Important notice

No part of this Whitepaper is intended to create legal relation between a recipient of this Whitepaper or to be legally binding or enforceable by such recipient against CopyrightsWorld.

No part of this Whitepaper should be used or shared as a separate entity. Share this Whitepaper in its entirety only. This notice should be always included.

An updated version of this Whitepaper may be published on a date to be determined and announced by Copyrightsworld in due course.

IF YOU ARE IN DOUBT AS TO THE ACTIONS YOU SHOULD TAKE, SHOULD ALWAYS CONSULT YOUR LEGAL, FINANCIAL, TAX OR OTHER PROFESSIONAL ADVISOR(S)

Copyright notice

CW-chain : An immutable, append-only copyright registration ledger

Alt: CopyrightsWorld's copyright registration formula explained

Georgios V. Efstratiadis
georgios@globyworks.com
COPYRIGHTSWORLD

Abstract. A copyright registration formula that can immediately provide strong, multiple and undeniable **Proof of Ownership** (PoO) to anyone that applies it to its own creations. The formula connects the creators digital id with his creation and the exact date and time the creator applied the formula to it (the creation). All this data is immutably recorded to a transparent, append-only ledger based on the **blockchain** technology. This chain, we call it "cw-chain", is **privately generated** and is **publicly available** to anyone. Recorded and stored data gets encrypted using the creator's public cryptographic key. Every read-only block on that chain can be easily verified by the "block verification" mechanism described below. Copyrightsworld provides the technology and the platform needed to accommodate such functionality.

1. Introduction

The **Berne convention** (http://www.wipo.int/treaties/en/ip/berne/summary_berne.html) binds most of the world (currently 174 of 195 countries) under the basic principle that beneficiary of any creation is the person capable of providing the strongest and earliest **Proof of Ownership** to that creation. Current registration mechanisms are mostly outdated, expensive, slow and in general fail to service that purpose in a sufficient way. By utilising the latest technological trends, Copyrightsworld creates a mechanism "the formula", based on

blockchain-like technology, that provides undeniable **proof of ownership** for its users. That formula will be discussed in this whitepaper.

2. Proof of Ownership (PoO)

When a copyright claim comes to litigation, according to the copyright law the original creator of that creation should be able to undeniably prove that he was the owner of that work at a specific time in the past in order to be declared as the beneficiary of that work. Copyrightsworld's unique formula provides exactly that. **Proof of Ownership (PoO).**

The **Certificate of Ownership (CoO)** our platform provides, verifies that a creator can prove that he submitted his work (so he was in possession of that work at that exact time) at a very precise day and time in the past. That is a solid undeniable **Proof of Ownership (PoO)**, since that digital asset gets **linked to its owner's digital account, encrypted, time stamped, distributed, digitally signed and immutably recorded to our append only "cw-chain" ledger**, making impossible for anyone to alter that information (the proof) without compromising the integrity of the ledger.

3. Submitting a creation

Every registered user creates his digital identity within the Copyrightsworld network. That identity consists of his **name, email, a unique identifier (cwid), a digital signature and a pair of private / public cryptographic keys**. A user must be connected (logged in) in order to initiate a creation submission.

A user initiates the "creation submission" process by selecting the type of creation he wants to submit. **Current supported types are: Small text, Long text, Sound, Visual and Software.**

Next, the user imports the actual creation to the submission process (by entering the creation text in the case of "Small text" type of submission or by uploading the creation's digital file in every other case).

Then, the user should enter some metadata information about the imported creation, such as: Title, Year of creation, Edition / Version etc.
After the user has entered the required information and has verified the correct receival of that information, he can then proceed with the final step of the submission.

At this stage a very complicated process gets initialised. We can't describe in full that process for proprietary and security reasons, but here's what we can publicly post.
Every creation gets **uniquely identified** with the use of **serial cryptographic algorithms**, **validated timestamps** and other security processes. The creation gets undeniably linked to its owners digital id and securely stored.
We manage to be able to **fully describe the creation publicly without the need to reveal the actual creation**. Actual creations can be only accessible and decrypted by its owners and their private keys.

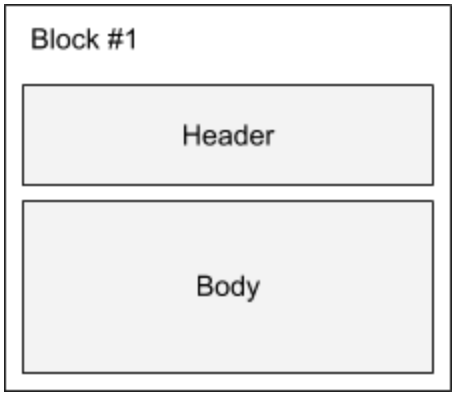
Finally, a submission receipt with all the information produced by the above process, is sent to the user's email address. That message gets **timestamped** by the mail service "third party" (eg. Gmail, Yahoo mail, Outlook etc.). That timestamp matches the records timestamp in CW-database.

That way, the user immediately holds all the information needed to prove the submission action in that specific day and time, time-stamped by Copyrightsworld and another third party (his email service provider).

Copyrightsworld uses the SHA-256 hash of a digital file as a digital identifier for that asset. SHA-256 cryptographic hash function, takes an arbitrary amount of input data and deterministically produces a fixed-length output. That output is called a "hash." It can be used to easily verify that data has not been altered because if any part of the input data is changed and the hash algorithm runs again against that data, that hash completely changes.

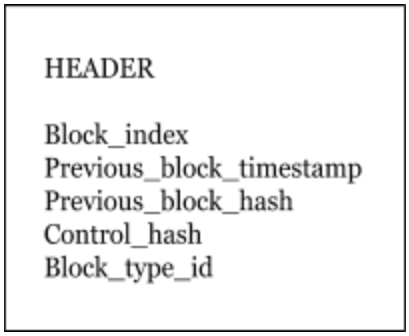
4. The block

Each submission becomes a new block on the immutable, append-only Cw-chain. Each block consists of three parts. The block's index, the header and the body.



The block index is a "key". An auto increment number based on the previous block's index. So if the previous block's index is 5, the new block's index will be 6 (5+1) and so on.

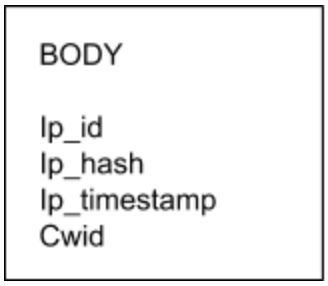
Header structure



Most of the header values are self-explanatory. Below we explain the rest in more detail.

Control_block_hash: The hashed value of the current chain-link session's control block.
Block_type_id: Types of block can be "Small text", "Long text", "Sound", "Visual", "Software", "Control","Blank". We record the ids of those types in the block.
Special blocks [Control block value is: 100. Blank block value is : 0 (zero)].

Body structure



Most of the body values are self-explanatory. Below we explain the rest in more detail.

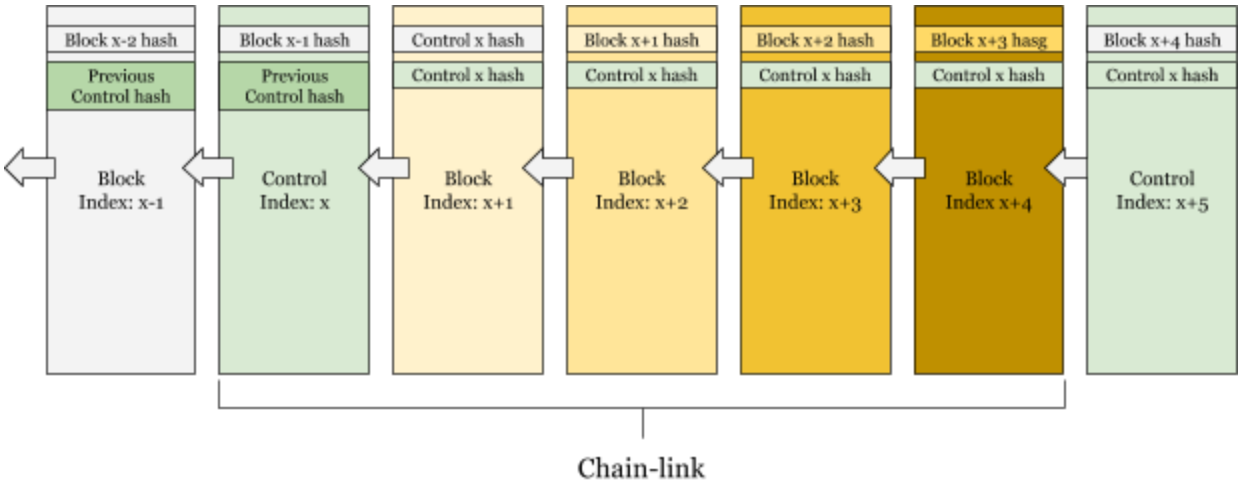
Ip_id: This is the id value this creation got when was recorded in the CW-database.
Cwid: This is the unique identification number every user gets on registration. This connects his digital identity to all his actions within the network.

5. The Cw-chain and it's chain-links

Every new block that gets added to the chain references in its header, the **cryptographic hash of its previous block** and the **current chain-link session's control cryptographic hash**. The cw-chain consists of a series of block groups called "chain-links".

A **chain-link** starts with a special block of type "Control" and ends just before another "Control" block occurs. This special block (control) is used to clearly define the start and end of a new chain-link.

Blank-blocks: These special randomly generated blocks are used in order to avoid multiple "in-a-row" submissions from the same user. **That way the system kills any attempt to "game" the block creation process of the chain.**

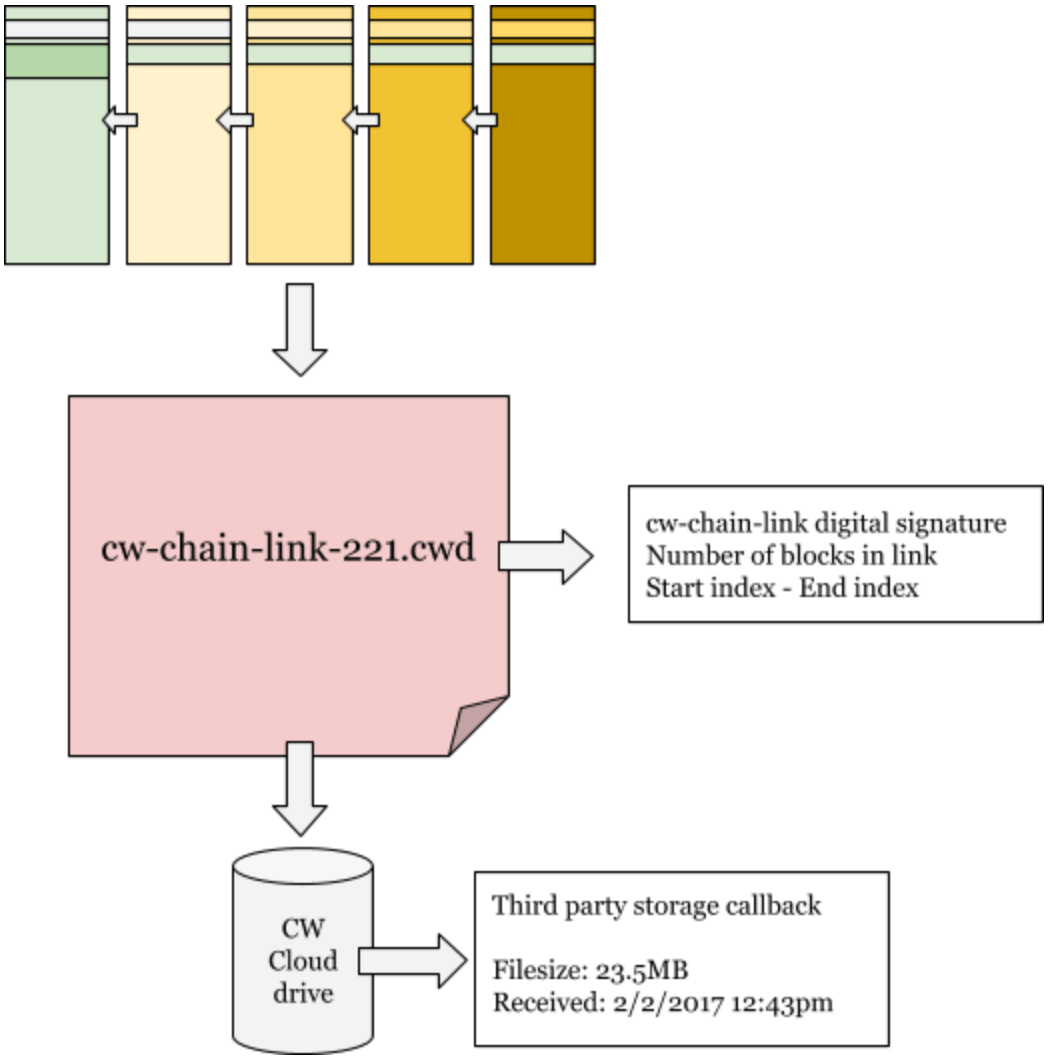


All block based cryptographic hashes occur by **hashing the block as a whole**. There are no exceptions. That way, if any past block changes its data in any way, a new cryptographic hash will occur out of it, making the Cw-chain invalid since all "prevous_block_hash" values following that block will be false. That mechanism ensures the integrity of the chain as a whole.

6. Syncing the chain links

Every a pre-specified time cycle (currently is 24h), a “Cw-chain sync” occurs automatically.

Here’s what happens.



The chain locks itself until the sync process is fully completed. **A full identical copy of the current chain-link at its current status is created as a separate file named “cw-chain-link-index”,** where index is the auto-increment value of that link’s position in the

chain. **A cryptographic hash of that file is generated and stored. The chain-link file gets pushed to a third party cloud storage that we can't control.** What we mean with "we can't control" is that after the file gets pushed to that cloud storage, it automatically gets a filesize value and a timestamp from that third party. These read-only values and the chain-link file's digital hash, are used to verify the file itself.

After the chain-link file gets pushed to that third party cloud storage service, we get as a call back two values from that third party. **A filesize value and a timestamp value. These are strictly read-only values to our network.**

The filename, its digital cryptographic hash and the callback values (size and timestamp) are recorded in the cw-database.

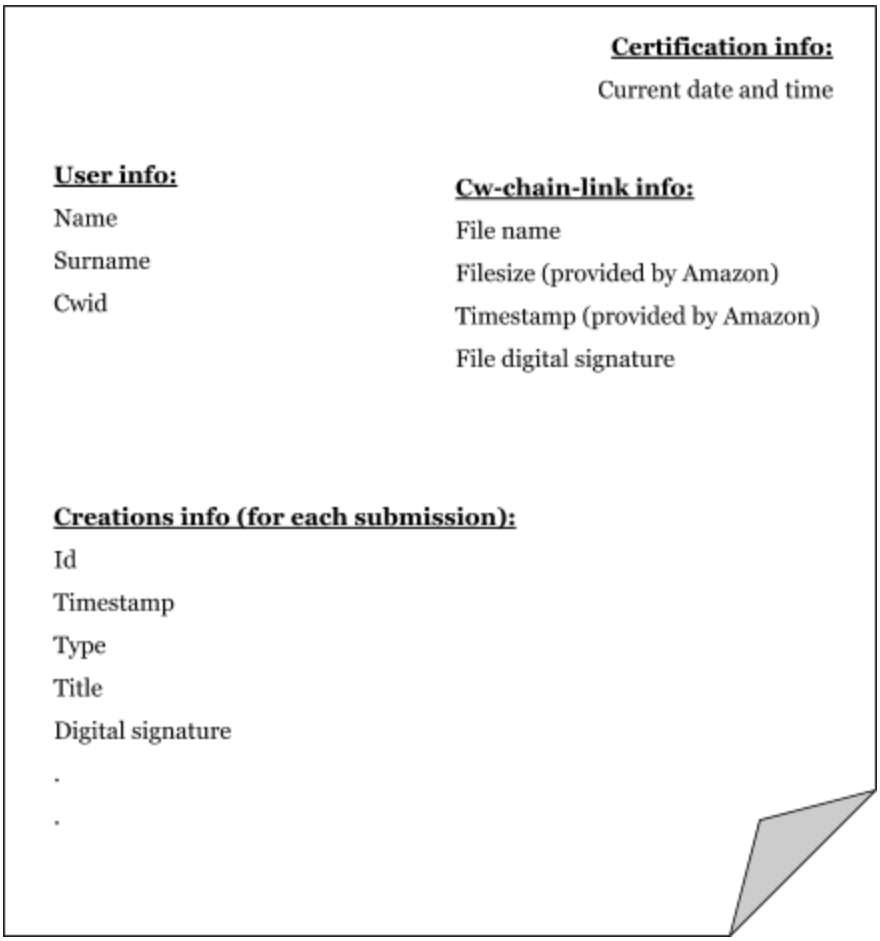
All chain-link files are publicly accessible to anyone. So anyone can download and store the full Cw-chain (consisting of all the cw-chain-link files) at its current status.

7. Immutable certificates of ownership.

Bringing everything together

Immediately after a chain sync is completed, the “Certificate generator” gets triggered. Immutable **Certificates of Ownership** (CoO) are automatically being generated and emailed to every user that had submitted at least one creation within that “chain-link” session.
That **Certificate of Ownership** includes all the details and data produced by the sync process.

Data included in every Certificate of Ownership:



Again, every Certificate gets timestamped by a third party email service provider on arrival.

Chain listeners

All users can be subscribed to become a “Chain listener”. These users will receive a notification about every “chain-link” being synced to the chain along with all the details (chain-link filesize, timestamp etc.), no matter if they have contributed to that chain-link session by submitting a creation of theirs or not. **That way we create a distributed network effect of the cw-chain metadata.**

8. Creating consensus

Every record, digital hash, file size value and timestamp (the output), is contributed by third parties and propagated to our users **creating a network effect**. All those different and complex generated values, land in the user's mailbox at the end of every chain sync. That way, the user has all the information he needs, in order to prove his actions in any case, and that information must agree with the cw-chain data in order to be valid. Also, the cw-chain is **publicly available** to anyone at anytime and can be locally stored.

9. Verifying the chain

There are two ways to verify the chain. A **full chain verification** and a **chain-link verification**. When a full chain verification occurs, the system will pull all cw-chain-link files and generate the full cw-chain out of them. Another cw-chain will be recreated, using the original records stored in its cw-databases.
If those two chains are identical, then the cw-chain passes the verification process successfully and is considered to be verified and in a healthy status.

A partial verification can occur in a random manner. A random index number is generated and the cw-chain file matching that number is pulled for check. The above process is then executed, but just for that cw-chain session.

10. Verification badges

When a chain-sync has been completed and every creation has been successfully recorded in the cw-chain, a verification badge can be obtained. That badge can accompany every public screening of a creation visually verifying the registration of that asset to the append-only ledger. By clicking that badge a viewer will be transferred to a verification page where the **Certificate of Ownership is verified and creation's metadata, timestamp, digital signature and ownership information gets displayed**. A CopyrightsWorld Verification Badge indicates **proper attribution** for every creation submitted and recorder in the cw-chain.

Sample verification badge

11. A bulletproof chain

As mentioned above, Cw-chain is a transparent, append-only ledger in a form of a chain. Since this is a privately generated chain, we needed a mechanism to secure its integrity by **decentralising** the actual data from the system that created it.

We accomplish that by **sharing the chain with third parties**, getting back feedback that we can't control but fully describes the current status of the chain (such as filesize, digital signature and timestamp) and by sharing all that information immediately and directly with every user via multiple third party email service providers.

In other words, if we wanted to change anything that has been recorded in the cw-chain, we would have to change the records on our databases, regenerate every Cw-chain-link, upload the files to every third party cloud storage, hack its filesize and timestamp values (on every cloud storage service provider), hack the user's email account, hack the third party email service provider in order to change the certificate of ownership values and the actual email timestamp. And all that, assuming that no user has a copy of the Cw-chain or his certificate of ownership stored offline.
Moreover, the digital signatures of the creations, secure the data integrity since those signatures (hashes) are recorded in the chain and mentioned in the certificates of ownership all users receive. Any data alteration, would change those values making the submission inconsistent and breaking the chain's integrity.

As a result, this formula makes it impossible to anyone, even us who run the service behind it, to alter any data once it has been added to the chain and a chain-sync has occurred.