Bitmark: The property system for the digital environment.

Christopher Hall, Casey Alt, Lê Quý Quốc Cường, and Sean Moss-Pultz

November 7, 2016

Abstract

This paper proposes a digital property system that achieves secure authentication and transfer of both digital and physical objects, from one party to another, without requiring a central authority. Digital signatures provide a method to issue and transfer titles ("bitmarks") within the system. Using a blockchain algorithm, distributed consensus on who owns what can be achieved. Digital assets can be uniquely identified using cryptographically hash functions. ObjectMinutiae provides a method to uniquely identify physical assets. Title transfers are peer-to-peer, verifiable, and create an unforgeable chain-of-ownership ("provenance").

Keywords

Property systems, digital property, ownership, digital assets, property tiles, property rights, secure provenance, Nakamoto blockchain, decentralized systems

1 Introduction

One of the most important functions a formal property system does is to transform assets from a less accessible state to a more accessible state, so that ownership can be easily communicated and assembled within a broader network.

Converting an asset such as a house into an abstract concept such as a property right requires a complex system to record and organize the socially and economically useful attributes of ownership. The act of embodying an asset in a property title and recording it in a public ledger facilitates a consensus among actors as to how assets can be held, used, and exchanged.[1]

[1] Hernando de Soto Polar, *The Mystery of Capital: Why Capitalism Triumphs in the West and Fails Everywhere Else*, 2003.

For hundreds of years property systems have evolved to handle property that exists in the real world—beginning with land and buildings and eventually encompassing even ideas. When computing technology developed to make digital property possible, lawyers naturally tried to make the existing property systems handle those digital assets. Yet this has proven to be more difficult than integrating intellectual property. (Digital creates a more tangible version of the unusual economic nature of information that we can all own an idea at the same time.) Companies trying to come up with a practical solution to the difficulty of handling digital property switched the conversation from ownership rights to licensing. But licensing digital assets for use is as different from developing property rights as renting real estate is from owning buildings. Our definitions of digital property have broken, and the patches we tried to build have not work.

The point where digital property currently finds itself structurally parallels the precipice of advancement that happens in many scientific fields. When Newtonian mechanics could not adequately describe the behavior of subatomic particles, physicists first tried to patch solutions. It was only when they stepped back and changed perspectives entirely that they discovered a new framework, quantum mechanics, that could explain the behavior of subatomic particles, and also the larger phenomena previously described by Newtonian mechanics. We need a new system of digital property ownership that can, once developed, also extend similarly to describe the existing frameworks of ownership.

This new system needs to be built from the perspective of the digital. Redefining digital property—true property rights that introduce digital scarcity alongside title and authenticity—can encompass physical property as well. The reason that the new system can handle physical and digital assets the same way is that title—which confers ownership rights—is already an abstract container that to begin with, is aligned with the properties of digital.

We propose using the “ObjectMinutiae”[2] framework to securely identify (“fingerprint”) physical assets based on unique surface-level texture patterns. Cryptographically-safe hash functions can be used to fingerprint digital assets. Digital signatures provide a method to issue and transfer titles (“bitmarks”), and using a blockchain algorithm, [3] distributed consensus on who holds title can be achieved. Title transfers are peer-to-peer and verifiable, and create an unforgeable chain-of-ownership (“provenance”).

Not requiring a central authority to operate or secure the whole system increases efficiency and lowers costs without being highly vulnerable to fraud and data loss. Scarcity of digital properties is also possible and can accommodate the conceptual and legal frameworks of the physical world. [4]

[2] Tzu-Yun Lin, Yu-Chiang Frank Wang, Sean Moss-Pultz, “ObjectMinutiae: Fingerprinting for Object Authentication”, 2015.

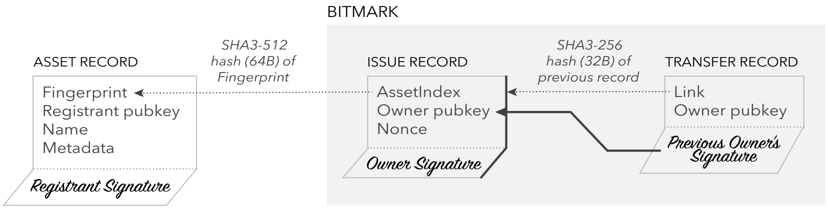
[3] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, https://bitcoin.org/bitcoin.pdf, 2008.

[4] Nick Szabo, “Scarce Objects”, http://szabo.best.vwh.net/scarce.html, 2004-5.

The techniques proposed should not be confused with “Digital Rights Management” (DRM). [5] DRM is outside the scope of this paper. Getting actors to respect the recorded property rights depends on the specific nature of the property and legal jurisdiction. What is proposed here is a method to securely agree upon who owns what. By giving all assets the same inviolable digital identity or title, we can adequately describe the ownership rights of physical, intellectual, and digital property.

2 Transactions

A *bitmark* is defined as a digitally signed chain consisting of a single *Issue Record* and one or more *Transfer Records*:



Users of the system are identified by their Ed25519 public keys.[6] An *Asset Record* contains metadata for a physical or digital asset as well as the unique asset fingerprint used to identify it within the Bitmark system. The Asset Record has the following fields:

- *Fingerprint* - hash of a digital representation of a physical object or digital file
- *Registrant* - public key (Ed25519) of user registering the asset
- *Name* - short UTF-8 identifier
- *Metadata* - key-value pairs of identifying UTF-8 text separated by NULs
- *Signature* - hash of the above fields signed by registrant’s private key

An *Issue Record* creates a new bitmark from an Asset Record. It establishes a link between the asset and digital information in the system and has the following fields:

- *AssetIndex* - a SHA3-512 hash (64 bytes) of the corresponding Asset Record’s *Fingerprint* value. The *AssetIndex* serves as a unique identifier for the Asset Record and will be identical across all Issue Records

[5] Wikipedia, “Digital rights management”, http://en.wikipedia.org/wiki/Digital_rights_management

[6] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, Bo-Yin Yang, “High-speed high-security signatures”, http://ed25519.cr.yp.to/ed25519-20110926.pdf, 2011.

for the same Asset Record. The Asset Record *Fingerprint* is hashed as a means for guaranteeing a consistent size regardless of the original size of the *Fingerprint* value.

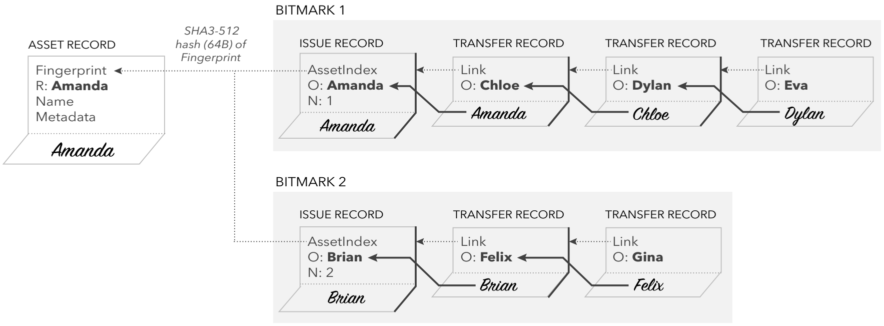
- *Owner* - the public key (Ed25519) of the user who created the issuance. When a new issuance occurs, the Issue Record is automatically owned by the issuer.
- *Nonce* - an unsigned integer that serves as a unique number to distinguish multiple issuances of the same asset.
- *Signature* - hash of the above fields signed by the issuer's private key

A *Transfer Record* records ownership changes of a bitmark. It has the following fields:

- *Link* - a SHA3-256 hash (32 bytes) of the entire previous record (including signature), which indicates the previous record in a bitmark's chain-of-ownership. The previous record may be either an Issue Record or another Transfer Record. The previous record is hashed as a means for guaranteeing a consistent size regardless of the original size of the previous record.
- *Owner* - the public key (Ed25519) of the bitmark transfer recipient.
- *Signature* - a hash of the above fields signed using the private key of the previous record's owner.
- *Countersignature* - a hash of the above fields (including *Signature*) signed using the private key of the new record's owner.

The bitmark's current owner (the rightmost record in the chain) is verified by checking the digital signatures in the chain. If a Transfer Record's digital signature matches the public key of the previous record's *Owner* **and** the countersignature matches the *Owner* field in this record, then the Transfer Record is considered valid and is recorded in the blockchain. If not, the invalid Transfer Record is discarded. The original Asset Record is verified by validating its reference fingerprint against the actual object.

Asset Records are self-signed. Thus, *any user* can issue new bitmarks for an asset:



In this case, *Eva* and *Gina* are both current owners (since they hold the last transfer records in their respective bitmark chains). Conflicting ownership claims stemming from bitmarks that point to the same asset yet have different issuing signatures must be settled externally by property rights adjudication. As an immutable, enduring history of all property transactions, the Bitmark blockchain will serve as evidence.

3 Operation

Note: This section assumes familiarity with Nakamoto blockchains. The reader is referred to the Bitcoin Wiki [7] *for an introduction to the topic.*

The system creates and processes transactions through a peer-to-peer network. High-level functionality is provided by the following parts:

- Client (GUI)
  - Connects to the JSON RPC of bitmarkd to send out transactions
  - Handles key generation and storage
- Server (bitmarkd)
  - JSON-RPC listener for client transaction submission
  - Custom P2P binary protocol for blockchain and transaction broadcasting
  - JSON-RPC listener for administration commands
  - Custom protocol for miners
  - Data storage in LevelDB database (individual tables are differentiated using a prefix byte)

The client connects to bitmarkd's RPC port and sends the transaction as a JSON-RPC request. Bitmarkd verifies the signature of the transaction. Asset Records and Issue Records are self-signed, whereas Transfer Records must be signed by both the current owner and the new owner. Invalid signatures and incorrectly linked records are rejected. Valid transactions are pooled as unpaid items and broadcast to other servers in the peer-to-peer network.

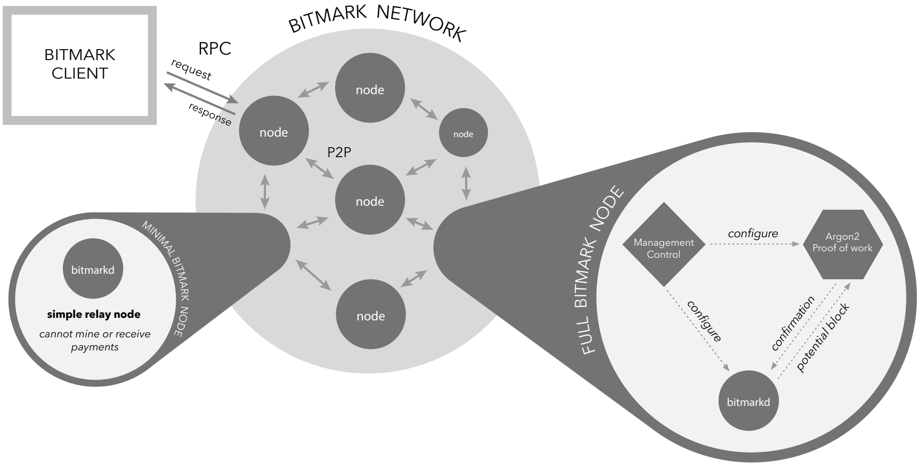
For each unpaid issuance transaction, bitmarkd will return a *payment id* and an array of *payment pairs* – currency names and payment addresses – that the network will accept as payment for mining the transaction (the "fee"). Alternatively, there is a *payment nonce* and a *difficulty* so the client can mine for a nonce. Similarly, for each unpaid transfer transaction, bitmarkd will return a *payment id* and an array of *payment pairs* – currency names and payment addresses – that the network will accept as payment for mining the transaction

[7] Bitcoin Wiki, "Block chain," https://en.bitcoin.it/wiki/Block_chain.

(the "fee"). However, unlike issuances, transfers do not have a payment nonce or difficulty.

Multiple issuance transactions (but not transfers) are to be paid in a single payment by paying scaled fees returned or mining a nonce to the scaled difficulty. Scaling is based on a maximum of 100 issuances submitted in a single call, and the fee is discounted compared to sending single issuance transactions.

From this response, the client constructs a payment transaction and sends it to bitmarkd for verification and relay. Servers wait up to one hour for payment to be received before expiring the record. Once payment is confirmed, the record(s) can be mined.



While much of the Bitmark system could have been developed directly using the Bitcoin blockchain, irrevocably binding a property system onto a network designed primarily for payments is not a sustainable long-term strategy.

The mining process itself is external to bitmarkd and uses a custom protocol. Each bitmarkd server accumulates available transactions into a list and computes a Merkle tree of transaction digests. A check is made for Issue Records to ensure that an Asset Record will be included before the Issue Record (i.e., the related asset either has been mined in a previous block or is known to the bitmarkd).

A Block Header containing the block number, 64-bit timestamp, and a Base record containing the Block Owner's payment address is created and broadcast along with the Merkle tree to the subscribed miners. If a miner is successful, it will return the nonce values it found. Bitmarkd will then create the full header and base along with the Merkle tree and verify that the digest is within the current difficulty and that its block number is one higher than the current block number. Blocks that meet both conditions are incorporated into the current blockchain.

A Bitmark Base Record establishes the payment address for this block. This address is used when determining the payments for issuances and transfers.

The data stored in the Block Header is as follows:

- *Block Number* - 8 bytes, little-endian
- *Timestamp* - 8 bytes, little-endian (UTC Unix time in seconds)
- *Merkle Root* - 32 bytes, Root hash of the Merkle tree
- *Diffiulty* - 8 bytes, Difficulty at the time the block was mined
- *Transaction Count* - 8 bytes, for miner to use
- *Previous Block Hash* - 32 bytes, Argon2[8] hash of previous block
- *Nonce* - 8 bytes, for miner to use

The data stored in Base Record is as follows:

- *Extra Nonce* - 8 bytes, for miner to use
- *Currency Name* - 0..16 bytes, lowercase ASCII currency name (e.g.: "bitcoin")
- *Payment Address* - 0..64 bytes, ASCII address of miner to receive payment (e.g.: Base58 Bitcoin address)

When a bitmarkd server receives additional transactions, it will periodically broadcast new work to all connected miners. A correctly solved block will have all of its transactions set to a mined state, thus removing them from the available pool. The server then continues to work with the remaining available transactions.

Mining will be suspended and the server will go into recovery mode until the pools of available transactions are fully reconstructed if any the following conditions occur:

1. a new block is received with a number higher than the next expected block
2. the server was offline for a time (or just missed some blocks)
3. the blockchain forks

The server recovers by determining the highest available block from neighbors and then fetches block hashes in reverse order, overriding any older blocks until its blockchain is consistent with neighboring blockchains.

Once all blocks have been received and their corresponding transactions have been set to the mined state, mining can resume.

[8] Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich, "Argon2: the memory-hard function for password hashing and other applications", https://github.com/P-H-C/phc-winner-argon2/blob/master/argon2-specs.pdf, 2016.

4 Light Ownership Verification

It is possible to verify the current owner of any bitmark within the system without running a full network node. Servers internally maintain a table of the current owners for each bitmark and thus can verify ownership requests from Clients with an easy lookup query.

There are vulnerabilities to this method. Among other concerns, this method is only reliable if honest nodes control the network. Therefore, actors that frequently transfer or receive bitmarks should run their own full nodes. Running full local nodes is also better for independent security and faster verification.

5 Incentive

There is a natural incentive within any property system: Participants implicitly agree that assets with titles are more valuable than those without. Titles represent an asset's potential to create value. Titles are what grant basic rights, such as the ability to resell, rent, lend, and donate.

Anyone can issue bitmarks to just about anything, yet the permanence of a signed Issue Record should deter bad actors from issuing bitmarks to assets with disputed ownership rights. When legal disputes arise, each provenance will serve as evidence in conflicting ownership claims.

The incentive to mine is funded with transaction fees – payable in currencies such as bitcoin – and also helps prevent abuse of the system. The transaction fee is the difference between the output value of a payment transaction and its input value.

6 Privacy and Identity

By necessity, the system announces all transactions publicly. Privacy can still be maintained by keeping public keys anonymous. As an additional precaution, a new key pair can be used for each issuance to prevent linking back to a common owner.

Owners may wish to reveal their identity within the system. Institutions such as museums often want their holdings known. A public key infrastructure (PKI) can be used by Clients to verify that a particular public key belongs to a certain entity.

7 Conclusion

This paper introduces a global property system that can adequately describe ownership rights by giving all assets a digital identity that is inseparable from ownership. Creation and transfer of property rights is enforced by protocol and employs a Nakamoto blockchain to record an unforgeable provenance. Architecturally, key technical aspects with Bitcoin are shared to enable decentralized payment.

This proposed solution broadens the way in which a property network can operate because it is sufficiently general to handle digital, intellectual, and physical property. The system's decentralized structure protects against fraud and allows it to function across political and economic environments, to create the broadest network for the authentication, trade, and management of ownership.

Change log

- *November 7, 2016* - Updated to reflect revised production blockchain structure and mining process. Changed method to authenticate physical assets from PUFs to ObjectMinutiae.
- *April 7, 2015* - Initial release.

Acknowledgments

Timothy Chen, Mike Hearn, Casey Reas, Bunnie Huang, Amy Whitaker, Peter Schwabe, Jenny Pat, Agnes Yen