Styx: Unlinkable Anonymous Atomic Payment Hub For Viacoin

Viacoin Dev Team

viacoin.org

Oct 14, 2016

Abstract

This paper presents Styx, an unlinkable anonymous atomic payment hub for Viacoin. In this paper we construct Styx which will provide strong anonymity. Our result is zero knowledge contingent payment proof. Styx is a payment scheme compatible with Viacoin protocol. This will allow participants to make payments faster and anonymously . Without any information leak that can lead to tell which payer paid during the epoch. A fair atomic exchange protocol which prevents theft of participants assets notably Viacoins.

The protocol combines off-the-blockchain cryptographic computations using Viacoin scripting functionalities and making use of security RSA assumptions, ROM and Elliptic Curve Digital Signature Algorithm.

Contents

1.1 Fungibility

One of the most important properties of a currency is fungibility. For example, gold is fungible because 1g of gold is worth 1g of gold. So the real question here is what leads to this property because if that property were to be missing, fungibility would break down.

Gold is fungible because you cannot distinguish one bit of gold from another. There is no concept of old gold, new gold, or soiled gold so therefor all gold has the same value by weight. If there were, the gold from some sources may be less valuable. It would introduce the ability to blacklist.

There are many forms of digital payment systems where fungibility is a major concern. For example, Paypal have the ability to freeze accounts and payments at their sole discretion based on their opinion about the source or destination of payments being made on their platform.

Centralized bitcoin payment providers like Coinbase often close down accounts because they believe payments are being received from, or being made to gambling sites. They can do this by looking at payment flows on the blockchain. Now if it were impossible to deduce where funds came from, or where they were being made to, it would be impossible to perform this kind of analytics and harm the fungibility of Bitcoin.

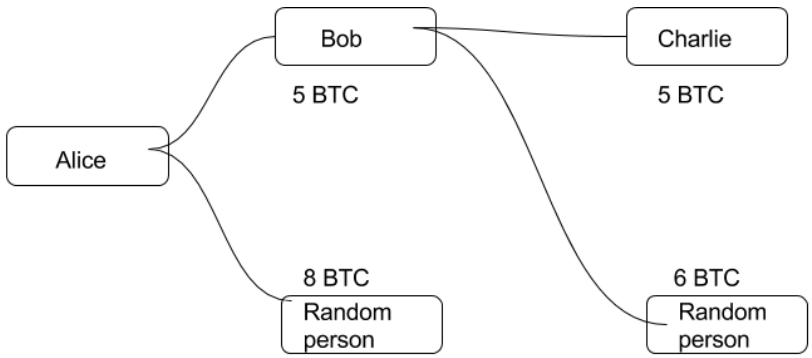
Because of the way payments are confirmed in Bitcoin like systems, it is, in theory possible for miners to blacklist specific coins from being spent at all. This was already tried in Ethereum after the infamous DAO hack. While some may view this as a legitimate case because it was viewed as theft, there is nothing stopping the same actions being taken against coins for other reasons.

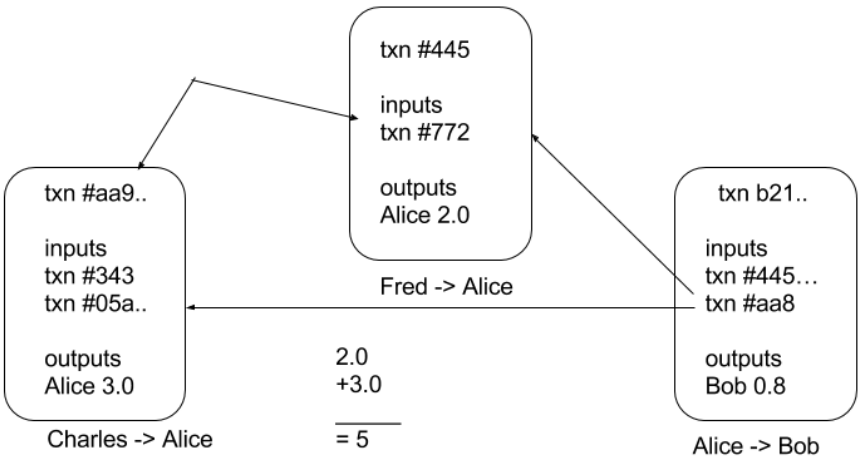
So the solution is to make payments completely unlinkable to participants, or at least, provide a level of obfuscation to such a degree that looking at the blockchain transaction flow yield no meaningful information. This would make all payments equal as no payment could be tainted by the past, nor tainted by where the payment is going to.

Let's dive deeper into the specific problem.

Alice has sent Bob an amount of 5 coins. She must reference a payment where she received 5 or more coins. The nodes will check if Alice was the recipient and that the coins add up to 5 or more coins. Person Bob later on receives those same coins. When someone even looks at the transaction of Bob and Alice, they can even track back that a the person Charlie send

coins to Alice.





Aside from the blockchain, nodes keep an extra database often called ***UTXO***. UTXO stands for ***Unspent Transaction Outputs***. The UTXO is a ledger that records funds available for every address. It works like a cache for the blockchain. When new transactions are made, the UTXO will be updated. Addresses can be linked to one another by checking the transactions between them. Comparable to having a public ledger of bank statements .There are several ways to track down

coins we present a few of them blow below :

-Transaction Graph and Off-network information:

Is a graph where transactions are placed and connect a transaction input associated with the transaction output.

An attacker can link an address by searching info about the user online and in various public databases they might have access to. Many users have their coin address public on their internet forums profile or social media. Users who may expose their address on a website or forum. An attacker could then combine the information gained from the network analysis with other information or private database they have access to. This could be exchanges, online wallet services etc.

-IP traffic relays:

An attacker can monitor the peer to peer payment network and observe which IP payments originate from.

-Cluster analysis:

A clustering behavior in the user graph can help to reduce the size of the user graph. For example, payments sent to exchanges or gambling sites might be trivially identifiable.

Flow and temporal Analysis: Large flows of funds can be traced through the network of users. A user can be deduced out if at some point they receive an odd large flow compared to their account balance or tainted coins.

It should be very clear by now that transactions can be linked to specific users or services and this is how fungibility is harmed. The current design of Bitcoin and cryptocurrencies based on Bitcoin do not have fungibility and every day as coin tracking becomes more advanced, fungibility is eroded.

To make **Viacoin** more robust and privacy-friendly it is essential to make sure Viacoin introduce fungibility to protect assets and increase user privacy and provide anonymous transactions between participants . It is our belief that only fungible digital currency will have a place in the future making this a top priority for Viacoin, which Viacoin users agreed with when surveyed .

A number of decentralized cryptocurrencies (e.g, Zerocash, Monero) have anonymous transactions implemented that are not compatible with Bitcoin.

Viacoin is based on Bitcoin and it requires a robust, safe anonymity solution for the existing code base.

1.2 eCash Centralized anonymous payments

Before revealing the Viacoin anonymous payment system, recall ecash. The pre-Bitcoin payment scheme. It used to rely on a bank and it wasn't decentralized.
It was a system to allow funds to be transferred anonymously, eCash used to charge sellers via paid fee.
eCash[1] used ***blind signatures***[2] . With this system, no link can be made between withdraw & transaction. The eCash system ensured that payments could not be linked between payer A and receiver B. The unlinkability was achieved with use of blind signatures.
A blind signature can be instantiated with RSA. Signer has the RSA *sk* (secret key). He also has the *sk* (secret key) and *pk* (public key). *pk, N* where *N* is the modulus. *G* as in full domain hash. *A* signs a message *sn*. *A* can produces blinded messages

$\bar{sn}$ by choose random blind r ← Z^*_N

$$= r^{pk} G(m) \bmod N^{\frac{-1}{RSA}}$$

Signer produces blind signature as:

$$\sigma = \frac{-1}{RSA}\ \text{sk} \bmod N$$

A can unblind signature as:

$$\sigma = \sigma / r = (G(sn))^{sk} \bmod N$$

This is how blind signatures are applied. This could be done with Viacoin if A sends *S* a viacoin and a blinded serial number $\bar{sn}$ and A obtains a blind signature *σ*. *A* unblinds the values to create an anonymous voucher *(sn, σ)* which A can use to pay person *B*.

[1] https://www.win.tue.nl/~berry/papers/cosic.pdf
[2] the message's content cannot be seen before it is signed

However this method is not robust. If *S* is malicious, he could steal by refusing to issue a blind signature to A in exchange for the received viacoins or refusing *B* anonymous voucher in exchange. This can be solved with using the Viacoin scripts that is able to enforce an atomic transaction mechanism. A swaps viacoin for a blind signature that's provided by *S* and in exchange swaps an anonymous voucher with *B*. An atomic exchange like this can be build with the Viacoin scripting functionalities.

2.1 Styx Fair Atomic Protocol

Viacoin wallet is to atomic mix payments from payers. Each atomic mix payment requires 3 confirmations. Nobody other than *Payer* A or B can determine linking. To make it DoS & *Sybil*[3] resistant we use fees. The use of *ephemeral keys*[4] to recover from malicious attackers that try to link A to B by payment abortion.

S = ***Styx***

Allow A to swap Viacoin for an anonymous Voucher from *S* & allow B to provide T an anonymous Voucher in exchange for Viacoins. The Fair Atomic Protocol prevents *S* from stealing Viacoins by refusing to issue a voucher. The security rely on RSA and equivocal encryption in the ROM [5](random oracle model).

Implement Atomic fair-exchange protocol using smart contracts with Viacoin current scripting functionality. It allows us to implement contracts that exchange a Viacoin for the preimage/hash or an Elliptic Curve Digital Signature Algorithm. Combined these limited scripting functionals (on the chain) with a crypto protocol performed off the chain to obtain the desired anonymity system we desire. The Crypto protocol will be fast. It can run in seconds because it rely on RSA blind signatures.

[3] http://freehaven.net/anonbib/cache/sybil.pdf
[4] https://en.wikipedia.org/wiki/**Ephemeral_key**
[5] http://eprint.iacr.org/2015/140.pdf

Every transaction decides the condition if a viacoin held in the transaction can be moved to another transaction. The rules are specified by Script. Script is stack-based and processed from left to right. Scripting provides the flexibility to change parameters or what's needed to spend the transferred Viacoins.

Soffer :*transaction, one offers to pay Viacoin to any party that can sign a transaction who can meet some C as condition.*

Sfulfill:*transaction, which meets C in Soffer*

Script will support time-locking for contracts (CHECKLOCKTIMEVERIFY). *Soffer* specify that a valid *Sfulfill* must be confirmed by the chain within a specific *tw* Time window. If this is not the case, the Viacoins in *Soffer are no longer on offer to others.*

Hashing **OP_RIPEDMD160** Condition *C* is specified in *Soffer*. *Sfulfill* must contain preimage of y under *H* hash function. *H* is **RIPEMD160**. *Sfulfill* condition by *x* such that *H(x) = y*

Signing condition **OP_CHECKSIGVERIFY** The condition *C* is specified in *Soffer*. *Sfulfill* must be signed by a signature which verifies under *PK*, if this condition is valid signed by the secret key which correspond to *PK*. Viacoin requires the signature to be Elliptic Curve Digital Signature Algorithm over **Secp256k1** elliptic curve.

2.2 Styx RSA Atomic Fair exchange

A pays one Viacoin to *S* and *S* computing the RSA exponentiation to an RSA secret key *sk*. The protocol allows a user to bind a signature and bind a decryption.

$$\frac{-1}{RSA}(y, sk, N) = y^{sk} \bmod N$$

input chose by *y* by *A*, *sk* is *S* secret RSA key and *N* RSA modulus. RSA verification is:

$$fRSA(x, pk, N) = x^{pk} \bmod N \text{ where pk is S public RSA key}$$

3.1 Zero knowledge contingent payments

Swap Viacoins from *A* in exchange to have *S* compute any publicly verifiable function *f* on input of *A* choosing. After *S* computes the result of *f(y)* on input *y*.

It will encrypt the results of under a randomly chosen key for obtaining *c* and hashes the encryption *h = H(k),* then *S* sends A the c and *h* along with a zero knowledge proof. This without *S* revealing k of *f(y)* to person A before being paid with A Viacoins. After proof verified by *A*, *A* posts a transaction *Soffer* to offer one Viacoin under condition **OP_RIPEDMD160.** *S* will claim the Viacoin by posting transaction *Sfulfill* containing *k*. *A* will use *k* to decrypt *c* for *f(y)*. This is an atomic fair exchange because offered coins will be given back to *A* if *S* fails to post a valid *Sfulfill* within the time window.

3.2 Styx Zero knowledge Security Protocol Description

$\frac{-1}{RSA}$ is the trapdoor function, *A* could possibly learn preimages of random values without interaction *F-atomic fair-RSA*. *A* could *"compute" mod N* which *A* can know that y^{sk} *mod N = x*. To make this impossible, users will have to first blind their *F-atomic~fair-RSA*.

S will be asked to provide *n + m* pairs. A will ask *S* to open n of these pairs, by revealing the randomly chosen keys k_i's used to create each of the n pairs. In order for a malicious S to successfully attack A would have to correctly identify all the *n challenge pairs* and form them properly, while at the same time malforming all the

m unopened pairs so it can claim a Viacoin from *A* without actually providing a **blind signature** in return. *S* cannot predict which pairs *A* asks it to open, it will succeed at its goal with an insanely very low probability. If *A* received the opening of *(c, h) pair, A* would be able to recover the blind signature without paying Viacoin. Serial numbers are random values *sn* have the value of a Viacoin. Fake values will be used, to specify *n (c, h)-pair* that *A* asks *S* to decrypt to fake values instead of blind signatures. if *S* opens, *A* has to prove that *n* values are fake.

$$\delta_i = (\rho_i)^{pk} \, mod \, N$$

for random chose $\rho_i \leftarrow R \, Z^*_N$. *A* has to prove that the values of an input providing ρ_i *to S*. When *S* eval *f(-1/RSA)* on a fake input , δ_i

$$(\delta)^{sk} = ((\rho_i)^{pk})^{sk} = \rho_i \, mod \, N$$

A knows . *S* will open *(c,h)* pairs responding to fake values while *A* has to be able to prove they are fake *(c, h)* values. A posts *Soffer* offering one viacoin for *k* which can open all of the *m* real *(c, h)* values. *Sfulfill* contains the hash preimages.

A will obtain *m* blind signatures instead of 1. To make it fair, an extra step will be added. A posts *Soffer,* A proves *S* all m values have the same input and if *S* verifies, *A* posts *Sfulfill* to contain all m of k values that open real *(c, h)* pairs. Real *m* inputs that A sends to *S*

$$dj \; = \; y(rj)pk \; mod \; N$$

d_j *is y* RSA binded under different bind $r_j \leftarrow R \, Z^*_n$ when S signs $\frac{-1}{RSA}$ of real input d_j

$$(d_j)^{sk} = (y(r_j)^{pk})^{sk} = (y)^{sk} \, r_j \, mod \, N$$

After A *Soffer* is confirmed by the chain, A prove *S* real inputs which correspond to d_j A reveals all binds r_j to *S*. *S* redeem it Viacoin from A, *Sfulfill* containing k required to open (c,h) pairs. *A* can obtain the output of f^{-1} on input *y* as long as 1 of *(c,h)* is validly formed and opened by a *k* value in S*fulfill*.

4.1 Styx Zero knowledge Security Voucher

Introducing voucher which can be used only with the viacoin-supported signature condition. *Sfulfill* must be signed by a signature which verifies under PK. The voucher is a transaction *Sfulfill* signed by Viacoin Elliptic Curve Digital Signature Algorithm over Secp256k1 elliptic curve (ECDSA-Secp256k1 signature). The signature will be computed under an *ephemeral* public key *SK(eph/S), PK(eph/S)* chosen by *S*.

S post a **timelocked** *Soffer* offering 1 Viacoin in exchange for a valid voucher from *B*. It must be digitally signed by an **ECDSA-Secp256k1** that will verify under *PK(eph/S) & PK_B*. *S* redeems the voucher by B if submitted by *B*.
B let *S* sign the voucher $z = fRSA(\in, pk, N)$

It's also possible for B to blind *z* before sending it to *A*. If A doesn't care about anonymity, B would still be anonymous. $\in$ let B open & encrypt c_e to a valid voucher. ECDSA-Secp256k1 σ_e on a transaction *Sfulfill*.

B creates a set μ+n hash val B_e for l = 1… μ+n. By permutation of the hashes of the real and fake, B_e are send to *S*. *S* signs every B_e to obtain an ECDSA-Secp256k1 σ_e and each is then hidden in inside C_e which is possible to decrypt with key $\in_e$. *S* hides each encryption key $\in_e$ by using the RSA trapdoor function

$$z_e = f\,RSA(\in_e, pk, N)$$

k pairs of will be send back to *B*. *B* must decide which encryption $\in_e$ to give to A. For Decryption B knows which C_e Z_e pair is valid formed. This can be done with chaining together $\in_e$ values that corresponds to real values. S provides B with μ -1 quotients

$q2 = \in j2/\in j1, \ldots, q\mu = \in j\mu/\in j\mu\text{-}1 \bmod N$
where $j1,\ldots,j\mu = R$ as in real values. $\in j1$ allows B to recover of all other $\in j_e$ *since*
$\in_{je} = \in 1 * q2 * \ldots * q_e$

B asks *A* to invert *z*. *A* interacts with *S* to blindly invert Z by using the RSA *sk* in exchange for *A* Viacoin. Nobody can link *S* current interaction with *A* with value *z*. *S* won't be able to who is paying who.
Blind inversion, A will obtain preimage of *z* by contacting *S*. *A binds to z*
$y = z^* r^{pk} \, mod \, N$

where r ← R Z^*_N where R is the random chosen blind value. *A* will perform the RSA Atomic Fair exchange with *S* on input *y*. *A* swap Viacoin in exchange to y^{sk}
$y^{sk} = (z * r^{pk})^{sk} = (z)^{sk} * r = \in * r \, mod \, N$

The result is the inversion of *z* blinded with *r*. *A* unblind
$(y)^{sk} / r = \in * r/r = \in mod \, N$ unveiling $\in$

A sends $\in$ *to B* and uses $\in$ to open the c_e that contains the ECDSA-Secp256k1 signature σ_e on transaction *Sfulfill*. Finally B ECDSA signs *Sfulfill* under SK_B and post the result to the chain to claim his Viacoins *from S*

4.2 Atomic anonymity

- Completeness — *B* obtains $z_i = f\,RSA(\in_i, pk, N)$ and c_i valid ECDSA-Secp2561k1 signatures that can be unlocked with $\in_i$. *S* issued locked signatures c_i that can be unlocked only if the A pays to learn $f^{-1}_{RSA}(z_{j1}, sk, N)$

- Fairness — If *S* is malicious there is a negligible probability that the atomic protocol successfully ends but B is not able to unlock c_i. If B is malicious, malicious *S* won't accept a voucher from *B*. *S* won't complete *fRSA*

- Anonymity — A&B should not be linkable. B provides *z* to A and *A* provides $\in$ to B. Communication between A and B is preserved. *S* or other 3rd party can use values to link payments. A "blinds" *z* to *y* before the atomic protocol with S. A will unblind results $\in$ and send it to *B*

- Sybil & DoS — *S* should not trust anyone. Use of anonymous vouchers is a blind signature which *S* can send to *A* in exchange for a payment. *A* is able to buy vouchers in bulk before participation with *S*. *A* can unblind the vouchers and send it to *B* who will pass it to *S*. This is all done off-chain.

5.1 Overview

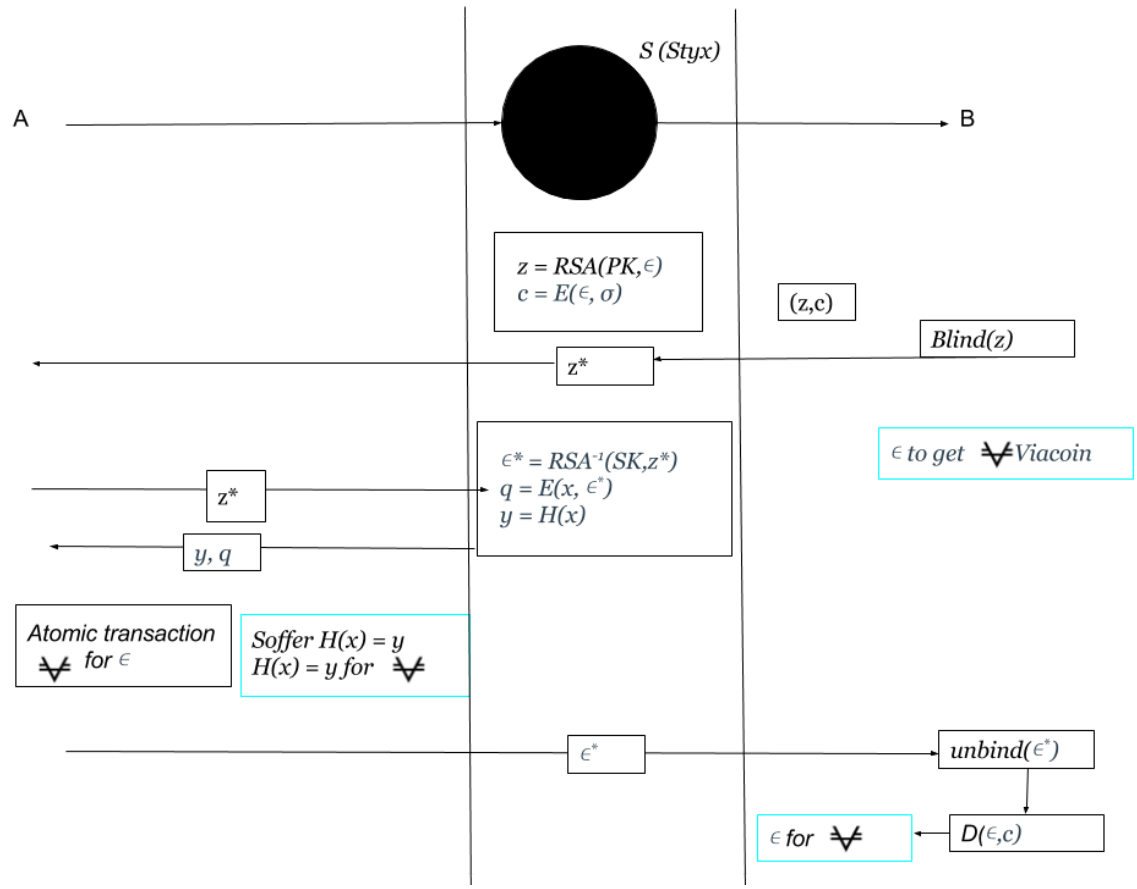


figure 1 Protocol overview

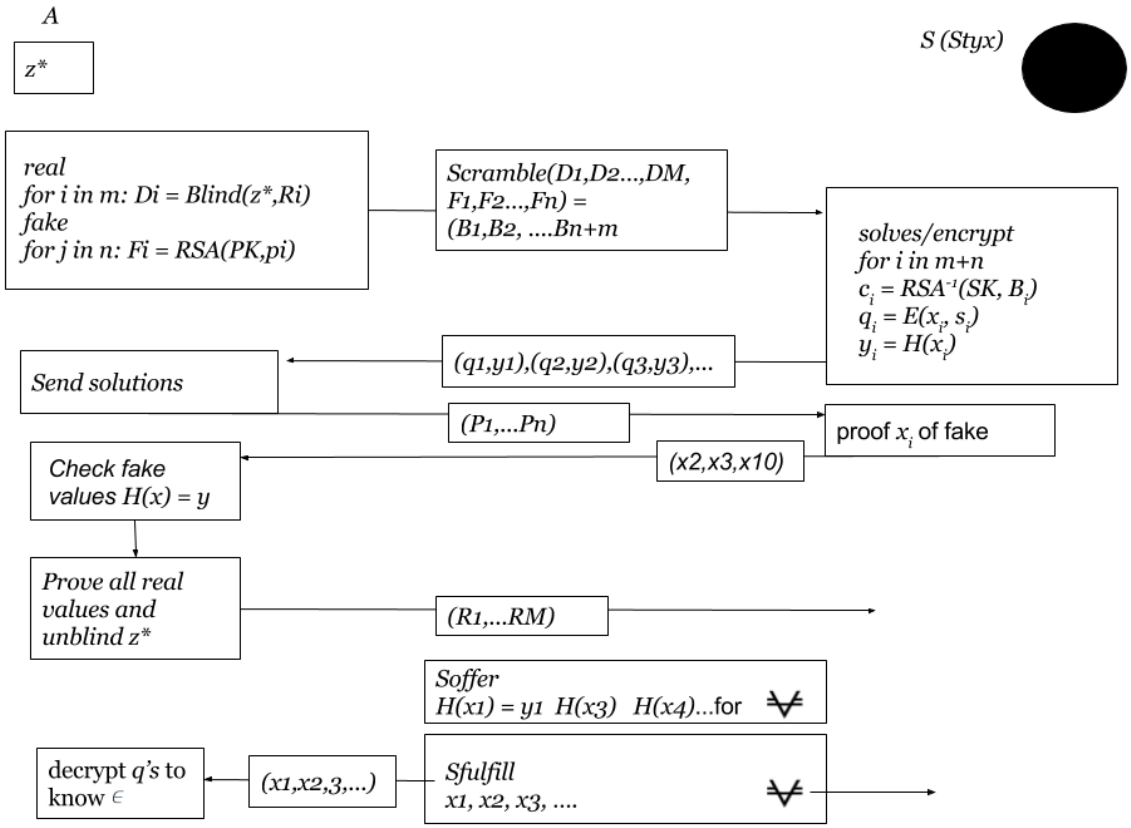


figure 2 Protocol solver

5.2 implementation

Instantiation of the protocol with 2048-bit RSA. Hash functions and signatures are instantiated as following.

Viacoin clients and miners operate transactions with the Viacoin script templates. Using P2SH transaction template that is used for standard transactions. This is a standard hash type.

Soffer P2SH transaction specified with the redeem script. Conditions must meet fulfill conditions of the transaction. The redeem script is hashed. The hash is stored in *Soffer*. For spending a transaction offer, the transaction fulfill has to be constructed.

Sfulfill includes the redeem script and a set of input values that the redeemscript is run against. The redeem script can be used for *fRSA*. The protocol can check that the input values in the transaction fulfill contain the preimages and the *Sfulfill* is signed by the public key of A. PubKey is *S* permanent viacoin address. Input values and redeem is run against the signature. It's a signature under the *S* permanent Viacoin address.

The real values RIPEMD-160 hash outputs must be stored in the redeem script (*fRSA)*. P2SH redeem scripts are limited up to 520 bytes. increasing real values also increases transaction fees. Fees paid to (miners) confirm a transaction on the viacoin chain scale with the size of the transaction. The number of fake values can be bigger since they are handled completely off the chain. The real values can be under 20 and fake values under 300. We could minimize the RSA computations that he fake values are below 50.

Conclusion:

Styx will be implemented as a layer on top of Viacoin and be compatible with the Viacoin protocol. Styx provides payments which are private. Privacy can't be violated and coins can't be stolen. Styx is a system that allows to make off-chain transactions with the same level of security as on-chain transactions. It's also an extra way to add anonymity and speedup transactions without choking the network.

Styx payment hub provides unlinkability during the payment phases. Because the use of off-chain transactions, the payments can be confirmed in seconds. Styx is possible because of existing opcodes that exist in viacoin, provides private and scalable payments while making it unlinkable to provide privacy and anonymity.

6.1 Acknowledgement

We want to thank all people who contributed on the whitepaper and thank to the authors of other books and whitepapers that was used to study and create the Styx whitepaper. Credits to the authors of the ROM whitepaper and Tumblebit. The Styx Unlinkable Anonymous Atomic Payment Hub white paper is based on the following reference material.

6.2 References

1. https://bitcoin.org/en/
2. http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization/
3. https://en.bitcoin.it/wiki/Script
4. Masayuki Abe. A secure three-move blind signature scheme for polynomially many signatures. In EUROCRYPT'01, pages 136–151, 2001.
5. Adam Back, G Maxwell, M Corallo, Mark Friedenbach, and L Dashjr. Enabling blockchain innovations with pegged sidechains. 2014.
6. https://en.bitcoin.it/wiki/Contract
7. Wac law Banasik, Stefan Dziembowski, and Daniel Malinowski. Efficient zero-knowledge contingent payments in cryptocurrencies without scripts. Cryptology ePrint Archive, Report 2016/451, 2016.
8. http://eprint.iacr.org/.
9. Pedro Franco. Understanding Bitcoin Cryptography, Engineering and economics pages 209 - 246, 2015
10. Christof Paar Jan Pelz. Understanding Cryptography pages 174 - 192, 2009
11. Greg Maxwell. Zero knowledge contingent payment.
12. Sarah Meiklejohn and Claudio Orlandi. Privacy-enhancing overlays in bitcoin. In Financial Cryptography and Data Security, volume 8976, pages 127–141. 2015.
13. Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. (2012):28, 2008
14. Ethan Heilman, Leen Alshenibr, Foteini Baldimtsi, Alessandra Scafuro, Sharon Goldberg Boston University. TumbleBit anonymous payment hub
15. https://eprint.iacr.org/2016/575.pdf
16. Ethan Heilman, Leen AlShenibr. Tumblebit
17. https://scalingbitcoin.org/transcript/milan2016/tumblebit
18. Eli Ben Sasson, Alessandro Chiesa, Christina Garman,
19. Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In IEEE Security and Privacy (SP), pages 459–474, 2014.
20. eCash http://www.investopedia.com/terms/a/anonymoustrading.asp
21. Dan Boneh and Xavier Boyen. Secure identity based encryption without random oracles. In Proceedings of the 24th Annual International Cryptology Conference, CRYPTO '04, pages 443–459, 2004.
22. Luke Valenta and Brendan Rowan. Blindcoin: Blinded, accountable mixes for bitcoin. In Financial Cryptography and Data Security, pages 112–126. Springer, 2015.

23. Peter Todd. Bip 65: Op OP_CHECKLOCKTIMEVERIFY. Bitcoin improvement proposal, 2014.
24. Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, and Madars Virza. Snarks for C: verifying program executions succinctly and in zero knowledge. In CRYPTO, pages 90–108, 2013.
25. David Chaum. Blind signature system. In CRYPTO, 1983.
26. Antoine Delignat-Lavaud, Cedric Fournet, Markulf Kohlweiss,and Bryan Parno. Cinderella: Turning shabby x. 509 certificates into elegant anonymous credentials with the magic of verifiable computation.
27. George Bissias, A Pinar Ozisik, Brian N Levine, and Marc Liberatore. Sybil-resistant mixing for bitcoin. In Workshop on Privacy in the Electronic Society, pages 149–158. ACM, 2014
28. Neal Koblitz and Alfred J. Menezes. The Random Oracle Model, 2015 http://eprint.iacr.org/2015/140.pdf