Marker addresses: Adding identification information to Bitcoin transactions to leverage existing trust relationships

Jan Vornberger

jan@uos.de

Abstract: This paper proposes a technique for creating Bitcoin transactions enriched with special "marker addresses"[1] to make it possible to easily identify the originating party and allow for special treatment by the recipient. In particular this technique can be used to implement secure zero-confirmation transactions between trusted parties without requiring an additional communication channel besides the Bitcoin network.

1 Introduction

First announced in 2008 [Nak08b], Bitcoin [Nak08a] is an emerging digital currency. It is completely decentralized and as such no central clearing place exists and instead the Bitcoin system solves the double-spend problem (spending the same coin twice with two different parties) through the application of a proof-of-work mechanism. The nature of this mechanism introduces a delay of up to an hour before a transaction can be considered completely confirmed. While there are a number of applications where such a delay is not an issue, many other situations would benefit from some type of secure "instant" transaction with no risk of double-spending. In the case of Bitcoin, a double-spend can only be attempted by the original sender of a transaction, as she is the only one with the necessary private keys. Therefore the risk of a double-spend becomes a question of the trustworthiness of the original sender. This paper therefore proposes a technique which allows the sender to identify herself to the recipient, to be used in cases where a preexisting trust relationship can be leveraged to allow for instant Bitcoin transactions.

Of course the sender could inform and prove her identity to the recipient using some other communication channel. In fact, in the long run Bitcoin would probably benefit from some type of common versatile out-of-band payment protocol which would be used to negotiate details about a Bitcoin transaction – possibly including sender authentication – before the actual transaction takes place. However, specifying such a protocol will not be the topic of this paper. Instead, a stop-gap solution is proposed, which works completely in-band – i.e. using only the Bitcoin network.

This has a number of advantages: Bitcoin transactions are very easy to accept, as giving out a Bitcoin address and listening for new transactions only requires an out-bound connection

[1] In previous writings about this concept, I have referred to this idea as "green addresses" [Vor11b]. As this term also evoked unrelated associations of environmental topics, this paper will instead use the term "marker address".

to the Bitcoin network. It therefore can also work behind firewalls and does not require a domain name or static IP address. The proposed technique preserves these features. Furthermore it does not require any changes to the Bitcoin protocol and is backwards compatible to existing infrastructure. This makes it very easy to deploy.

2 Technical background

This section will give a brief overview of some of the aspects of the Bitcoin system that are relevant for the technique proposed. See [Nak08a] and [Wikc] for a description of the complete system.

A central component of the Bitcoin system are Bitcoin transactions. They consist of a number of inputs and a number of outputs. Each input references an output of a previous transaction. Following these chains back to their origin will always end up at a special type of transaction, called a generation, which is allowed to have no inputs and a single output.

Furthermore, each output has a number attached to it, which describes the amount of Bitcoins this output is worth. A transaction is worth the sum of all the outputs referenced by its inputs. This total is then in turn distributed among the outputs of this new transaction. Anything that is left over – that is the sum of the referenced outputs minus the sum of the newly specified outputs – is the fee associated with the transaction.

All this is tied together by the next important concept: Bitcoin blocks. A block bundles a number of transactions and adds (among other things) a reference to the previous block and a nonce. The nonce is, in a computationally expensive process, chosen in such a way, that the hash of the block has an easy to check property. Namely, that it starts with a particular number of zeros or, in other words, is smaller than a given target value. This makes the creation of blocks difficult, as the best known way of finding a suitable nonce is to try a lot of them in a brute-force fashion. Yet given a block, it is easy and fast to verify that the hash fulfills the requirement. The difficulty of this process can be adjusted by lowering or raising the mentioned target value. All Bitcoin nodes follow the same adjustment algorithm, which aims to keep the creation of new blocks at a specific rate, regardless of how much processing power is network-wide working on finding the next suitable nonce.

This mechanism constitutes the proof-of-work security of the Bitcoin network, as this chain of blocks – all the way back to the genesis block – represent the shared understanding of all Bitcoin nodes as to which transactions are valid and have already occurred (see figure 1). Specifically a node will consider the longest[2] such chains to be the valid one, should the situation occur that two blocks both point to the same predecessor and as such create a fork. On one hand this can happen naturally in cases where two new blocks are created in quick succession so that the creators are not aware of each other and as such both point to the same predecessor. On the other hand it can occur if an attacker attempts to rewrite

[2] Where, technically, length is calculated as total combined difficulty of that chain.

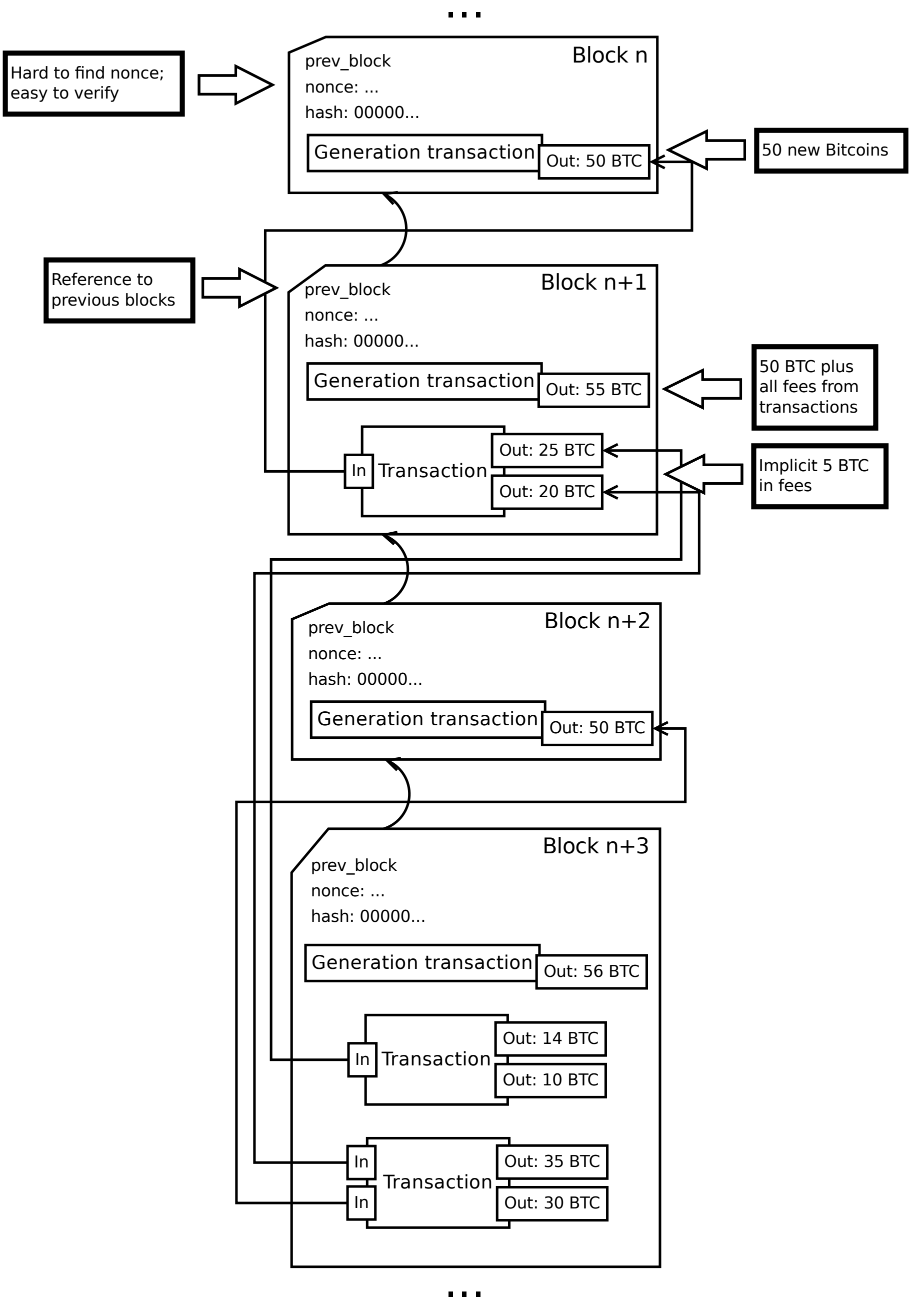

Figure 1: Simplified structure of the Bitcoin block chain.

history by forking the chain at some point and creating a longer alternative. This becomes harder for the attacker the further back from the end of the chain he wants to create the fork, as he needs to create new blocks to catch up to and then surpass the length of the current chain, which continues to be extended. For this reason a transaction becomes very hard to reverse as soon as it is buried under a number of blocks.

Blocks also form the source of new Bitcoins, as the very first transaction in every block is allowed to be a generation transaction. Besides creating 50 new Bitcoins (currently) the generation transaction also includes all fees attached to the other transactions included in the block. This incentivizes the creation of new blocks – also called “mining” – and rewards the creator of the block for contributing processing power to secure the network.

Finally, to allow ownership of specific Bitcoins, outputs have additional details attached to them, which specify the requirements of using this output as an input in a new transaction. To verify them, the details of the output as well as the new input are combined to form a script in a simple programming language which is then executed on a virtual machine to yield the result of this check. The details of this script language will not be covered here, but it does allow to create a variety of different types of transactions.

One common type of script, which is also the focus of the technique proposed in this paper, ties an output to a single Bitcoin address. A Bitcoin address is derived from the public key of a public/private cryptographic key pair. To use the output as an input in a new transaction, the input needs to contain the public key matching the Bitcoin address referenced and be signed by the matching private key. Using this construct, only the owner of the Bitcoin address – the person who knows the associated private key – is able to use the output and move the Bitcoins to, for example, a new Bitcoin address.

Given a Bitcoin address, one can find all open (not yet used) outputs which reference the address in this manner. The total sum of these outputs can be thought of as being “in possession” of this Bitcoin address.

In summary, Bitcoin pulls together concepts from peer-to-peer networks, cryptographic signing techniques and proof-of-work mechanisms to create a fully decentralized network, which works like a public accounting system that tracks movements of coins between Bitcoin addresses. Participants do not have to trust any other participant in particular, they just have to trust that at least 51 % of the processing power behind the proof-of-work mechanism is acting fair. In that case, two participants can be sure, that their transactions with each other are recorded faithfully in the block chain and that they both see the same information.

3 Marker addresses

The proposed technique is simple and works as follows: The sender picks one of her Bitcoin addresses to be used as a way to identify her. This address will be called “marker address” and should be published somewhere. The sender ensures that a number of small coins - for example several 0.01 BTC - are in possession of that address, i.e. requiring the corresponding private key to redeem them.

When creating a new transaction, the sender proceeds as usual but adds a final step, in which an additional input and an additional output are added to the transaction. The input will reference one of the 0.01 BTC from the marker address and the output will simply send those 0.01 BTC back to that address (see figure 2). This means that the sum of the other outputs is not changed, but the transaction will now also include the signature from the private key of the marker address, as it is required to sign the additional input[3].

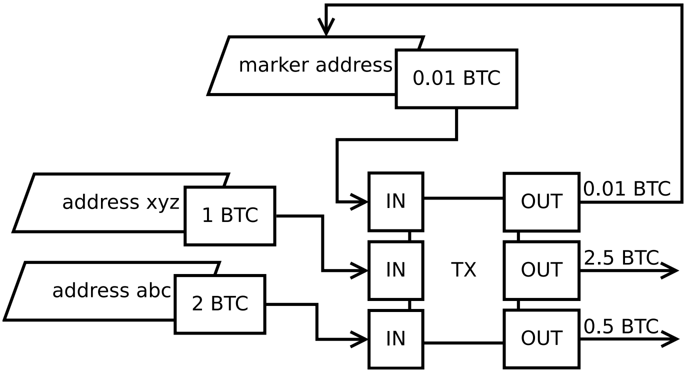

Figure 2: Transaction with a coin from a marker address.

This gives the recipient the ability to identify the sender by checking the inputs for known signatures. Depending on the trust relationship with the sender the transaction can then be treated accordingly. In particular, even an unconfirmed transaction – one that is not in a block yet – can then be accepted immediately, if the sender is trusted to let the transaction confirm eventually[4].

4 Advantages and disadvantages

As mentioned earlier, the main advantage of this approach is the fact, that it happens in-band and will therefore work in any situation where Bitcoin transactions are currently taking place. This makes it very easy to deploy and compatible with existing infrastructure, as it also does not require changes to the Bitcoin protocol.

On the other hand this approach brings with it a number of disadvantages. It encodes additional information about the sender into the block chain. This might be undesirable from a privacy point of view in some situations, as well as taking up space in the block

[3] Many thanks to Peter Cooper Jr., who suggested this improvement on the Bitcoin forum [Coo11]. The original proposal – then called green addresses [Vor11b] – simply suggested to have all inputs point to the marker address. This required to have the necessary funds available at that address and would therefore generally require additional transactions to ”top up” this address from time to time. This improvement avoids the need for additional transactions.

[4] In corner cases this might take a long time if – for example – not enough fees are attached to the transaction. In practice however, most of these problems are solved by periodically rebroadcasting the transaction – something that the recipient can do as well.

chain for information that is not relevant in the long term.

Because of these drawbacks, marker addresses might only be a stop-gap solution, until the Bitcoin community can agree on a versatile out-of-band payment protocol that works in all relevant situations (on the web, person-to-person, using QR codes, using NFC, etc.).

5 Implementation

I announced the original proposal – then called green addresses – in July 2011 [Vor11b] and provided a first implementation of sending such transactions as a feature on Instawallet[5], a browser-based wallet service, which I operated at the time[6]. To experiment with how receiving these transactions in a point of sale setting could work, I released a prototype of a point of sale system (see figure 3) in August 2011 [Vor11c] along with a video demonstrating it [Vor11a].

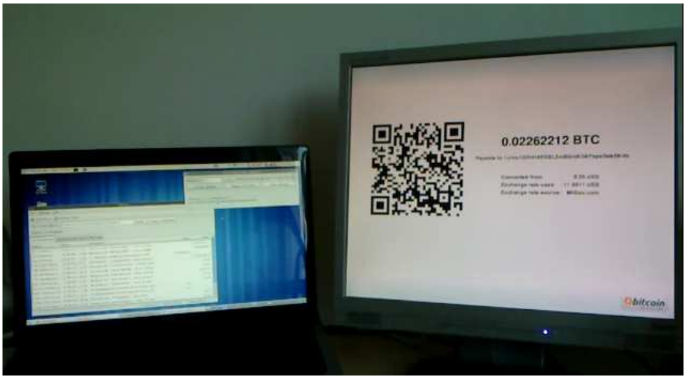

Figure 3: Point of sale demonstration setup.

In October 2011, Mt.Gox – the Bitcoin exchange with the largest volume [Cha] – decided to experiment with offering a green address option for withdrawals [Kar11]. On the receiving side, BTCPak[7], a service to exchange Bitcoins for MoneyPak, is one website with support for instant funding through the Mt.Gox green address. It started operating in February 2012 [DBo12a].

As can be seen by this timeline, marker addresses have only slowly and to a limited extend found their way into production systems. One reason for this might be the mentioned disadvantages of this approach, which some implementers will not find acceptable.

But I think the main reason is probably the fact, that there are not yet any ready to use software solutions. While this approach does not require changes to the Bitcoin protocol, it does require software support for creating transactions of this type and for checking for

[5] `https://www.instawallet.org/`

[6] Currently (June 2012) this feature is not available because of technical difficulties [iwf].

[7] `https://www.btcpak.com/`

the presence of marker addresses. These functions are not yet easily available in – for example – the mainline Bitcoin client. A fact which also complicated the introduction of marker addresses for BTCPak [DBo12b]. More work is required, to lower this barrier to entry.

6 Alternative approaches to instant transactions

There are certainly various alternative ideas regarding instant transactions, which can not all be covered in this paper. However, one particular idea, that is often suggested, is the attempt of early double-spend detection.

6.1 Early double-spend detection

Proponents of this approach argue, that double-spending usually has to happen quickly after the original transaction has been broadcasted. They argue, that if one would be able to detect such attempts during those critical first few seconds, it would then be safe to accept transactions right away after they passed the check. Satoshi Nakamoto also talked about this idea [Nak10]:

> I believe it'll be possible for a payment processing company to provide as a service the rapid distribution of transactions with good-enough checking in something like 10 seconds or less. [...] The payment processor has connections with many nodes. When it gets a transaction, it blasts it out, and at the same time monitors the network for double-spends. If it receives a double-spend on any of its many listening nodes, then it alerts that the transaction is bad. A double-spent transaction wouldn't get very far without one of the listeners hearing it. The double-spender would have to wait until the listening phase is over, but by then, the payment processor's broadcast has reached most nodes, or is so far ahead in propagating that the double-spender has no hope of grabbing a significant percentage of the remaining nodes.

It is important to know, that Bitcoin nodes – in their current implementation – will not accept or relay unconfirmed transactions that are in conflict with transactions they have received earlier [Wikb]. So the strategy in this approach is to monitor the network for conflicting transactions for a few seconds, while the original transaction has time to propagate throughout the network. Given this head start, one would then consider it safe to accept the transaction right away, as most miners will be working on including it into the next block. They will ignore later, conflicting transactions – at least as long as those are not part of a block yet.

This approach is very attractive, as it does not change anything about the transaction process itself, but instead works as a passive security check.

6.2 Attack: Colluding miner

However, this check will not help against double-spend attacks which do not broadcast the conflicting transaction, but instead try to embed it into the next block with the help of a colluding miner. Consider the following attack as an example of this[8]: An attacker wants to double-spend with a merchant who accepts unconfirmed transactions after they have passed the check described above. The attacker is furthermore in control of a pool that has 5 % of the global hashing power. Now, when the attacker creates the payment transaction A → B (where B is the merchant) and broadcasts it, she also creates a conflicting transaction A → C (where C is another of her addresses) which she keeps to herself, but tries to integrate into the next block her pool is working on. Should her pool be the one that finds the next block (chance of 5 %), the unconfirmed transaction A → B will be reversed. Otherwise the attack was unsuccessful and the pool will need to continue building on the latest block if it does not want to end up on a fork. As one can see, this attack will only succeed with a small chance. However, the attacker risks nothing in performing it and depending on the situation it might be possible to repeat the attempt.

It is clear that the feasibility of this attack very much depends on the payment circumstances. If the attacker can easily repeat the attempt, for example when depositing into an account where transactions from a failed attack can simply be withdrawn later, it is more of a problem than when buying a pizza where the attacker has to buy 20 pizzas to get one for free.

Yet, even for a pizza place it might not be so clear-cut. One might imagine that this merchant wants to forward the payment to a Bitcoin exchange for currency conversion right away. The exchange however will not accept unconfirmed transactions – for the reasons described above – and as such the merchant has to wait for confirmations here. This exposes him to more exchange rate risk which he has to price into the transaction.

6.3 Additional disadvantages

Besides the attack illustrated above, this approach brings with it the disadvantage of requiring to monitor the Bitcoin network in the described manner. This complicates the technological setup for the merchant or requires him to outsource this task to a service provider.

Finally, such a monitoring solution will necessarily involve an additional – even if small – delay. This delay needs to be long enough to check for conflicting transactions, but short enough to not impact the “instantaneousness” of the transaction – a difficult trade-off to make.

The proposed solution based on marker addresses – while it comes with its own set of disadvantages and requirements – has none of these problems, as it requires no monitoring setup and adds no additional delay.

[8] A variation of this attack is known as the Finney attack [Wikd].

7 Managing trust

Certainly one big burden the marker address solution brings on the recipient is the fact, that it requires him to manage - potentially a large number of – trust relationships. This is indeed a hard problem, as recent difficulties with the certificate authorities have shown [Lea11].

Yet, even given these difficulties, existing infrastructure could be used in tackling this challenge. It would be conceivable for a merchant to have the policy of accepting any marker address as long as it is associated with a website verified by an extended validation certificate and thereby leveraging this existing mechanism. Such a check could be performed by querying a neutral marker address database and then verify the certificate listed there by connecting to the site in question. Alternatively or additionally some form of a web of trust solution could be envisioned.

Yet another idea is to back marker addresses with security deposits managed by a trusted neutral party. The security deposit associated with a marker address would be paid to anyone who can prove a double-spend attempt involving this address (by presenting two conflicting transactions signed by the address in question). A merchant could query this database before accepting a new marker address to check the amount of the security deposit.

To reduce the number of trust relationships, some form of aggregation might be helpful. Hosted wallet services, like Instawallet, are in a good position to publish a single marker address and provide their users with the ability to use it with deposits that are already fully confirmed. Users could even retain some control over the deposits through the use of multi-signature transactions [Wika]: Requiring both the signature from the hosted wallet service as well as the user, would prevent the service from running off with the deposits, but would still allow for marker transactions to function.

Revocation mechanisms are another point in need of further discussion. It might provide useful to agree in advance on how to proceed in cases where a marker address is compromised. This could also happen in-band: Each marker address could be given an accompanying revocation address, which is only to be used for a dummy transaction that signals that the main address has been compromised. Alternatively, revocation might be handled by some announcement server out-of-band.

It remains to be seen which approach to trust management will work best. However, if some general agreed upon mechanism can be found, it would not only be beneficial to the marker address approach, but would most likely also be useful for any out-of-band solution.

8 Conclusion

This paper proposed a technique where coins from designated marker addresses are used in the construction of Bitcoin transactions to allow the recipient to identify a trusted sender.

As the risk of a double-spend is a question of the trustworthiness of the sender, this enables secure instant transactions between trusted parties in an easy to deploy manner.

However, the adverse effects on sender privacy and block chain space requirements make it clear, that this technique should only be a stepping stone to a proper out-of-band payment protocol. Yet, as a discussion of risk management of unconfirmed transactions showed, there are likely a number of situations where this approach could prove useful.

It does however bring with it the considerable challenge of trust management. Further research and experimentation in this area will be needed to find an acceptable solution that will work within the Bitcoin ecosystem.

Finally, more software development is required to go beyond the prototyping stage and lower the barrier to entry by providing software solutions, that are easy to deploy.

9 Acknowledgments

I would like to thank the numerous discussion partners within the Bitcoin community, who have helped to critique and improve upon this concept. Marker addresses, as described in this paper, are very much a product of all these discussions.

References

[Cha] Bitcoin Charts. Exchange volume distribution. http://bitcoincharts.com/charts/volumepie/. Last accessed on June 10, 2012.

[Coo11] Peter Cooper, Jr. Re: MtGox: Green address option. Bitcoin Forum thread, posting as user “pc”, October 2011. https://bitcointalk.org/index.php?topic=48170.msg581863#msg581863. Last accessed on June 5, 2012.

[DBo12a] User DBordello. BTCPak – Exchange your Bitcoins for MoneyPak. Bitcoin Forum thread, February 2012. https://bitcointalk.org/index.php?topic=64805.0. Last accessed on June 10, 2012.

[DBo12b] User DBordello. Using bitcoind to determine sending address to check for green address. Bitcoin Forum thread, February 2012. https://bitcointalk.org/index.php?topic=64547.0. Last accessed on June 10, 2012.

[iwf] Instawallet – FAQ. https://www.instawallet.org/static/faq. Last accessed on June 12, 2012.

[Kar11] Mark Karpeles. MtGox: Green address option. Bitcoin Forum thread, posting as user “MagicalTux”, October 2011. https://bitcointalk.org/index.php?topic=48170.0. Last accessed on June 10, 2012.

[Lea11] Neal Leavitt. Internet Security under Attack: The Undermining of Digital Certificates. *Computer*, 44(12):17–20, 2011.

[Nak08a] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[Nak08b] Satoshi Nakamoto. Bitcoin P2P e-cash paper. Announcement on “The Cryptography Mailing List”, 2008. http://www.mail-archive.com/cryptography@metzdowd.com/msg09959.html. Last accessed on June 3, 2012.

[Nak10] Satoshi Nakamoto. Re: Bitcoin snack machine (fast transaction problem). Bitcoin Forum thread, posting as user “satoshi”, July 2010. https://bitcointalk.org/index.php?topic=423.msg3819#msg3819. Last accessed on June 11, 2012.

[Vor11a] Jan Vornberger. Bitcoin - Point of sale system. YouTube video, August 2011. http://www.youtube.com/watch?v=o84SfChQ-S8. Last accessed on June 5, 2012.

[Vor11b] Jan Vornberger. Instawallet introduces new approach to instant payment: Green address technique. Bitcoin Forum thread, posting as user “jav”, July 2011. https://bitcointalk.org/index.php?topic=32818.0. Last accessed on June 3, 2012.

[Vor11c] Jan Vornberger. Release of open source point of sale system (w/ video). Bitcoin Forum thread, posting as user “jav”, August 2011. https://bitcointalk.org/index.php?topic=38893.0. Last accessed on June 5, 2012.

[Wika] BIP 10 – Multi-Sig Transaction Distribution. Bitcoin Wiki. https://en.bitcoin.it/wiki/BIP_0010. Last accessed on June 12, 2012.

[Wikb] Protocol rules. Bitcoin Wiki. https://en.bitcoin.it/wiki/Protocol_rules. Last accessed on June 11, 2012.

[Wikc] Protocol specification. Bitcoin Wiki. https://en.bitcoin.it/wiki/Protocol_specification. Last accessed on June 4, 2012.

[Wikd] Weaknesses – The “Finney” attack. Bitcoin Wiki. https://en.bitcoin.it/wiki/Weaknesses#The_.22Finney.22_attack. Last accessed on June 11, 2012.