Incentivised Distributed Computation for Registered Digital Content Search

INTRODUCTORY WHITEPAPER

Samuel Brooks
Veredictum

July 2017

Abstract

Piracy events move video and audio files from their authorised, revenue-generating location, to other locations on the internet unbeknownst to the copyright holder. Early detection of these events is key to issuing take-down notices in order to preserve rightful revenues. Modern video steganography and blockchain technology allows us to watermark digital content and immutably link this to an authorised identity. Centralised download and decoding of video files to automatically ascertain whether a file contains a watermark is prohibitively expensive. This paper explores the feasibility of a *distributed* system that can be used to download and decode video files on the open internet to "search" for pirated video and audio files.

v0.9.4

A REFERENCE IMPLEMENTATION OF THIS DESIGN MAY CHANGE FROM THE DESCRIPTION CONTAINED WITHIN THIS DOCUMENT

Contents

1 Introduction

This paper discusses a novel method for searching for watermarked video content on the internet.

Tracking the movement of digital content online presents significant challenges due to the client-server architecture of the internet; files are hosted on opaque remote servers and one must browse, download, and inspect the contents of the file in order to understand if a certain piece of digital content is present. Further, digital files are easily and frequently re-distributed to uncontrolled and unauthorised web servers (digital piracy). File names can be easily changed, files can be compressed, down-sampled, and otherwise transformed, all the while preserving the content to be suitable for human viewing.

Given this client-server architecture, telemetry from digital files (status information sent from a remotely configured device back to a central location) is practically impossible. File telemetry would require the cooperation from the remote server to install additional software and transmit information relating to its digital content inventory. Server owners have no particular incentive to distribute an accurate catalogue of the digital content held on their servers.

Blockchain-based cryptocurrencies can incentivize large networks of machines to perform computations as proofs-of-work [1]. Within the context of video content being commonly pirated (such as freebooting between YouTube and Facebook), this paper describes the technical feasibility of an incentivised, distributed search capability for digital creative content that has been "marked" using modern steganography. This capability involves the downloading and processing sectors of the internet via a distributed network of computers (rather than a cluster of dedicated centralised servers), in order to develop a profile of where pirated content has been distributed across the internet. We focus our attention on video content in particular (being a superclass of audio content).

2 Overview

2.1 Problem statement

Video piracy is a significant parasitic cost to major film studios and small content creators alike. Several studies have suggested that the cost to the US economy alone is up to $20B USD [2].

In 2016, Facebook generated over $30 billion in revenue, and yet 73% of the top-performing videos distributed on the social network were stolen from content creators who had originally uploaded their work to YouTube [3]. This is more commonly known as "Freebooting".

Currently, in order for a content creator's stolen content to be removed from Facebook, they must:

1. initially be alerted to the fact the content has been stolen and re-uploaded onto Facebook, and
2. complete a lengthy online form as well as upload the original content to Facebook in order to help prove the copyright claim and have the material removed.

While the YouTube-Facebook interplay is a high-profile example of the nature of the problem (a recent YouTube video discussing the subject ironically had more views on Facebook [4]), it is not the only place where it occurs [5]. A solution that could eliminate this kind of inefficiency would have a significant reduction on the revenue drag for online video, resulting in both higher-quality content and lower prices for consumers.

2.2 Centralised Versus Distributed Computation

A single entity intending to search the open web for a single piece of content is facing the manual review of an intractable number of files. In the case of Facebook, approximately 1 Petabyte of data is added to their servers each day, around 1/3 of this as video [6]. In order to independently verify the copyrights of uploaded video, this 300TB of new video data would need to be downloaded and examined on a daily basis. Using a centralised cloud service such as Amazon Web Services (AWS), this analysis would cost of the order of $100,000 USD [7].

2.3 Solution summary

This paper describes the technical feasibility of a content download and review function performed by a distributed network of incentivised nodes. The system relies upon the steganographic pre-processing of digital files (watermarking) and the registration of ownership to a public blockchain. Then consuming the search space suspected of serving copyrighted material would provide a profile of when and where the content has been re-distributed (illegally and legally).

Our system is defined in two key phases:

1. **Content Encoding and Registration:** In the first phase, content is encoded with a unique identifier using a centralised server. We then register this identifier to a public blockchain (e.g. Ethereum) and distribute the content to its intended destination.
2. **Content Search and Reporting:** We then assume that some piracy event will occur with some significant probability. We then search, download and review a number of suspect video files using a set of incentivised computers, and report results back to the subscribers of the service. Coordination of this phase is centralised and the computational effort is distributed.

We make some assumptions regarding the cost of latent processing and bandwidth resources available in devices distributed across the internet and leverage this to propose a practical content search engine for registered content. We describe an incentive scheme, inspired by Ethereum, designed to ensure that computations are verified to within a high probability.

With a more mature distributed technology ecosystem, it may be possible to ultimately perform the centralised tasks in a distributed fashion. We leave this possibility to future work.

3 High Level Solution

3.1 Definitions

3.1.1 Digital content and files

We define digital *content* as distinct from a digital *file*. Content is defined as static and files are defined as polymorphic; the file is simply the container for the content. The properties of a digital file can change significantly but the nature of the content remains the same (e.g. a music file can be down-sampled and have its filename and file type changed but the file can still contain content of Chuck Berry singing Johnny B. Goode). The *content* is the valuable asset we wish to protect; the file itself has no intrinsic value.

3.1.2 Client and server

We refer to a *client* as being the application that a node will download from a central server to participate in the system. We refer to this centralised, orchestrating server as the *Coordinator*.

3.1.3 Logical actor definitions

3.1.3.1 Subscribers

Subscribers are customers who pay for the search capability. It is envisioned that the fees associated with the functionality are paid on a subscription basis so that the expected pay-through to miners is more predictable.

3.1.3.2 Miners

A miner is a person participating in the distributed computation network to earn cryptocurrency. For the purposes of this whitepaper we assume a single miner per client, however in practice there may be multiple clients under the control of a single person or company.

3.1.3.3 Clients and Nodes

A client is a single instance of the software on a miner's computer. We define a node as a single virtual machine (VM) within a client. There can be multiple VMs instantiated per client if the underlying hardware is sufficiently powerful.

3.2 Solution Phases

3.2.1 Phase 1: Content Encoding and Registration

The process of content registration is outlined below. These tasks are assumed to be operational as this paper mainly deals with the distributed search engine. Resilience testing results from our in-house steganography software is covered in our full technical whitepaper.

- Securely establish the identity of the content owner who wishes to register a new piece of creative content.
- Generate a new, unique identifier for the content.
- Steganographically embed this new content identifier within the file.
- Register the content identifier to the content owner using a smart contract on a public blockchain.
- Distribute the file to its approved destination(s).
- This process is ongoing.

3.2.2 Phase 2: Content Search and Reporting

The search phase consists of:

- Crawling servers of interest to compile a list of video files available for download.
- Sorting the search space to attempt to prioritise files that have a higher probability of containing an identifier.
- Partitioning the search space into a number of sectors such that each sector is able to be downloaded and processed by a single node in some reasonable time period (e.g. a sector size of 1GB).
- Recruiting a set of computers to download and process each sector.
- Returning the results to a central server (the Coordinator) to create a profile of files that contain content identifiers for that cycle.
- This process then repeats once per cycle time.

3.3 Video steganography

Video steganography is the practice of concealing a message within a video. Modern techniques exist that enable a video file to be transcoded into a new file that contains an embedded message that is both invisible and inaudible to regular viewing. This provides a method of 'watermarking' the content so that when it is later examined, a unique content identifier can be recovered that associates the content with the identity laying claim to the copyright. Given the immutability of the blockchain and the strength of modern cryptography, this claim becomes extremely strong.

In order for the marked video to remain valuable, it must be of near identical video quality as well as be resistant to transformations designed to defeat the reproduction of the embedded identifier (such as horizontal flipping of the video or re-recording).

Modern steganographic techniques achieve this with no noticeable degradation in audio-visual quality and with high transformation resistance (resistance strength is exponentially related to the quality of the resultant file). This means that even with a high degree of transformation, recovery of the embedded information may still fail to detect the watermark, but only where the video quality is so degraded that the content becomes effectively worthless for consumer applications. The results of robustness testing of our steganography software are provided below.

3.4 Search space preparation

A digital content search space is indexed to produce a list of target URLs for further processing. This process initially utilises a web crawler to produce a raw index of video files from a list of target domains. The crawler operates according to certain conditions optimised for our application, such as: do not follow hyperlinks and only index seed domains.

The search space is sensibly bounded as to maximise the probability of finding marked digital content. This can be done for example through prioritising domains and subdomains that are suspected of holding unauthorised content. Bayesian probability and learning algorithms can be applied to both (a) minimise the search space over time in order to reduce the compute resources required and (b) rank individual files by their probability of being watermarked. This information can be aggregated to form a 'piracy reputation' for web servers and specific sub-domains, such as individual users on social media.

Once the search space has been sorted, it is segmented and provisioned to the connected nodes in the distributed computation network.

3.5 Client download, configuration and connection

Participating users in the distributed computation network download and install an application (the 'client') from a central server (the 'Coordinator'). A precise copy of the correct version of software must be installed due to the need to take snapshots of the internal state of the application as it performs computations (see below sections relating to verifiable computations). This can be verified via regular checksum matching.

The client contains an internal virtual machine cluster and manages the interfaces to both the Coordinator and Arbitrator. The client also manages the file download and state hashing of the virtual machine and provides a user interface.

Account log-in by the miner is required via the client. This ensures the miner's reputation can be tracked and that sufficient cryptocurrency is available in their account to form a deposit. These are necessary for participation in the network in order to enforce the design of the game-theoretic computation reperformance. The process of acquiring cryptocurrency is beyond the scope of this document.

The client is also intended to manage scaling; if a miner has access to greater computer resources (such as a server farm), the client will check the user-defined settings and then automatically spawn additional nodes to use the available resources. This effectively homogenises the nodes in the network, which simplifies the matching of nodes by the Coordinator and enables a greater number of connections to be made for better randomness (higher value of n for n choose k).

Once the client has made a server connection, it submits information relating to the user-defined settings and details of the available hardware resources on the device.

Once a cut-off period has been reached, the Coordinator:

1. Stops accepting new nodes for the next search cycle, but starts a queue for the one after. A search "cycle" is one period where a finite set of file URLs are collated for download and processing, and one tranche of a finite amount of cryptocurrency is gathered from subscribers to pass through to the distributed computation network. The Coordinator also analyses the total computational capacity in the network and assigns the sub-networks ("subnets"): nodes are grouped into small sets (e.g. 2 or 3) that have been chosen to perform the computation on the same search space partition. Nodes are chosen in such a way as to minimise the possibility of collusion, such as random selection with geographical segregation. The subnets are assigned an ordinal at random, in the special case that the number of subnets exceeds the number of subspaces, the remaining subnets are re-assigned their ordinal at the start of the next cycle. We do not consider if this occurs past more than one cycle.
2. Provisions nodes with their allocated URL list (subspace). Note that by this step the number of subnets equals the number of subspaces.

Note also that there is no requirement for nodes allocated in the same subnet (i.e. peers) to connect to one another directly; the Coordinator does not provide any information relating to what nodes are in the same group. Note too that the subnet size can decrease to below 2; a subnet size of 1.5 would indicate that every second node's state is replicated at random, meaning that each node would know that, on average, half of their submissions are checked by an independent node, but would not be aware of which submissions specifically. This is explored further in the discussion of reperformance design.

3.6 Client processing

All nodes that are connected in the distributed network utilise their bandwidth and processing power to download and process their allocated sector of the total search space.

In order to verify that nodes are performing the computation correctly and not returning bogus results, at regular intervals within the virtual machine, the state is hashed to form a Merkle tree. Merkle tree roots are then submitted from the network back to a centralised module for comparison. If all the Merkle roots are identical for all nodes in the subnet, this means that either all nodes have performed the computation correctly, or they have perfectly colluded in an attempt to fool the system. Attack surfaces and verification of computations are explored in our full technical whitepaper.

Once the allocated search space has been processed, the recovered content identifiers are also submitted back to the central Coordinator and compared. If the results are consistent, then no more action is taken by the client until the next search space is provisioned; the results are deemed valid.

3.7 Content telemetry

After the distributed network has resolved the content identifiers from the network, the submissions are collected by the Coordinator module into a central database. We now have an up-to-date profile of what content exists across a number of web servers and this can be queried by subscribers (with appropriate access permissions). The information can be used to automatically send offending servers a Digital Millennium Copyright Act (DMCA) Take-Down Notice or serve an invoice or Letter of Demand. The amount requested can be manually configured or automatically calculated based on simple metadata metrics, such as how many views the video has received.

In summary, network orchestration manages the following process:

- Miner downloads the client and is authenticated via username and password.
- Client checks to see if the system is running any other clients.
- Submission of deposits where necessary.
- Client collects system information, internet connection diagnostics and user settings, and then submits this information to the central Coordinator.
- The Coordinator creates a network pool and a sorted search space, then allocates the URL sets to the nodes.
- Client downloads and processes the URLs and returns state information (Merkle tree).
- Coordinator collects state results from the network’s computations before finally collecting the watermark results to form a searchable database of URLs for registered content.

3.8 Client compensation

The value transfer between the subscriber (the customer wishing to search for their registered content) and the miner (the service provider performing the download and decoding) is facilitated via a cryptocurrency token. This payment for the expenditure of computational resources is managed via a smart contract. Fees for the task are deposited, in tokens, by the Coordinator into the contract for the benefit of each node in the network before the computations have been initiated. Upon the successful completion of the submission of their states, the Coordinator sends a transaction to initiate the update of account balances based on how many jobs were completed (how much computational effort was expended). Initially, payment is made into an escrow account to be held for some time period to ensure any bogus results can be challenged. If the time elapses and no challenge over the results has occurred, the tokens are unlocked and the node can retrieve them. At this point, the person who earned the cryptocurrency is able to then sell it on an exchange back to subscribers.

The smart contract is also required to hold the deposits of all participating nodes. Deposits must be made before each node is issued with its partition of the search space. The deposits are required to ensure that a penalty can be applied in the event that a node submits incorrect information (i.e. when the states do not match, or when a challenge is initiated).

The cross-checking of state submissions is performed centrally. Ideally, this calculation would be performed by public blockchain infrastructure, however our system generates far too much data for this to be feasible with the current state of technology. Hence, this “trust” must be placed within the Coordinator in performing the function of verifying computations of remote nodes correctly. This is important because it is the checking of the state submissions that determines whether a miner has acted properly; only on the basis that all the states agree can the payment to the miner be approved.

3.9 System Components

Our system is logically segmented into the following high-level software components which will be explored further in the technical whitepaper:

- **Web Application** (web application for users to interact with the system)
- **Transcoder** (application to embed content IDs into video)
- **Registrar** (smart contract registering ownership of content)
- **Indexer** (search space web crawler)
- **Sorter** (search space sorting algorithm)
- **Coordinator** (central application coordinating the distributed nodes)
- **Client** (the application running on the node's machine)
- **Arbitrator** (central coordination of state matching and resolution)

4 Bibliography

[1] filecoin.io, “Filecoin: A Cryptocurrency Operated File Storage Network,” 2015.

[2] C. Bialik, “Putting a Price Tag on Film Piracy,” 5 Apr 2013. [Online]. Available: https://blogs.wsj.com/numbers/putting-a-price-tag-on-film-piracy-1228/.

[3] Tubular Labs, “The Rise of Multi-Platform Video: Why Brands Need a Multi-Platform Video Strategy,” Social@Ogilvy, 2015.

[4] Kurzsegagt, “How Facebook is Stealing Billions of Views,” 10 November 2015. [Online]. Available: https://www.youtube.com/watch?v=t7tA3NNKF0Q.

[5] L. Overweel, “Stop Freebooting Now,” [Online]. Available: http://stopfreebootingnow.com/#wall.

[6] Facebook, “Facebook’s Top Open Data Problems,” October 2014. [Online]. Available: https://research.fb.com/facebook-s-top-open-data-problems/.

[7] Amazon, “AWS Cost Calculator,” 2017. [Online]. Available: https://calculator.s3.amazonaws.com/index.html.