BitMarkets White Paper

Introduction

Bitmarkets is an open source protocol and free client for a decentralized marketplace which uses Bitcoin as its currency and Bitmessage as its communications network. Mutual security deposits between buyer and seller on individual transactions are used to ensure incentives are aligned for completing market transactions without the need for either reputation systems or 3rd party escrow agents.

This document is a non-technical overview of how the Bitmarkets client and protocol work. For more technical details, see the Bitmarkets protocol specification.

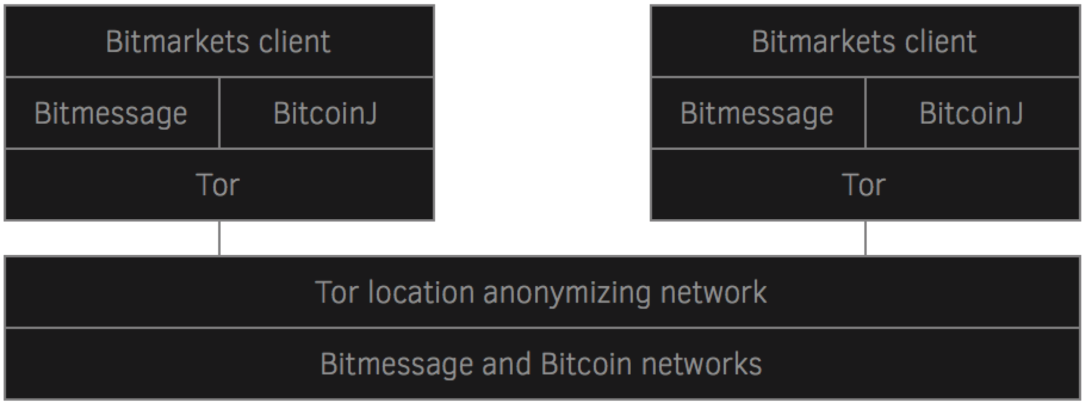
Motivation

Existing centralized marketplaces and payment services extract high fees, impose and abuse excessive control and remove any hope of privacy from users. As more commerce moves online, many consumers may find their lifetime history of purchases (including books, personal items and location details) for sale to advertisers, employers, curious neighbors, stalkers, political opponents and government agencies.

An ideal system would be one of secure private transactions directly between buyer and seller without middle men collecting data or adding fees. Such systems are now feasible by combining recently developed technologies for anonymous decentralized payment and messaging systems.

Implementation

Architecture



Terms

Bitcoin - an anonymous p2p digital currency
Bitmessage - an anonymizing encrypted p2p messaging network
Tor - a communication network that conceals user's locations
Bitmarkets client - an application used to buy/sell on Bitmarkets

Client

To use the marketplace, a client application which implements the Bitmarkets protocol is needed. The client is used to post, browse and execute transactions.It also hosts a Bitcoin wallet that handles security deposits, payments and refunds. A working client is available at: https://voluntary.net/bitmarkets.

Identities

The client uses Bitmessage (a decentralized identity hiding messaging network) and Tor (a communications source hiding network) to broadcast sales and exchange transaction related messages. Upon starting the client, a Bitmessage identity (a public encryption key) is generated which, like an email address, will be used for communications on the network.

Depositing Funds

In order to complete a transaction, both buyer and seller must have sufficient funds in their wallets for escrow and payment. The client contains a Bitcoin wallet in which these funds can be deposited and withdrawn.

Posting Sales

Using the client, a seller specifies the region, category, title, description and price for the sale and broadcasts this to a specific Bitmessage channel. These posts are then relayed throughout the Bitmessage network where other clients see them and make them browsable by buyers. Each client keeps its own copy of the posts. There is no centralized database of postings or central servers through which they pass.

Requesting a Purchase

Using the client, buyers can browse regions and categories to find products/services for sale. If/when they choose an item to buy, the client requests a purchase by sending a bitmessage to the seller.

Accepting/Rejecting a Purchase

If the seller accepts a purchase request, an accept message is sent to the buyer and the client automatically sends reject messages to any other purchase requesters.

Escrow Lock

When the seller approves the purchase, the seller's client constructs the first part of the escrow lock transaction which contains their security deposit and sends this to the buyer's client. The buyer's client then adds their own Bitcoin inputs for payment and the buyer's deposit, signs the transaction and sends it to the seller to sign and broadcast the completed transaction to the Bitcoin network.

Neither party's funds are locked (unavailable for spending) until the transaction is submitted to the Bitcoin network. If either party fails to complete the escrow message exchange or submit the completed escrow to the Bitcoin network, either can cancel the escrow by sending their own Bitcoin inputs for the transaction to another address they own. Cancellation is only possible if the completed escrow transaction has not been sent to the Bitcoin network first.

Delivery

Once escrow lock is complete, the buyer's client prompts them for delivery details for the order. These are then sent to the seller for delivery.

Escrow Release

Upon receiving delivery of satisfactory good/service, if the buyer chooses to initiate payment, their client constructs a transaction which releases the deposits to their respective parties and sends the payment to the seller in a similar process to how the escrow was locked.

If the buyer does not receive the good/service or finds them unsatisfactory, they can request a refund from the seller. The refund request constructs a transaction which sends the deposits to their respective parties but returns the payment to the buyer. If the seller approves the refund, the transaction is completed in a similar process to how the escrow was locked.

Notes on Escrow and Reputation

Choice of Two Party Escrow

To appreciate the value of the two party escrow system and why it is so important for secure, private and middle men free exchange, it is worth considering problems with the traditional approach of using a 3rd party escrow agent.

In a third party escrow system the buyer and seller ask a third trusted party (an escrow agent) to receive payment from the buyer before the item is delivered and to then release the payment to the seller after the item is delivered. If there is a dispute about the quality of the item or whether it was delivered, the third party acts as a mediator and decides whether to send the payment or refund the buyer. Some of the problems with this model include:

1. Unless escrow agent receives the item before forwarding it to the buyer and has the expertise to verify the quality of the item, they have no means of fairly mediating a dispute. This problem becomes much worse when services are being exchanged. Proof of package delivery doesn't verify what was delivered or it's quality and photos from the buyer could be of another item, etc.

   Credit cards and centralized marketplaces seem like existence proofs of working escrow agents but they are non-anonymous, do not resolve disputes fairly (as most small merchants can tell you) and are known to break in markets with high fraud potential (e.g. third world countries, porn, etc).

2. Untrustworthy escrow agents who may steal escrow, fail to release escrow or collude with (or be) one of the parties.
3. Escrow agents with strong reputation may be doing a long con to take off with or extort a large payment or group of payments.
4. Escrow agent expenses in the form of labor and risk are passed on to the buyer and seller.
5. Finding a trustworthy agent requires a reputation system and building reputation systems in anonymous networks remains an unsolved problem (see next section).

Fortunately, two party escrow made possible by Bitcoin's multi-signature transactions can solve all of these problems.

Where Reputation Systems Are Required

For transactions involving goods delivery of the size typical for a users cash transactions, we suspect two party escrow will be sufficient to address the problem of trust. But there are classes of transactions where reputation systems seem to be required. Some examples include:

1. Larger transactions such as those of the size that typically involve debt (e.g. purchasing a car), where the necessary size of security deposits for two party escrow may be prohibitive.
2. Transactions involving services where buyer and seller meet in person (e.g. taxis) pose a risk that either party may physically coerce the other into releasing the transaction or otherwise harm them.
3. Transactions where there are significant cost for non delivery beyond the price of the item (e.g. urgent medical supplies and services, critical parts, deadline driven projects).
4. Transactions where quality is difficult to objectively assess or where the client may not have the expertise to assess it (e.g. professional services).
5. Transactions where a buyer has strong concerns about exposing their physical address to a potentially malicious seller.

Reputation System Failures

Although reputation systems are useful, there are many potential ways for malicious agents to attack them. Systems that do not address these may be worse than no reputation system at all as they can result in users extending trust to untrustworthy parties or unwittingly exposing their real identities. These attacks include:

1. Build false trust: Posting fake positive reviews by creating many identities and performing transactions and posting reviews between them.
2. Build false distrust: Posting fake negative reviews of competitors.
3. Long cons: completing in good faith many transactions in order to accumulate and then exploit trust extended for a large or large group of transactions.
4. Trusted parties whose computers become compromised. This is not a threat with two party escrow as economic incentives are aligned regardless of the trustworthiness of the parties.
5. Identity leaks via delivery: Attaching reviews or history information to identities in externally verifiable ways can leak information (e.g. timing analysis by combining real world delivery records with online

transaction and review times), particularly when gathered over time.

6. Identity leaks via network analysis: Combining known real world connection data (e.g. social network, business or friend networks) with the reputation system's network of reviews and even a single exposed real identity may provide strong hints to the real identities to other relatively direct connections.

While the Bitcoin blockchain and cryptographic signatures could provide a means of externally verifying trade history, when trades where done and whether or not they were completed, it doesn't address the fake review problem and it makes the system potentially more vulnerable to identity leaks. As far as we know, there are no existing reputation systems that sufficiently address any of these issues.

Trust Network Challenges

Network of trust solutions, where users keep contact lists of other users they trust and potentially share reviews between them solve the fake review problem. However the trade off is having fewer available reviews and that comprising one user can leak the identities of all members of their trust network if the cryptographic identities are tagged with real names.

This problem can be addressed if the client only allows the tagging of trusted reviewer's cryptographic identities with the user's trust level in them. This also requires a more proactive role for the user in requesting these identities from those they trust and doing so in a way that does not record the connection between cryptographic and real identities.

This hidden real identity network of trust system is the one we will likely include in a future version of Bitmarkets. The decision not to include it in the first version was made because it would significantly extend the development time and because two party escrow makes this added complexity unnecessary for a large set of transactions which could be immediately valuable to users.

Choice of Escrow Ratios

The first version of the Bitmarkets client is configured to use a 1-1 deposit/price ratio. So for the sale of an item costing X, the buyer would put up 2X (1X for payment and 1X for deposit) and the seller 1X so a total of 3X would be locked in escrow.

While later versions will may allow this to be adjustable per transaction, the current fixed ratio was chosen to meet the requirements that:

1. Deposits are large enough that consistently malicious buyers or sellers will on average go bankrupt before destroying the market assuming their bankroll is small compared to the volume of the market.

2. Deposits are large enough incentive to ensure seller will deliver an acceptable product and that buyer releases escrow at end of transaction (signing a payment or refund request).

3. Deposits are small enough that parties are willing to accept losses in order to punish bad actors. The results of the Ultimatum game experiments provide evidence that this is likely for the sizes in which the people might make cash purchases.

4. Deposits are small enough that locking the funds over the period of the transaction wouldn't be a burden for either party in the sizes in which buyer might make cash purchases.

5. Simple enough to be easily understood by users.

The 1-1 ratio seems to be the simplest choice which meets all of these requirements.