A Decentralized Private Marketplace: DRAFT 0.1

Ido Kaiser[1]

***Abstract*— The online services we use are increasingly demanding more of our personal data, a disturbing trend that threatens the privacy of users on a global scale. Entities such as Google, Facebook and Yahoo have grown into colossal, seemingly unaccountable corporations by monetizing their users' personal data. These entities are charged with keeping said data secure and, in the case of social and economic interactions, safeguarding the privacy of their users. Centralized security models are not applicable to the new generation of technologies such as Bitcoin. This paper discusses a system which combines a Bitmessage-style network with anonymous payment schemes to create a privacy-centric marketplace. Furthermore we apply a multi-signature escrow technique involving insurance deposits should which deter fraudulent actors from participating in trades, given that their incentive is to make a profit.**

I. INTRODUCTION

Satoshi Nakamoto, the visionary and creator of Bitcoin[1], originally intended that Bitcoin be paired with a marketplace, as evidenced by beginnings of a market framework included in early snapshots of the Bitcoin codebase.[2] The market related code was eventually stripped out however, presumably as he decided to focus first on creating his world-changing currency. The concept of a decentralized marketplace in itself is not novel, there have been a small set of academic constructions and even serious attempts at creating them.[3], [4], [5], [6] They either propose solutions that scale extremely well and neglect the privacy implications, or they propose very privacy conscious solutions that do not scale well. Privacy and efficiency are often at odds with each other, "to hide the signal there must be noise". [7] Tor exemplifies this well, the traffic is pushed through various nodes with several layers of encryption before arriving at its destination, it is deliberately inefficient but the privacy provided by the trade-off is well worth it.

A. Bitcoin

Understanding the high level architecture of Bitcoin's blockchain is a prerequisite, more specifically the structure of (multi-signature) transactions and in some degree the accompanied script signature language. Bitcoin is a marvelous invention but nonetheless the link-ability of transactions is worrisome for maintaining privacy in a completely public ledger. To re-iterate, the transactions and data they carry is public and one has to be more careful in comparison to the traditional model where only one or more entities are in control.

A more generalized approach is used in this document by utilizing the word blockchain, what follows is built on the structure provided by the Bitcoin blockchain but is equally applicable to any of it derivatives, meaning the marketplace is indifferent about the underlying cryptocurrency used for payments.

[1]I. Kaiser is with The Particl Project, Department Research and Development code at particl.org

II. HIGH LEVEL OVERVIEW

The overview consists of two main components: a blockchain and a data storage network. Technically speaking these networks can operate over the same set of nodes. But for clarity we separate them to highlight that it does not have to be the same set.

A. Blockchain

The blockchain is typically tasked with processing payments but for our purpose it will also be storing the marketplace index and the identities.

New privacy enhancing techniques have spurred in the wake of recent revelations surrounding corporate blockchain analysis firms and government surveillance. The most notable inventions obscure the origins, destiny and the amount of a transaction. Signature schemes like SNARKs [1] and RingCT prove ownership of an output without conveying which specific output was spend yet without infringing double spend prevention.

B. Data Storage Network

The Data Storage Network (DSN) is tasked with storing market listings as well as messages between sender and receiver and all other accompanied data (such as but not limited to images and videos). Sensu stricto the market listing references, stored in the blockchain, are also protocol-agnostic, simply meaning it can handle different DSN protocols such as BitMessage, IPFS, HTTPS... It is however advisable to use one main protocol to prevent segmentation from occurring, which can potentially create interoperability issues between clients and may negatively impact privacy due to a smaller sized network.

BitMessage protocol[8] is used as an example because the message mixnet is simple to understand yet theoretically offers resilience against powerful attackers who aim to undermine privacy by analyzing traffic. A message traveling across a BitMessage network does not explicitly reveal any metadata as to whom might be the sender or receiver. Other protocols such as BitTorrent, IPFS, Kademlia and other DHTs always leak some information about which node is requesting which data. The BitMessage approach

[1]SNARKs are technically not just a signature scheme, but for the sake of simplicity and in the light of transaction signing it is reduced to merely that.

of ”everyone stores a copy of everything” is not a viable method for markets with storage requirements that exceed the performance of an average computer.

III. BLOCKCHAIN

A. Market Index

The blockchain stores references to market listings, the actual data is delegated to the DSN. The ’protocol id’ specifies which DSN protocol has to be used to gather the data. The ’listing id’ is a unique identifier used to retrieve the content from the DSN. More specifically it is the hash of the data to be retrieved, serving as an authentication mechanism against the DSN, ensuring the integrity. The hash must abide the MultiHash format, such that a multitude of hash functions can be used.[9] A public key ’item pk’ is attached and used to sign the listing registration transactions, this allows for multiple listings to be grouped to one item and creates a window of opportunity for future integrations such as reputation or reviews[5]. It is worth noting that the term ’data’ in this paragraph refers to encrypted(content), where encryption is handled by the software interacting with the DSN. Technically encryption is not required for a public marketplace, since anyone can decrypt the message it is rendered obsolete. Note that the registration transaction must be signed for ’item pk’, this provides authentication and prevents adversaries from adding listings to ’item pk’ without having the corresponding private key. See ’Index or constant reference’ in the appendix.

TABLE I
FORMAT OF THE REGISTRATION OF A LISTING

OP_RETURN	OP_REGLIST	item pk	listing id	protocol id

A transaction using the above mentioned script signature is unspendable therefore the output is also not added to the UTXO database due to OP_RETURN. The fee however is available to the actors that ensure consensus, miners in the case of Bitcoin. The fee determines the duration for which a listing is active, serving as a mechanism to expire and automatically garbage collect references. The inputs of the listing registration transactions would reveal the financial history of whoever initiated the market listing thus a payment scheme which obfuscates the origin is required to maintain privacy. Another optional approach could be to require a short PoW phase instead of a fee, including an adjustable difficulty arguably like mining blocks, this prevents network spam yet has the additional benefit that it eliminates any potential trace to the financial history of whoever created the listing. To maintain interoperability with Bitcoin derivatives, we did not make this the primary option. See ’input correlation attack’ in the appendix.

If only the item public key is present in the deletion transaction then all of the listing id’s for that respective item public key will be deleted. If additionally a listing id is specified only that instance will be removed. If additionally a protocol id is specified then only the protocol for that respective listing id will be removed.

TABLE II
FORMAT OF THE DELETION OF A LISTING

OP_RETURN	OP_DELLIST	item pk	listing id	protocol id

Note that OP_REGLIST and OP_DELLIST are ’virtual’ opcodes, they are always after OP_RETURN, meaning there is no requirement to implement new opcodes into the scripting language of the blockchain it is operating on.

This draft does not yet include information pertaining the details to how categories should work. Ideally this is also stored in the registration transaction, allowing to categorize listings without putting an unnecessary stress of the DSN network. Performing searches, especially full text ones, over decentralized networks remains a hard problem.

B. Private payment scheme

The transparent nature of the Bitcoin blockchain can potentially give away a trophy of information about the finances of the transaction creator. Therefore a payment scheme such as RingCT is required to shield the privacy of all users. It is worth noting that the payment scheme must provide hidden amounts to prevent a blockchain analysis technique that is described in the appendix of this document.

C. Identities

The current system does not support linking items to an identity. In other words, buyers have no way to see what items one specific merchant has for sale. The primary reason is to disable a wide category of passive analysis techniques that this could enable. The time at which listings are registered on the blockchain for example can reveal the timezone of the merchant given they have posted enough items to the network. Identities also aggregate data about the possible funds of the merchant when registering listings.

A more nuanced vision is required to balance this issue. Insinuating that the listings can not be linked to each other breeds false sense of security to the merchants. Time, image and linguistic analysis can provide a crucial trophy of information to a passive adversary, generating a fairly unique fingerprint that is hard to reduce through software design. These are issues related to the input of humans and only can only be dealt with in a certain degree. The defacto removal of image metadata for example can greatly reduce the fingerprint, auto correcting words can provide improvements in linguistic analysis.

Branding and name awareness are a common practice in todays world and are vital to a good working of the market. It improves the overall level of trust as quality merchants can build long term relationships with their customers. Therefore having an identity system is a trade-off worth having, the economic benefits outweigh the seemingly small negative consequences to privacy, given that you take the analysis techniques into account.

When a listing is registered to the blockchain is should actually follow the format as described in Table III. The reason for it being excluded from the table in the previous section was due to bad formatting.

TABLE III
FORMAT OF THE REGISTRATION OF A LISTING WITH IDENTITY

OP_RETURN	OP_REGLIST	identity flag	identity pk	item pk	...

Identities are required to be unique values, therefore an implementation can follow a format similar to how DNS systems operate. The first alternative cryptocurrency was Namecoin, a decentralized open source information registration and transfer system based on the Bitcoin cryptocurrency. [13] The blockchain is used to store all DNS records, this feature can be leveraged to store the identities of our users in a decentralized manner, but only if they wish to use the identity system. Implementing a DNS system also offers use a wide range of capabilities that is not available through simpler nickname system.

IV. MAD ESCROW

The 'Mutually Assured Destruction (MAD) Escrow' consists of a multi-signature transaction combined with OP_CHECKLOCKTIMEVERIFY in such a way that it will destroy access to all funds within the transaction after a certain interval. The script signature of the outputs from the MAD escrow transaction will drop all the public keys beyond a certain blockheight. Both merchant and buyer lose access to their funds, motivating both to find equilibrium before doomsday.

A. Ultimatum Game

In the Ultimatum Game, first studied by Werner Gth, Rolf Schmittberger, and Bernd Schwarze (1982), the proposer proposes how to split a pie between herself and a responder. Then the responder decides whether to accept or reject this proposal. If the responder accepts, then the proposal is implemented; otherwise, both players receive nothing. [10]

B. Ultimatum Game Differences

Unlike traditional Ultimatum Games, there is room for repetition of offers creating a more complex interaction and negotiation procedure. Another difference is that in this model the money is sourced by the players, this adds extra psychological factors to the decision making. People are likely more emotionally attached to money that they have worked for, and will value it more than money they get by being lucky and for free. They will thus react differently than in the traditional game. You'd react different about losing $100 from your wallet than someone coming to you and saying "I was going to give you $100, but I did not". Under the assumption that the players approach the problem in a purely rational sense then they do not make a distinction between a loss of their own money and a loss of potential gains. The net result is that we've lost that value, whether it was previously in our wallet or expected to be given to us.

C. Insurance deposits

Both merchant and buyer deposit an insurance amount into the multi signature escrow address. The insurance deposit is a percentage of the sale price, throughout this paper we will assume that this ratio is 1 (100%). This concept is more easily explained by an example: a painting is listed for $100, the insurance ratio is 80%. The merchant would make an insurance deposit of $80, the buyer would deposit $100 (payment) and $80 (insurance). If both actors are honest and the sale went through, then both will agree to each return the insurance deposit ($80) and the merchant receives $100 as payment. The insurance put up by the merchant introduces a risk to being dishonest.

This concept was first used by BitMarkets, this paper however proposes a small modification. Splitting the payment and the insurance deposit into two outputs, only time locking the output for insurances. A fraudulent merchant who has not shipped an item has leverage over the buyer, since the merchant has less money in the escrow and thus has more to lose. The segregation of payment and insurance allows the buyer to punish the merchant, both losing the insurance deposit. Any of the actors can punish the other party without making any decision about the payment output just yet. This has the potential benefit that an actor has more incentive to punish the other if they act fraudulent since the stakes seem smaller. More specifically the amount the buyer loses at that moment is smaller (in comparison to also losing the payment output) and thus may promote faster punishment.

V. DATA STORAGE NETWORK: BITMESSAGE

A. Introduction

BitMessage is a decentralized messaging network which we can leverage to host listings and to perform the communication between merchant and buyer. All nodes participating in the network store all messages of the past 48 hours, this means anyone in the network could have been the sender or receiver. The benefit of the decentralized topology is that there are no direct ties between the messages and any IP addresses. Nodes must try to decrypt all incoming messages to check if it was destined for them. A feature called 'streams' was proposed in the BitMessage protocol, aimed at increasing scalability but we will conveniently leave this out of our scope.

B. Metadata

A message traveling on the BitMessage network does not include metadata that can reveal who the receiver or sender is. Only the encrypted payload, IV, HMAC and temporary public key are public, the receiver of a message is the only able to decrypt the message and only for them the HMAC will properly verify.

C. Weaknesses

The cryptography behind BitMessage is simplistic, there is only one key for decryption meaning an adversary can read all future and past messages once in possession of said key. The protocol in itself does not provide perfect forward secrecy, nor perfect future secrecy. These two features should not lack from an end-to-end encrypted messaging solution with a completely decentralized topology, mainly because adversaries can collect all messages and store them indefinitely. We can't erase the encrypted messages from third parties therefore at least it should use a proper key ratchet and delete the private keys to decrypt the messages when they're being deleted. Another issue is that the curve secp256k1 is hard to properly do in constant time. BitMessage solely relies on OpenSSL to do the cryptographic lifting for them. In recent years Bitcoin Core has developed their own faster and potentially more secure library libsecp256k1, removing the dependency on OpenSSL for the most part. Another concern is the fact that BitMessage (at the time of writing) still uses SHA1 as checksums with their ECDSA signature. SHA1 is a hash function for which the first collision has been found.[11] The author of this paper has disclosed this but the maintainer of the PyBitMessage implementation was already aware and quickly replied with an upgrade plan, a move to SHA256 seems to be planned in the near future.[12]

D. Improvements

E. Dual-key stealth address

The BitMessage protocol can be extended to work with dual-key stealth addresses, the scan key can be used to authenticate the HMAC while the spend key is used to generate the shared secret for encryption and decryption. This would allow for Simple Message Verification (SMV) clients, where a scan private key is shared to a trusted party, which is then able to scan for messages belonging to the user, but the untrusted party will not be able read the contents. This approach does partially reveal the dual key stealth address, a full address can be pieced together given that it is public. The chance that two stealth addresses share the same private key is negligent. Uniqueness of scan keys is not a property that is enforced, thus decoy addresses can be created, sharing the same scan key.

F. Future and forward secrecy

A necessary improvement would be to provide perfect forward secrecy (PFS) and future secrecy for private messaging. This would be possible by treating BitMessage's encryption solely as a transport layer and instead relying on encryption libraries such as libsignal and noise to encrypt private messages. Another option is to implement the Signal key ratchet into the core of the BitMessage protocol, essentially creating a more optimal solution.

G. Blockchain key distribution

The blockchain can store and link public keys to an identity (such as a nickname), providing a tremendous improvement over traditional key distribution servers. An adversary would have to perform eclipse attacks or a consensus attacks (51% mining power for example) to have a chance at spoofing a public key for a given identity. The blockchain is traditionally stored locally, preventing information leakage of key look ups. The act of requesting a key from a key distribution server in itself reveals that you're interested in communicating with an individual.

H. Improving efficiency with RMIDs

Increasing the efficiency of the BitMessage protocol is possible by including a 'ratchet message id' (RMID) in the encrypted message. Alice includes RMID_1 (hidden) in the message she sends to Bob, when Bob wants to reply then he will publicly include RMID_1 in the header of the message. This allows nodes to scan all messages for known RMID's (without performing any decryption) and those matching known RMID's will be the only ones they will attempt decrypting. This improvement can potentially be integrated onto the existing BitMessage infrastructure by prefixing the HMAC hash, therefore allowing backwards compatibility. RMID's can also be used, instead of scan keys to provide SMV functionality.

VI. MARKET PROTOCOL

The actual protocol falls outside the scope of this whitepaper. The protocol specifications however are available on GitBooks. https://public.etherpad-mozilla.org/p/WlHuAPoWxO

VII. CONCLUSIONS

The marketplace as proposed in this paper provides an extensible framework that will operate on any Bitcoin-based blockchain and allows for a multitude of data storage network solutions to be utilized. The field of decentralized data storage is constantly expanding, it seems wise to not commit to one protocol when there are so many new innovations spurring everywhere, hence a generic approach was adopted.

A. Future research: Data Storage Networks

One DSN was discussed, namely 'BitMessage' because it does not leak details about lookups (what listings you're viewing) to other nodes. A extended research will be the comparison of different solutions for data storage such as BitMessage, Kademlia, IPFS, MaidSafe etc. We chose to discuss BitMessage in this paper because the system design intuitively seems like the most privacy protective. A formal academic backing to this claim is planned as future research. Scaleability however does remain a concern hence the reason why DHTs are a necessity. Private messaging on any DHT does remain a challenge, the receiver needs to be made aware of the hash of the message to be able to retrieve it. BitMessage and RMIDs can help solve this issue by linking RMIDs to hashes that can be retrieved on a DHT.

B. Future research: Private Payment Schemes

A wide range of academic research into payment schemes that provide privacy guarantees have been released in 2015 and 2016. None of them have yet achieved the holy trinity of providing (a) an anonymity set that is every previous transaction on the blockchain, (b) a trustless setup and (c) a highly scaleable network.

APPENDIX

C. Input correlation attack

Due to obvious structure of a MAD escrow transaction it is possible to divide all transactions in two categories: (a) normal transactions and (b) transactions that purchase an item on the marketplace. The second category possesses an interesting characteristic for blockchain analysis, more specifically merchants are more likely to use these outputs for creating new listings. This is not an odd perspective if we operate on the assumption that the venture is profitable and the they minimize the cashflow going in and out from the cryptocurrency. For example a ring signature with multiple potential spenders is used to fund a new listing, it is more likely that merchants will be using outputs coming from transactions of type b. In simple terms, the funds from previous sales are more likely to be used to create new listings. A naive coin input selection could aid blockchain analysis tools to link financial transactions to identities, given that we have an aggregation of listings that we know belong to one identity. This is can be avoided by using a different way to mitigate spam, a PoW proof for example, eliminating any traces to the finances of a merchant.

D. Amount correlation attack

If the amounts of transactions (type b, see previous section) were to be public then it would be easy for a passive adversary performing blockchain analysis to link these transactions to their corresponding listing simply by matching the amount to the active listings on the network.

E. Index or constant reference

This paper proposes using the index of the inputs to determine which of the transaction signers is the item key in favor of the more simple and static approach of 'input number 1 is always the item key' or the even more inefficient approach of duplicating all 32 bytes of the public key in the op_return. This allows more room for future developments surrounding the idea of reputation and review. A further improvement for example could be to have multi signature inputs for a registration transaction where a third party is involved acting as an extra insurance for reputation. This would essentially allow for a 'network of trust'.

REFERENCES

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf

[2] S. Nakamoto, Strip out unfinished product, review and market stuff, https://github.com/bitcoin/bitcoin/commit/cc4b78d59f566ff43881f57797a16ce45eb1b8

[3] OpenBazaar, https://openbazaar.org/

[4] S. Dekorte, BitMarkets, https://voluntary.net/bitmarkets/whitepaper/

[5] K. Soska, A. Kwon, N. Christin and S. Devadas, Beaver: A Decentralized Anonymous Marketplace with Secure Reputation, IACR, https://eprint.iacr.org/2016/464.pdf
[6] M. Max, DropZone, https://github.com/17Q4MX2hmktmpuUKHFuoRmS5MfB5XPbhod/dropzone-lib/blob/master/Drop%20Zone%20-%20Whitepaper.pdf
[7] A quote I made up, inspired by G. Maxwell
[8] J. Warren, BitMessage, https://bitmessage.org/Bitmessage%20Technical
[9] MultiHash, https://github.com/multiformats/multihash, 2017.
[10] S. Andersen, S. Erta, U. Gneezy, M. Hoffman, and J. A. List, Stakes Matter in Ultimatum Games, http://rady.ucsd.edu/faculty/directory/gneezy/pub/docs/ultimatum_aer_published.pdf
[11] Google, Announcing first SHA1 collision, https://security.googleblog.com/2017/02/announcing-first-sha1-collision.html
[12] Peter Surda, BitMessage: Move from SHA1 to SHA256, https://github.com/Bitmessage/PyBitmessage/issues/953
[13] Namecoin, https://namecoin.org/docs/faq/