assumes that digital contents and watermarks can be represented block-wise in the forms $X = \{x_1, x_2, \ldots x_l\}$ and $W = \{w_1, w_2, \ldots w_l\}$, respectively. The elements of X can be either the original host signal samples or the features of the host signal computed by transforms such as, for example, the discrete Fourier transform or the discrete cosine transform [34]. The elements of W are usually binary values representing fingerprinting anti-collusion codes [35].

The protocol also assumes that encryption and watermarking processes are represented by block-wise functions. As a consequence, the encryption E of a content $X = \{x_1, x_2 \ldots x_l\}$ under the key k can be calculated as [34]:

$$E_k(X) = E_k(x_1, x_2 \ldots x_l) = (E_k(x_1), E_k(x_2) \ldots E_k(x_l))$$

Likewise, the insertion of the watermark W into the content X can be expressed in the form:

$$X \oplus W = \{x_1 \oplus w_1, x_2 \oplus w_2, \ldots x_l \oplus w_l\} = \bar{X}$$

in which $\bar{X}$ is the watermarked content and the symbol $\oplus$ represents, in principle, an arbitrary function [1,36,37].

Finally, the protocol employs an encryption function $\mathbb{E}$, which is "homomorphic" with respect to the watermark insertion [18]. Therefore, the following formula can be used to insert any linear watermark directly into the encrypted domain [37]:

$$\mathbb{E}_k(X \oplus W) = \mathbb{E}_k(X) \oplus \mathbb{E}_k(W) = \mathbb{E}_k(\bar{X})$$

In particular, the function $\mathbb{E}$ has to be "probabilistic" and "semantically secure". This means that the knowledge of a ciphertext does not provide any useful information on the plaintext to an adversary having only a reasonably restricted computational power represented by polynomial resources [1]. Furthermore, $\mathbb{E}$ can be limited to homomorphic encryption schemes that support addition or multiplication since such operations are functionally complete sets over finite sets [1,18]. More in detail, if x is an element of X and w is an element of W, the multiplicatively homomorphic encryption schemes ensure that [38]

$$\mathbb{E}_k(x \cdot w) = \mathbb{E}_k(x) \cdot \mathbb{E}_k(w)$$

whereas the additively homomorphic encryption schemes ensure that [39,40]

$$\mathbb{E}_k(x + w) = \mathbb{E}_k(x) \cdot \mathbb{E}_k(w)$$

5. Proposed Protocol

The experiences described in [9,16] inspire the proposed watermarking protocol, whose main aim is to meet all the requirements reported in [1]. In this regard, the protocol exploits smart contracts and blockchain to generate the tokens used in the protection transactions and to lock the final purchase licenses in a public ledger. In particular, two specific smart contracts are executed during each transaction. The former generates a first "secret" block, which includes the tokens needed to start the purchase transaction and to apply the protection to the chosen content. In this phase, the block is not published by the blockchain but is kept secret and shared only by the buyer and the content provider involved in the transaction. The latter generates a final "public" block, representing the protected content's purchase license. It is published in the blockchain and built on the tokens in the secret block created by the former smart contract. The overall result is a protocol that works without TTP.

Even though the proposed protocol can run without the centralized control of a TTP, it still needs a "judge" to implement the "identification and arbitration protocol". This protocol makes it possible to determine the identity of illegal distributors of copies of copyrighted digital content [9,16]. The judge is a TTP, which cannot be considered a WCA since it plays a limited role in the protocol: it does not participate in the watermark embedding into the digital content distributed on the Internet [16,17].

The proposed protocol can be briefly described as follows: (1) the seller or content provider ($\mathcal{CP}$) releases only encrypted and watermarked digital contents; (2) the buyer ($\mathcal{B}$) decrypts the received content and thus obtains its copyrighted version; (3) the protection transaction is based on two smart contracts automatically executed within a blockchain ($\mathcal{BC}$); the contracts take charge of generating and managing all the security tokens needed to implement the transaction without resorting to a TTP; (4) the judge ($\mathcal{J}$) determines if a buyer has illegally distributed copyrighted contents.

The protocol consists of the *protection protocol* and the *identification and arbitration protocol*. Table 1 defines the symbols used to describe the protocols.

Table 1. Symbols used in the protocol.

Symbols	Meaning
$\mathcal{B}$	buyer
$\mathcal{CP}$	content provider or seller
$\mathcal{BC}$	blockchain
$\mathcal{J}$	judge
X	digital content purchased by $\mathcal{B}$
X_d	information used by $\mathcal{CP}$ to unambiguously identify X
T_X	timestamp referred to the transaction by which $\mathcal{B}$ buys X
B_{id}	information used to identify $\mathcal{B}$
N	nonce used to mark the watermarking transaction
W	watermark
$W_{Ent.}$	part of the watermark W generated by the entity $Ent.$
$\bar{X}$	watermarked X
$pk_{Ent.}$	public key of the entity $Ent.$
$sk_{Ent.}$	secret key of the entity $Ent.$
pk^X	one-time public key used to watermark X
sk^X	one-time secret key used to watermark X
$E_{key}(\dots)$	token encrypted with the key key
$\mathbb{A}_{sm}(\dots)$	activation of the smart contract sm upon receipt of messages
$\mathbb{G}_{sm}(\dots)$	creation of a block from the smart contract sm
$\mathbb{END}_{sm}(\dots)$	end of the smart contract sm
$SBlock$	secret block generated by the blockchain
$PBlock$	public block generated by the blockchain
$\mathbb{E}_{key}(\dots)$	token encrypted with the key key and using a privacy homomorphic cryptosystem with respect to the watermark insertion
$\mathbb{D}_{key}(\dots)$	decryption function inverse of the function $\mathbb{E}_{key}(\dots)$

5.1. Protection Protocol

As shown in Table 2 and in Figure 1, after choosing the content X, $\mathcal{B}$ starts the protocol by sending the purchase request to $\mathcal{CP}$ in the message m_1.

$\mathcal{CP}$ receives the request and sends the message m_2 to $\mathcal{B}$ and $\mathcal{BC}$. m_2 contains X_d and T_X: the former is a string that unambiguously identifies the requested content X, whereas the latter is a timestamp that refers to the ongoing transaction.

When $\mathcal{B}$ receives m_2, it can send the message m_3 to $\mathcal{BC}$. m_3 includes X_d and T_X. Moreover, it also contains B_{id}, which is the token used to recognize $\mathcal{B}$ uniquely. It can be, for instance, a personal digital certificate, an anonymous digital certificate, or a credit card. The choice is made by $\mathcal{B}$, who thus decides which "negotiation mechanism", defined within the concept of "multilateral security" [41], best suits the current transaction.

Once m_2 and m_3 are received, the smart contract sm_s can run in the blockchain $\mathcal{BC}$. It carries out two preliminary checks: it verifies (1) if the tokens included in the received messages coincide and (2) if X_d and T_X were never used in previous purchase transactions, that is, if these tokens are present or not in the blocks published by $\mathcal{BC}$. If the checks are passed, sm_s can continue execution and generate the security tokens needed for the protocol. In particular, it generates: (1) a one-time public and private key pair, pk^X and sk^X, to be used only in the current transaction; (2) a "nonce" N, which is represented by a binary string and is encrypted with pk^X and a "privacy homomorphic" cryptosystem [18] with respect to the watermark insertion. The resulting token $\mathbb{E}_{pk^X}(N)$ is then used to generate the watermark to be inserted into the content X.

sm_s generates the "secret" block, named $SBlock$, that contains a list of tokens encrypted with $k_{\mathcal{BC}}$, which is the secret key of $\mathcal{BC}$. $SBlock$ contains the tokens B_{id}, X_d, T_X, pk^X, sk^X, N, and $\mathbb{E}_{pk^X}(N)$ encrypted with $k_{\mathcal{BC}}$. This enables only $\mathcal{BC}$ to access the content of the

"secret" block. Then, $SBlock$ is sent to $\mathcal{CP}$ and $\mathcal{B}$ in the messages m_4 and m_5, respectively. $\mathcal{BC}$ uses $SBlock$ to run the subsequent smart contract sm_p, which verifies the outcome of the ongoing transaction and, if all data match, publishes a new node in the blockchain $\mathcal{BC}$. Furthermore, m_4 communicates pk^X and $\mathbb{E}_{pk^X}(N)$ to $\mathcal{CP}$, thus enabling the watermark embedding into X.

Table 2. Protection protocol.

Entities and Interactions		Actions and Data
$\mathcal{B}$	:	browses the web site of $\mathcal{CP}$ and picks the content X
$\mathcal{B} \xrightarrow{m_1} \mathcal{CP}$	:	$m_1 = \{\text{request for } X\}$
$\mathcal{CP} \xrightarrow{m_2} \mathcal{B}, \mathcal{BC}$	:	$m_2 = \{X_d, T_X\}$
$\mathcal{B} \xrightarrow{m_3} \mathcal{BC}$	:	$m_3 = \{B_{id}, X_d, T_X\}$
$\mathcal{BC}$	:	$\mathbb{A}_{sm_s}(m_2, m_3)$
$\mathcal{BC}$	:	$\mathbb{G}_{sm_s}(SBlock = [E_{k_{\mathcal{BC}}}(B_{id}, X_d, T_X, pk^X, sk^X, N, \mathbb{E}_{pk^X}(N))])$
$\mathcal{BC} \xrightarrow{m_4} \mathcal{CP}$	:	$m_4 = \{X_d, T_X, pk^X, \mathbb{E}_{pk^X}(N), SBlock\}$
$\mathcal{BC} \xrightarrow{m_5} \mathcal{B}$	:	$m_5 = \{X_d, T_X, SBlock\}$
$\mathcal{BC}$	:	$\mathbb{END}_{sm_s}$
$\mathcal{CP}$	:	generates $W_{\mathcal{CP}}, \mathbb{E}_{pk^X}(W_{\mathcal{CP}}), \mathbb{E}_{pk^X}(X)$
$\mathcal{CP}$	:	generates $\mathbb{E}_{pk^X}(W) = \mathbb{E}_{pk^X}(W_{\mathcal{CP}}) \Vert \mathbb{E}_{pk^X}(N)$
$\mathcal{CP}$	:	generates $\overline{\mathbb{E}_{pk^X}(X)} = \mathbb{E}_{pk^X}(X) \oplus \mathbb{E}_{pk^X}(W)$
$\mathcal{CP} \xrightarrow{m_6} \mathcal{B}$	:	$m_6 = \{\overline{\mathbb{E}_{pk^X}(X)}\}$
$\mathcal{CP} \xrightarrow{m_7} \mathcal{BC}$	:	$m_7 = \{X_d, T_X, pk^X, \mathbb{E}_{pk^X}(N), SBlock\}$
$\mathcal{B} \xrightarrow{m_8} \mathcal{BC}$	:	$m_8 = \{X_d, T_X, B_{id}, SBlock\}$
$\mathcal{BC}$	:	$\mathbb{A}_{sm_p}(m_7, m_8)$
$\mathcal{BC}$	:	makes the payment and notifies $\mathcal{CP}$
$\mathcal{BC}$	:	$\mathbb{G}_{sm_p}(PBlock = [X_d, T_X, SBlock])$
$\mathcal{BC}$	:	publishes $PBlock$ in the blockchain
$\mathcal{BC} \xrightarrow{m_9} \mathcal{B}$	:	$m_9 = \{sk^X\}$
$\mathcal{BC}$	:	$\mathbb{END}_{sm_p}$
$\mathcal{CP}$	:	inserts a new entry in its databases, including X_d, T_X, pk^X, $\mathbb{E}_{pk^X}(N)$, and $SBlock$ whose search key is $W_{\mathcal{CP}}$
$\mathcal{B}$	:	$\bar{X} = \mathbb{D}_{sk^X}(\overline{\mathbb{E}_{pk^X}(X)})$

$\mathcal{CP}$ receives m_4 and generates the watermark $W_{\mathcal{CP}}$ as a fingerprinting binary code composed of an anti-collusion code [2] and an error-correcting code needed to cope with errors that may occur when watermarks are extracted from contents. $W_{\mathcal{CP}}$ and X are then encrypted by $\mathcal{CP}$ with pk^X applied to the same homomorphic cryptosystem used by $\mathcal{BC}$ to encrypt N, thus generating $\mathbb{E}_{pk^X}(W_{\mathcal{CP}})$ and $\mathbb{E}_{pk^X}(X)$.

Then, by the formulas shown in Section 4, $\mathcal{CP}$ can generate the encrypted watermark $\mathbb{E}_{pk^X}(W)$ by simply concatenating $\mathbb{E}_{pk^X}(W_{\mathcal{CP}})$ and $\mathbb{E}_{pk^X}(N)$:

$$\mathbb{E}_{pk^X}(W) = \mathbb{E}_{pk^X}(W_{\mathcal{CP}}) \Vert \mathbb{E}_{pk^X}(N) = \mathbb{E}_{pk^X}(W_{\mathcal{CP}} \Vert N) \tag{1}$$

Since the protocol assumes a privacy homomorphic encryption scheme with respect to watermark insertion [37], $\mathcal{CP}$ can embed the encrypted watermark $\mathbb{E}_{pk^X}(W)$ directly into the encrypted content $\mathbb{E}_{pk^X}(X)$:

$$\overline{\mathbb{E}_{pk^X}(X)} = \mathbb{E}_{pk^X}(\bar{X}) = \mathbb{E}_{pk^X}(X \oplus W) = \mathbb{E}_{pk^X}(X) \oplus \mathbb{E}_{pk^X}(W) \tag{2}$$

thus obtaining $\overline{\mathbb{E}_{pk^X}(X)}$, which is the watermarked and encrypted content to be sent to $\mathcal{B}$ in the message m_6.

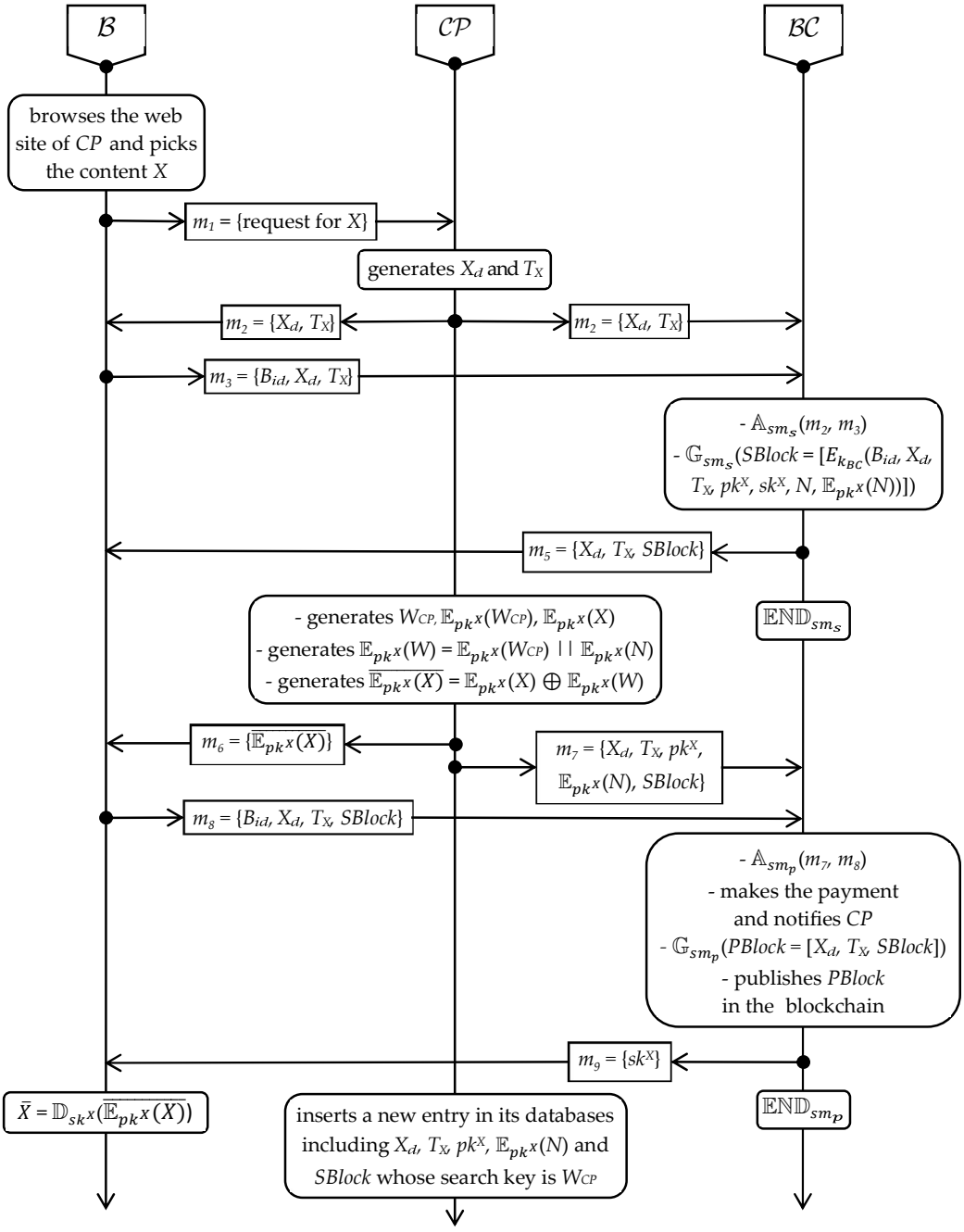

Figure 1. Protection protocol.

After sending $\overline{\mathbb{E}_{pk^X}(X)}$ to $\mathcal{B}$, $\mathcal{CP}$ and $\mathcal{B}$ can send the messages m_7 and m_8 to $\mathcal{BC}$, which can automatically run the smart contract sm_p. In fact, both messages return to $\mathcal{BC}$ the tokens previously received in m_4 and m_5. This enables the smart contract to verify whether all the received data matches. If so, sm_p can make the payment and generate a new block in $\mathcal{BC}$, which publishes the tokens uniquely identifying the current transaction: X_d, T_X, and $SBlock$. sm_p ends by sending the message m_9 to $\mathcal{B}$. m_9 contains the secret key sk^X, which enables $\mathcal{B}$ to decrypt $\overline{\mathbb{E}_{pk^X}(X)}$, thus obtaining the final copyrighted content, as indicated by the following expressions:

$$\overline{\mathbb{E}_{pk^X}(X)} = \mathbb{E}_{pk^X}(\bar{X}), \qquad \bar{X} = \mathbb{D}_{sk^X}(\overline{\mathbb{E}_{pk^X}(X)}) \tag{3}$$

The protocol ends with $\mathcal{CP}$ saving a new entry in its databases. The entry contains the security tokens that characterize the concluded transaction: X_d, T_X, pk^X, $\mathbb{E}_{pk^X}(N)$, and $SBlock$. It can be retrieved by $\mathcal{CP}$ by using $W_{\mathcal{CP}}$ as a search key. In fact, the saved tokens make it possible to recognize $\mathcal{B}$ as the legitimate owner of the copyrighted content $\bar{X}$ released by $\mathcal{CP}$ through a transaction whose data are recorded in a block of $\mathcal{BC}$.

5.2. Identification and Arbitration Protocol

The protocol is shown in Table 3 and is employed by $\mathcal{CP}$ to recognize, with statistical certainty, who was the legitimate copyright owner of a pirated copy of $\bar{X}$, denoted as X', illegally distributed on the Internet [9].

The protocol starts with extracting the watermark W' from X'. Since W' can be expressed as $W'_{\mathcal{CP}} \| N'$, $\mathcal{CP}$ can use $W'_{\mathcal{CP}}$ to search its databases for a match. If a possible match is found [3], $\mathcal{CP}$ can access the corresponding entry and obtain the tokens X_d, T_X, pk^X, $\mathbb{E}_{pk^X}(N)$, and $SBlock$ previously saved during the purchase transaction of $\bar{X}$. These tokens, together with N', are included in the message m_1 and sent by $\mathcal{CP}$ to $\mathcal{J}$. $\mathcal{J}$ then forwards the received tokens, except N', to $\mathcal{BC}$ in the message m_2.

When m_2 is received, the smart contract sm_v can be automatically executed. It searches the blockchain $\mathcal{BC}$ for a block publishing X_d, T_X, pk^X, $\mathbb{E}_{pk^X}(N)$, and $SBlock$. If a block is found, sm_v decrypts $SBlock$ and compares the retrieved tokens with those received in m_2. If all the tokens match, $\mathcal{BC}$ sends B_{id} and N to $\mathcal{J}$ in the message m_3. This condition ends sm_v.

Table 3. Identification and arbitration protocol.

Entities and Interactions		Actions and Data
$\mathcal{CP}$	:	finds X' in the market and extracts $W' = W'_{\mathcal{CP}} \| N'$
$\mathcal{CP}$	:	searches its databases for a possible match on $W'_{\mathcal{CP}}$
$\mathcal{CP} \xrightarrow{m_1} \mathcal{J}$	:	$m_1 = \{N', X_d, T_X, pk^X, \mathbb{E}_{pk^X}(N), SBlock\}$
$\mathcal{J} \xrightarrow{m_2} \mathcal{BC}$	:	$m_2 = \{X_d, T_X, pk^X, \mathbb{E}_{pk^X}(N), SBlock\}$
$\mathcal{BC}$	:	$\mathbb{A}_{sm_v}(m_2)$
$\mathcal{BC}$	:	retrieves $PBlock = [X_d, T_X, SBlock]$
$\mathcal{BC}$	:	decrypts $SBlock$ and obtains $B_{id}, X_d, T_X, pk^X, sk^X, N, \mathbb{E}_{pk^X}(N)$
$\mathcal{BC} \xrightarrow{m_3} \mathcal{J}$	:	$m_3 = \{B_{id}, N\}$
$\mathcal{BC}$	:	$\mathbb{END}_{sm_v}$
$\mathcal{J}$	:	compares N' with N and adjudicates

Upon receiving m_3, $\mathcal{J}$ compares N' with N. If $N' == N$, $\mathcal{B}$ is found guilty of being a traitor, and the identity associated with B_{id} is made known. Otherwise, the protocol ends with no result.

6. Security Analysis

As indicated in Section 4, the following assumptions are the basis of the analysis conducted on the proposed protocol. Firstly, all communications among the web entities involved in the protocol are protected by the TLS protocol to avoid man-in-the-middle attacks in point-to-point communications. Furthermore, digital contents are protected by adopting a watermark insertion technique that can resist the most common manipulations. This is a reasonable assumption since examples of such techniques are well-documented in the literature [34,42,43]. Finally, the adopted encryption scheme is privacy homomorphic with respect to watermark insertion. It is "probabilistic" and "semantically secure", meaning that a malicious user with polynomial computational resources cannot obtain useful information from a ciphertext [18]. In this regard, the analysis can be restricted to the asymmetric homomorphic encryption schemes that support addition operations [1,39,40]

since such schemes can be adopted in conjunction with both "spread spectrum" (SS) watermarking techniques [34] and data hiding schemes defined as "informed embedding", such as watermark insertion schemes based on "quantization index modulation" (QIM) [37,42].

For the present analysis, the watermarking protocol can be synthetically described as follows: $\mathcal{CP}$ sells the copyrighted digital content $\bar{X}$ to $\mathcal{B}$; $\mathcal{BC}$ behaves as a ledger able to publish the information identifying the purchase transaction of $\bar{X}$; sm_s and sm_p are the smart contracts that are automatically executed within $\mathcal{BC}$ to generate and manage the tokens needed to conduct the purchase transactions; $\mathcal{J}$ determines whether $\mathcal{B}$ has infringed the $\mathcal{CP}$'s copyright by distributing pirated copies of $\bar{X}$. Furthermore, the entities participating in the protocol behave as follows:

- When uncorrupted, buyers and content providers do not infringe copyright by distributing pirated copies.
- $\mathcal{J}$ cannot be corrupted since it is implemented by a TTP.
- $\mathcal{BC}$ can be considered an "honest-but-curious" entity [44]. This means that $\mathcal{BC}$ could obtain information about purchase transactions, but it cannot exploit it to collude with $\mathcal{B}$ or $\mathcal{CP}$ since it is forced to behave according to the protocol rules. This is a realistic assumption since the protocol forces $\mathcal{BC}$ only to run two smart contracts, sm_s and sm_p, whose code, once approved, is not allowed to be modified during the life of the blockchain [13–15].
- $\mathcal{CP}$ and $\mathcal{B}$ can be corrupted only "statically", i.e., deciding which of the two entities is corrupt is made before running the protocol and can no longer be changed [44].

As a result of what is reported above, the protocol enables $\mathcal{B}$, in the absence of corrupt entities, to obtain the personalized copyrighted content $\bar{X}$ from $\mathcal{CP}$ at the end of the purchase transaction. It also makes it possible to trace pirated copies of $\bar{X}$ found on the web back to $\mathcal{B}$ and the corresponding purchase transaction.

In the presence of corrupt entities, the protocol can always disclose malicious behaviors. In particular, the copyrighted contents released by a corrupt $\mathcal{CP}$ turn out to be incorrectly protected and cannot be traced back to the corresponding buyers. Such a condition just damages $\mathcal{CP}$, since traitors cannot be identified. Likewise, the corrupt participation of $\mathcal{B}$ in the protocol is always disclosed, which causes the protocol to abort, thus preventing $\mathcal{CP}$ from distributing any content.

In order to prove the correctness of the protocol, a brief analysis based on simple security considerations is reported in the next Section.

Analysis

The security of the protocol is tied to the execution of two smart contracts, namely sm_s and sm_p (see Table 2 and Figure 1).

sm_s receives tokens referring to the content to purchase (X_d), the current transaction (T_X), and the buyer's identity (B_{id}) from two distinct sources: $\mathcal{B}$ and $\mathcal{CP}$. As reported in Section 5.1, only if the received tokens coincide and were not used in previous purchase transactions, sm_s continues execution. As a result, the $SBlock$ is created. It securely encapsulates all the tokens needed to protect the content to purchase, such as the one-time encryption keys and the watermark to be embedded into the content.

After the watermark insertion, the tokens included in $SBlock$ must be confirmed by $\mathcal{B}$ and $\mathcal{CP}$ in messages m_7 and m_8 to activate sm_s. Only if the received tokens coincide, sm_p securely encapsulates the tokens in $PBlock$ and definitively publishes this block in $\mathcal{BC}$, thus validating the purchase transaction. In other words, the publication of $PBlock$ in $\mathcal{BC}$ guarantees that $\mathcal{B}$ and $\mathcal{CP}$ have both confirmed the tokens published in the block and that such tokens unambiguously identify the purchase transaction.

Suppose that $\mathcal{B}$ is corrupt. $\mathcal{B}$ provides only the single token B_{id}, which is assumed to be valid since the problems of false identity are out of the scope of this paper. The other tokens managed by $\mathcal{B}$ are only received and re-sent. They are created by the other entities involved in the protocol, namely $\mathcal{CP}$ and $\mathcal{BC}$. Therefore, if $\mathcal{B}$ tries to alter the received tokens, the smart contracts can disclose the attempt since $\mathcal{CP}$ and $\mathcal{BC}$ are in a position

to disclaim the altered tokens. As a consequence, smart contracts can abort the purchase transaction without releasing any content, according to what is reported in Section 6.

Suppose that $\mathcal{CP}$ is corrupt. Unlike $\mathcal{B}$, $\mathcal{CP}$ plays a more relevant role in the purchase transaction since it generates two tokens, namely X_d and T_X, and takes charge of embedding the watermark into the content to protect. Therefore, $\mathcal{CP}$ can behave maliciously. However, based on what is reported above, two main constraints are imposed by the protocol due to the smart contracts:

1. $\mathcal{CP}$ cannot reuse tokens employed in previous transactions;
2. $\mathcal{CP}$ cannot alter or change the watermark generated by sm_s by reusing, for example, watermarks employed in previous transactions.

In the former case, the reuse of tokens is disclosed by sm_s, which carries out a specific check.

In the latter case, the content X ends up being protected by a watermark different from that encapsulated in $PBlock$ and published in $\mathcal{BC}$. Therefore, the protected content $\bar{X}$ cannot be correctly tied to any buyer, thus preventing anybody from being adjudicated as a traitor. As reported in Section 6, the malicious behavior of $\mathcal{CP}$ just turns against itself.

The smart contracts sm_s and sm_p can secure the protocol without resorting to a TTP. This condition is mainly due to several specific characteristics of the smart contracts. Firstly, they can autonomously execute preset and unchangeable actions when given events are triggered under specific conditions. Therefore, their behavior cannot be modified. Furthermore, they can carry out specific checks on the tokens exchanged among the entities involved in the protocol by exploiting the public ledger implemented by $\mathcal{BC}$. Finally, they can use the secret block $SBlock$ and the public block $PBlock$ to prevent $\mathcal{B}$ and $\mathcal{CP}$ from maliciously altering or reusing tokens.

7. Implementation and Performance

This Section reports on a few hints about the implementation and performance of the proposed protocol. In particular, the implementation is based on previous experiences conducted in [16,17], and follows the scheme shown in Figure 2. The entities involved in the protocol run as C++ separate programs on Linux OS. They use the OpenSSL standard socket library to communicate and implement the encryption/decryption and watermark insertion procedures by exploiting the NTL and GNU Multi Precision Arithmetic libraries.

The C++ programs run on distinct PCs connected by a GBit Ethernet. Each PC has a CPU Intel(R) Core(TM) i5-12450HX with frequency up to 4.4GHz, 16GB of RAM, and an SS disk of 512GB.

More in detail, watermark insertion is carried out by running two well-known algorithms: the SS algorithm described in [34] and the QIM algorithm presented in [42]. Both are adapted to the homomorphic cryptosystem proposed by Paillier [40] according to what is documented in [36,45–47]. The insertions take advantage of the optimizations reported in [37,48], which can improve the watermark insertion carried out directly into the encrypted domain.

The watermark insertion based on the SS algorithm can be carried out by applying the following formula:

$$\bar{x} = x + \alpha(2b - 1)s$$

where x is a host signal feature obtained by calculating the cosine discrete transform, $\bar{x}$ is the corresponding watermarked feature, $b \in \{0, 1\}$ is the bit to embed, s is the component of a spreading sequence, and α is a scaling factor that controls the watermark's strength. The corresponding watermark insertion directly into the encrypted domain is carried out according to the expression [36,37,46]:

$$E[\bar{x}] = E[x] \cdot E[b]^{2\alpha s} \cdot E[\alpha s]^{-1} \quad (4)$$

The watermark insertion based on the QIM algorithm [42] is carried out according to the following expression [37]:

$$\bar{x} = f(x) + b\Delta(x)$$

where x, $\bar{x}$, and b maintain the signification reported above, while $f(x)$ and $\Delta(x)$ denote a function of the original signal features and a signal-dependent quantization step, respectively [1,42]. Therefore, by directly operating in the encrypted domain, the watermark insertion can be obtained by applying the following expression [37,46,47]:

$$E[\bar{x}] = E[f(x)] \cdot E[b]^{\Delta(x)} \quad (5)$$

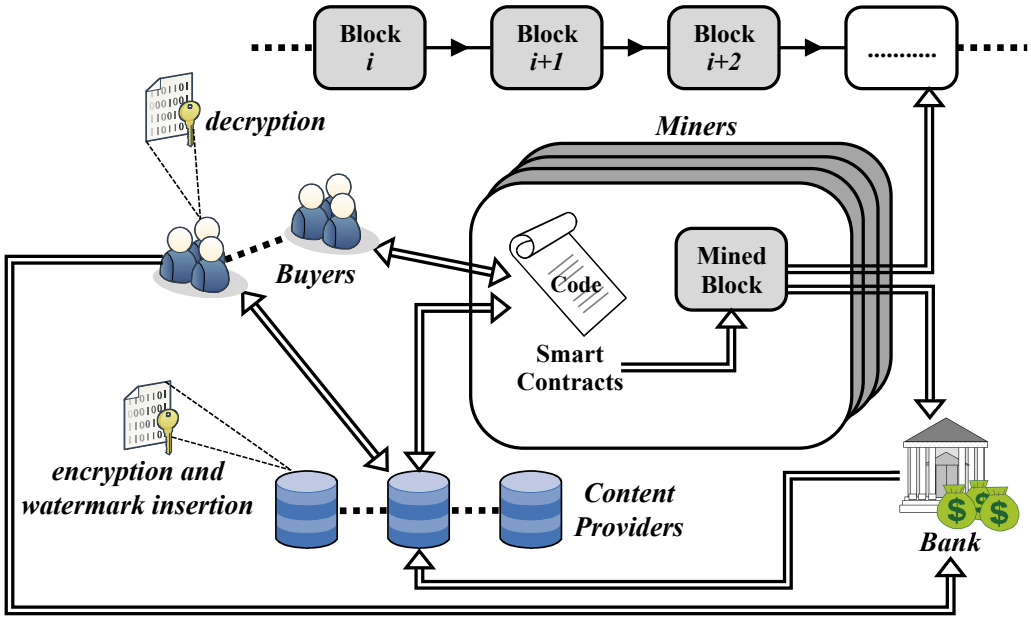

Figure 2. The functional scheme of the proposed watermarking protocol.

In the first implementations documented in [16,17], the adopted blockchain is classified as "public", with a fully decentralized architecture, and is based on the "proof of work" (PoW) consensus algorithm [10–12]. Its nodes are implemented in Ethereum [49], whereas the smart contracts are coded in Solidity [50]. In particular, the introduction of cryptographic primitives within smart contracts has made it necessary to start a specific experimental phase that has not yet been completed. So, the results reported in the following refer to previous implementations that simulate the execution of the necessary cryptographic primitives on a specific server external to the smart contracts.

The watermark embedding algorithms reported above have been used on two sets of images: images of 512×512 pixels and images of 1024×1024 pixels. A 128-bit fingerprint has been embedded in each image, whereas the cryptosystem employed in the conducted tests uses keys with 1024 bits. In these hypotheses, a watermark insertion in 512×512 images takes about 29–41 s depending on the watermarked image and on the watermarking insertion algorithm. The same insertion in 1024×1024 images takes about 107–145 s.

Finally, the computational cost concerning the blockchain can be summarized as follows: 140,000 gas units are needed to complete a single purchase transaction, whereas 8,000,000 gas units represent the limit of the computational cost per block. Therefore, the number of transactions per block is about 57–60. Since the time needed to mine a block of transactions in the blockchain is about 18–24 s, the rate at which the blockchain can commit the purchase transactions is about 3–4 per second. However, the performance of the blockchain is complex to evaluate since it depends on several factors, such as the node implementation, the consensus algorithm, and the number of nodes that are averagely involved in the mining process [51,52]. In this regard, it is worth noting that the PoW

consensus algorithm is particularly suited to a public and decentralized blockchain [51–53]. However, it is characterized by low performance due to the time needed for propagating, processing, and validating the purchase transactions [54]. In fact, the higher the number of nodes participating in the blockchain is, the more limiting power consumption and block generation rate become. Other consensus algorithms, such as the "practical byzantine fault tolerance" (PBFT) algorithm, can solve PoW performance problems. However, they are commonly employed for private blockchains [29]. Therefore, a possible solution consists of experimenting with specific and innovative consensus algorithms, such as the "weighted authentication byzantine fault tolerance" (WBFT) algorithm, which can efficiently validate transactions by involving both authenticated and non-authenticated miners, thus turning a public blockchain into a semi-public blockchain [29]. Such a solution promises good performance, and it appears to be the only way to make the use of blockchain in digital copyright protection systems possible.

8. Conclusions

TTPs are generally considered a problem for watermarking protocols since they can give rise to collusion or conspiracy problems or behave as some sort of "big brother". However, when they are not employed, watermarking protocols end up forcing buyers to carry out complex actions to participate in the transactions needed to purchase digital content distributed on the web, thus strongly reducing their usability. The protocol proposed in this paper can avoid using TTPs without limiting usability and security. It can exploit smart contracts executed within a blockchain to manage the protection process characterizing purchase transactions autonomously. Such a process develops in two simple phases. In the former, a smart contract generates the tokens needed to apply the protection to the chosen content and keeps them secret in a block that is not published by the blockchain but shared only by the buyer and the content provider. In the latter, a smart contract, with the collaboration of the buyer and content provider, confirms the tokens encapsulated in the previous secret block and includes them in a final public block representing the purchase license for the protected content published in the blockchain. Such use of smart contracts represents a new, relevant experience in the field of digital copyright protection since it finally enables the proposed protocol to meet all the requirements that make it suited to the current web context.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

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

References

1. Frattolillo, F. Watermarking Protocols: A Short Guide for Beginners. *Future Internet* **2023**, *15*, 163. [CrossRef]
2. Liu, K.J.R.; Trappe, W.; Wang, Z.J.; Wu, M.; Zhao, H. *Multimedia Fingerprinting Forensics for Traitor Tracing*; Hindawi Publishing Corporation: New York, NY, USA, 2005.
3. Lei, C.L.; Yu, P.L.; Tsai, P.L.; Chan, M.H. An Efficient and Anonymous Buyer-Seller Watermarking Protocol. *IEEE Trans. Image Process.* **2004**, *13*, 1618–1626. [CrossRef] [PubMed]
4. Fan, C.I.; Chen, M.T.; Sun, W.Z. Buyer-Seller Watermarking Protocols with Off-line Trusted Parties. In Proceedings of the IEEE International Conference on Multimedia and Ubiquitous Engineering, Seul, Republic of Korea, 26–28 April 2007; pp. 1035–1040.
5. Rial, A.; Deng, M.; Bianchi, T.; Piva, A.; Preneel, B. A Provably Secure Anonymous Buyer—Seller Watermarking Protocol. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 920–931. [CrossRef]
6. Rial, A.; Balasch, J.; Preneel, B. A Privacy-Preserving Buyer–Seller Watermarking Protocol Based on Priced Oblivious Transfer. *IEEE Trans. Inf. Forensics Secur.* **2011**, *6*, 202–212. [CrossRef]
7. Bianchi, T.; Piva, A. TTP-free asymmetric fingerprinting based on client side embedding. *IEEE Trans. Inf. Forensics Secur.* **2014**, *9*, 1557–1568. [CrossRef]
8. Bianchi, T.; Piva, A.; Shullani, D. Anticollusion solutions for asymmetric fingerprinting protocols based on client side embedding. *Eurasip J. Inf. Secur.* **2015**, *2015*, 6. [CrossRef]
9. Frattolillo, F. Watermarking protocols: An excursus to motivate a new approach. *Int. J. Inf. Secur.* **2018**, *17*, 587–601. [CrossRef]

10. Agrawal, K.; Aggarwal, M.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Sharma, R. An Extensive Blockchain Based Applications Survey: Tools, Frameworks, Opportunities, Challenges and Solutions. *IEEE Access* **2022**, *10*, 116858–116906. [CrossRef]
11. Cao, B.; Wang, Z.; Zhang, L.; Feng, D.; Peng, M.; Zhang, L.; Han, Z. Blockchain Systems, Technologies, and Applications: A Methodology Perspective. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 353–385. [CrossRef]
12. Chen, X.; He, S.; Sun, L.; Zheng, Y.; Wu, C.Q. A Survey of Consortium Blockchain and Its Applications. *Cryptography* **2024**, *8*, 12. [CrossRef]
13. Zou, W.; Lo, D.; Kochhar, P.S.; Le, X.B.D.; Xia, X.; Feng, Y.; Chen, Z.; Xu, B. Smart Contract Development: Challenges and Opportunities. *IEEE Trans. Softw. Eng.* **2021**, *47*, 2084–2106. [CrossRef]
14. Wu, C.; Xiong, J.; Xiong, H.; Zhao, Y.; Yi, W. A Review on Recent Progress of Smart Contract in Blockchain. *IEEE Access* **2022**, *10*, 50839–50863. [CrossRef]
15. Tanwar, S.; Gupta, N.; Tomar, J.S.; Kumar, K.; Treľová, S.; Ivanochko, I. A Review on Blockchain Smart Contract Applications. In Proceedings of the International Conference on Advances in Communication Technology and Computer Engineering, Bolton, UK, 24–25 February 2023; Iwendi, C., Boulouard, Z., Kryvinska, N., Eds.; Springer: Cham, Swizerland, 2023; pp. 175–187.
16. Frattolillo, F. A Watermarking Protocol Based on Blockchain. *Appl. Sci.* **2020**, *10*, 7746. [CrossRef]
17. Frattolillo, F. Blockchain and Cloud to Overcome the Problems of Buyer and Seller Watermarking Protocols. *Appl. Sci.* **2021**, *11*, 12028. [CrossRef]
18. Acar, A.; Aksu, H.; Uluagac, A.S.; Conti, M. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. *ACM Comput. Surv.* **2019**, *51*, 79. [CrossRef]
19. Zhang, J.; Kou, W.; Fan, K. Secure buyer-seller watermarking protocol. *IEE Proc. Inf. Secur.* **2006**, *153*, 15–18. [CrossRef]
20. Hu, Y.; Zhang, J. A Secure and Efficient Buyer-Seller Watermarking Protocol. *J. Multimed.* **2009**, *4*, 161–168. [CrossRef]
21. Katzenbeisser, S.; Lemma, A.; Celik, M.U.; van der Veen, M.; Maas, M. A Buyer—Seller Watermarking Protocol Based on Secure Embedding. *IEEE Trans. Inf. Forensics Secur.* **2008**, *3*, 783–786. [CrossRef]
22. Tardos, G. Optimal probabilistic fingerprint codes. In Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 9–11 June 2003; pp. 116–125.
23. Frattolillo, F.; Landolfi, F. Designing a DRM System. In Proceedings of the 4th International Conference on Information Assurance and Security, Naples, Italy, 8–10 September 2008; pp. 221–226.
24. Zhaofeng, M.; Weihua, H.; Hongmin, G. A new blockchain-based trusted DRM scheme for built-in content protection. *Eurasip J. Image Video Process.* **2018**, *2018*, 91. [CrossRef]
25. Ma, Z.; Jiang, M.; Gao, H.; Wang, Z. Blockchain for digital rights management. *Future Genereration Comput. Syst.* **2018**, *89*, 746–764. [CrossRef]
26. Zhao, S.; O'Mahony, D. BMCProtector: A Blockchain and Smart Contract Based Application for Music Copyright Protection. In Proceedings of the International Conference on Blockchain Technology and Application, Xi'an, China, 10–12 December 2018; pp. 1–5.
27. Peng, W.; Yi, L.; Fang, L.; XinHua, D.; Ping, C. Secure and Traceable Copyright Management System Based on Blockchain. In Proceedings of the IEEE 5th International Conference on Computer and Communications, Chengdu, China, 6–9 December 2019; pp. 1243–1247.
28. Zhao, B.; Fang, L.; Zhang, H.; Ge, C.; Meng, W.; Liu, L.; Su, C. Y-DWMS: A Digital Watermark Management System Based on Smart Contracts. *Sensors* **2019**, *19*, 3091. [CrossRef] [PubMed]
29. Heo, G.; Yang, D.; Doh, I.; Chae, K. Efficient and Secure Blockchain System for Digital Content Trading. *IEEE Access* **2021**, *9*, 77438–77450. [CrossRef]
30. Qureshi, A.; Megías Jiménez, D. Blockchain-Based Multimedia Content Protection: Review and Open Challenges. *Appl. Sci.* **2021**, *11*, 1. CrossRef]
31. Xiao, X.; Zhang, Y.; Zhu, Y.; Hu, P.; Cao, X. FingerChain: Copyrighted Multi-Owner Media Sharing by Introducing Asymmetric Fingerprinting Into Blockchain. *IEEE Trans. Netw. Service Manag.* **2023**, *20*, 2869–2885. [CrossRef]
32. Zhang, Z.; Pei, Q.; Ma, J.; Yang, L. Security and Trust in Digital Rights Management: A Survey. *Int. J. Netw. Secur.* **2009**, *9*, 247–263.
33. Islam, M.M.; In, H.P. Decentralized Global Copyright System Based on Consortium Blockchain with Proof of Authority. *IEEE Access* **2023**, *11*, 43101–43115. [CrossRef]
34. Cox, I.; Miller, M.; Bloom, J.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*; Morgan Kaufmann: Burlington, MA, USA, 2007.
35. Zhao, H.V.; Liu, K.J.R. Traitor-Within-Traitor Behavior Forensics: Strategy and Risk Minimization. *IEEE Trans. Inf. Forensics Secur.* **2006**, *1*, 440–456. [CrossRef]
36. Kuribayashi, M. On the Implementation of Spread Spectrum Fingerprinting in Asymmetric Cryptographic Protocol. *EURASIP J. Inf. Secur.* **2010**, *2010*, 694797. [CrossRef]
37. Bianchi, T.; Piva, A. Secure Watermarking for Multimedia Content Protection: A Review of its Benefits and Open Issues. *IEEE Signal Process. Mag.* **2013**, *30*, 87–96. [CrossRef]
38. ElGamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
39. Goldwasser, S.; Micali, S. Probabilistic encryption & how to play mental poker keeping secret all partial information. In Proceedings of the 14th Annual ACM Symposium on Theory of Computing, San Francisco, CA, USA, 5–7 May 1982; pp. 365–377.

40. Paillier, P. Public-key Cryptosystems Based on Composite Degree Residuosity Classes. In Proceedings of the Eurocrypt'99, Prague, Czech Republic, 2–6 May 1999; Volume 1592, *Lecture Notes in Computer Science*, pp. 223–238.
41. Rannenberg, K.; Royer, D.; Deuker, A. *The Future of Identity in the Information Society—Challenges and Opportunities*; Springer: Berlin, Germany, 2009.
42. Chen, B.; Wornell, G. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. Inf. Theory* **2001**, *47*, 1423–1443. [CrossRef]
43. Piva, A.; Bianchi, T.; De Rosa, A. Secure Client-Side ST-DM Watermark Embedding. *IEEE Trans. Inf. Forensics Secur.* **2010**, *5*, 13–26. [CrossRef]
44. Canetti, R. Security and Composition of Cryptographic Protocols: A Tutorial. *ACM SIGACT News* **2006**, *37*, 67–92. [CrossRef]
45. Kuribayashy, M.; Tanaka, H. Fingerprinting Protocol for Images Based on Additive Homomorphic Property. *IEEE Trans. Image Process.* **2005**, *14*, 2129–2139. [CrossRef]
46. Kuribayashi, M. Recent Fingerprinting Techniques with Cryptographic Protocol. In *Signal Processing*; Miron, S., Ed.; IntechOpen: Rijeka, Croatia, 2010; Chapter 10.
47. Prins, J.P.; Erkin, Z.; Lagendijk, R.L. Anonymous fingerprinting with robust QIM watermarking techniques. *EURASIP J. Inf. Secur.* **2007**, *2007*, 031340. [CrossRef]
48. Deng, M.; Bianchi, T.; Piva, A.; Preneel, B. An efficient buyer-seller watermarking protocol based on composite signal representation. In Proceedings of the 11th ACM Workshop on Multimedia and Security, Princeton, NJ, USA, 7–8 September 2009; pp. 9–18.
49. *Ethereum*. 2024. Available online: https://ethereum.org (accessed on 11 May 2024).
50. *Solidity*. 2024. Available online: https://soliditylang.org/ (accessed on 11 May 2024).
51. Casino, F.; Dasaklis, T.K.; Patsakis, C. A systematic literature review of blockchain-based applications: Current status, classification and open issues. *Telemat. Inform.* **2019**, *36*, 55–81. [CrossRef]
52. Aggarwal, S.; Chaudhary, R.; Aujla, G.S.; Kumar, N.; Choo, K.K.R.; Zomaya, A.Y. Blockchain for smart communities: Applications, challenges and opportunities. *J. Netw. Comput. Appl.* **2019**, *144*, 13–48. [CrossRef]
53. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
54. Bamakan, S.M.H.; Motavali, A.; Bondarti, A.B. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Syst. Appl.* **2020**, *154*, 113385. [CrossRef]