Styx: Via 币上的匿名制原子性支付平台

Via 币研发团队

viacoin.org

2016/10/14

大纲

这份白皮书介绍 Styx，一个 via 币使用的匿名原子性交易平台。Styx 的特色是高匿名性，采用零知识或有付款(zero knowledge contingent payment proof)，可让使用者匿名付款、既安全又迅速，过程中不留下任何可追踪信息，让 via 币的持有者们能够安全地保护自己的资产，减去被他人盗窃的风险。

这项协议透过 via 币的 script 功能结合了区块链外的加密计算并使用 RSA 加密算法、随机预言机模型(ROM)及椭圆曲线密码学。

目录

1.1 替代性

替代性是任何一个正常运作的货币所该具有的功能。例如，黄金是可替代的，因为一公克的黄金价值为一公克的黄金，但是如果今天这项资产消失了，替代性就会消逝殆尽。

黄金是可替代的，因为你无法分辨一块黄金与另一块的区别，并没有分出旧黄金、新黄金等的概念，每一块都是依重量计价的，不会因开采地点而改变价位，也不会受到审查制度

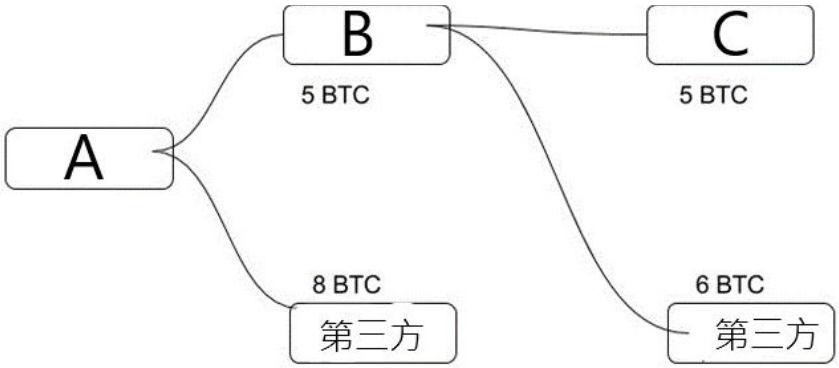
在许多的现代数字交易系统中，替代性是个很显着的问题。例如，Paypal 的审查制度可因买卖方的可疑交易信息而冻结用户的帐户及交易程序。
中心化的虚拟货币交易平台像是 Coinbase 也可因款项是来源可疑，像是在线聚赌网站，而冻结使用者的账户。他们可以从区块链上查看交易款项的来源，但是现在假如交易来源变得无法追踪，这类的审查制度将会无法实行，也将不会威胁货币的替代性。

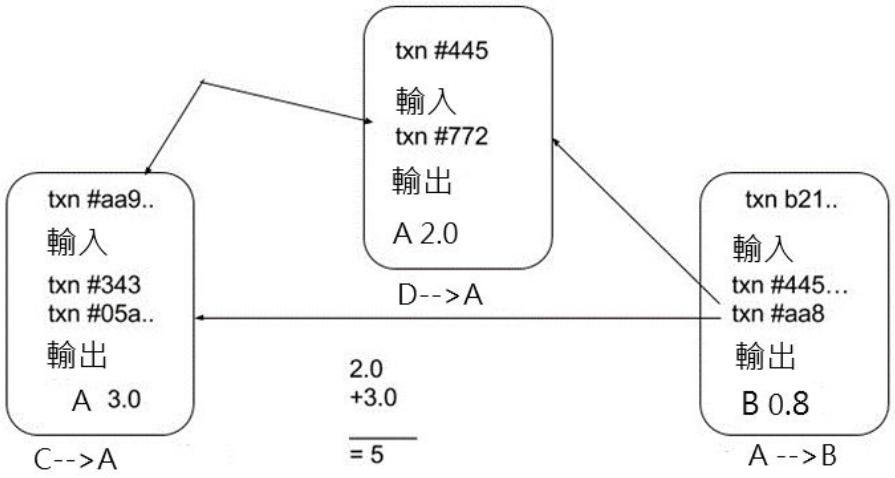
理论上，在比特币的交易认证程序中，挖矿者有可以避免特定的货币被交易的可能性，像是在之前恶名昭彰的以太坊 DOA 黑客事件中就被尝试过。有些人将这次事件看待为一场单纯的窃盗案，但是暂时也没有其他方式阻止有心人对其他货币做出相同的行为。

其中一个解决的方式就是让付款的流向变得无法追踪，或者，将信息模糊处理至在区域链上无法辨识的程度，也就表示，所有的付款将会是相等的，没有任一项付款会受到过去纪录与流向影响，或是受到第三者操弄。

让我们再更深入探讨:

假如今天 A 传送五枚硬币给 B，A 需要申明说他之前领取了五枚硬币，节点便会检查 A 是否为领取者与现持有者，以及总额是否加起来为五枚硬币，B 之后便会取得相同的五枚硬币，而当有第三者查看 A 与 B 在区块链上的交易纪录时，他甚至能查看到之前传送给 A 的使用者 C。


B
C
5 BTC
5 BTC
A
8 BTC
6 BTC
第三方
第三方


txn #445
輸入
txn #772
輸出
A 2.0
D-->A
txn #aa9..
輸入
txn #343
txn #05a..
輸出
A 3.0
C-->A
2.0
+3.0
= 5
txn b21..
輸入
txn #445...
txn #aa8
輸出
B 0.8
A -->B

除了区块链之外，节点会另外储存数据，名为比特币交易请求量(UTXO)。UTXO 是个会记录每个地址中货币总额的账本，有如区块链上的缓存。

当新的交易进行时，UTXO 就会被更新。相对于公开账本，还有需多方式可以追踪货币的交易纪录:

-交易图及脱机信息

从交易图上可看出交易在何处发生的以及连接以及款项的输出与输入。恶意人士可以透过用户在网上的信息及公开的数据库中 寻找出用户的交易地址。有许多用户都会将交易地址遗留在社群交流平台上以便他人交易。恶意人士便会整合所有从网络分析上搜集的信息与一些个人的数据库，像是交易平台、在线电子货币包等等。

-IP 流量中继

恶意人士可以监控 P2p 网络，观察资金的源头 IP。

-集群分析

从图表式的数据中寻找聚集的数据集群，从中筛选出吻合的数据组，例如，送至交易平台或者是博弈网站的数据就具有可识别性。

-使用分析与时相分析

较大量的资金可从互通的使用者中追踪。若一位特定的使用者在特定的时间点获取了相对于之前余额大量的资金，他就会被归纳出来。

对于交易可被连接到特定使用者及服务这点严重伤害了货币本身的替代性。目前的比特币及以比特币作为基础了另类货币目前还尚未解决替代性的问题，而在交易追踪技术愈加发达的每一天，替代性也更备受威胁。

为了让 Via 币更加地有可塑性及防盗，我们有义务加强 Via 币的可替代性来保护用户的个人资产与提升用户的隐私，并提供匿名的交易方式。我们坚信虚拟货币市场未来将不容有不可替代的货币，所以我们以可替代性为 Via 币的当务之急。

有一定数量的去中心化虚拟货币(例如: 门罗币、Zcash)有匿名的交易功能，但是它们与比特币并不兼容。Via 币，则是以比特币为基础，由原本的原始码中探索一个既可靠又安全的方案，这就是 Via 币脱癒而出的地方。

1.2 eCash 中心化匿名付款机制

在我们讨论 Via 币的匿名付款系统之前，让我们回想 eCash。一个在比特币之前的付款系统，eCash 仰赖银行而且并无去中心化，他是个可以让使用者在匿名下汇款的系统，再从用户上收取手续费。

eCash 使用的技术是盲签名。由这个系统，金额的输入与输出之前并无联系。eCash 的系统保证付款人 A 及收款人 B 之前没有联系。

一个盲签名可用 RSA 加密演算呈现，签名者有着 RSA *sk*(密钥)，他也同时拥有 *sk*(密钥)及 pk(公钥)。Pk, N，N 代表系数。G 代表全局哈希。A 写了讯息 sn，A 可以创造盲签名讯息 $\bar{sn}$，由选择一个任意价值 r ← Z* N

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

创造的盲签名:

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

A 可以将签名去盲化:

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

这就是盲签名的运作原理，Via 币也可以采用相同原理，若 A 传送 S 一个 Via 币及一个盲化的序号$\bar{sn}$，而 A 取得一个盲签名 σ。A 然后将这个价值去盲化来创出一个匿名的凭证 (sn, σ)，藉由这个将款项汇给 B。

但是这个方式是有瑕疵的，若今天 S 的意图不轨，他可以拒绝给予 A 盲签名或者是拒绝匿名凭证并且取得对方的 Via 币。这个可被 Via 币内建的原子性交易机制解决: A 以 Via 币与 S 交换一个盲签名，然后再与 B 交换匿名凭证，原子性的交易就可以达成透过 via 币的 script 功能达成。

2.1 Styx 原子性支付协议

Via 币有原子性混币功能，每一笔付款项目都需通过三次认证，除了交易人 A 与 B 之外没有人可以查明连结，我们已收取手续费的方式抵挡 DoS 及 *Sybil* 的攻击，以及使用临时密钥来恢复受攻击的使用者。

S = ***Styx***

首先让 A 用 Via 币交换 B 的匿名凭证然后再让 B 供给 T 一个匿名凭证去交换 Via 币。Styx 原子性支付协议禁止 S 拒绝给予 A 凭证，也避免了 S 将 Via 币偷走的风险。这项技术仰赖了 RSA 加密及随机预言机模型中的模糊加密。

我们使用 Via 币 Script 功能中的智能合约来执行原子性支付协议，它让我们能用一个 Via 币去交换一个哈希或是一个椭圆曲线密码计算。将这些各种 script 功能与区块链外的协议结合后我们就能得到一个我们满意的系统。这个协议的运行速度会十分迅速，它可在数秒内进行因为它仰赖的是 RSA 盲签名系统。

每一项交易将可决定 Via 币现在的交易状态及可否被传输，这些规定都在 Script 上具体指出。Script 是基于堆积的并是由左而右来处理的，也提供了可塑性让我们可以改变 Via 币送出时的参数。

S 报价： *一个人送出 Via 币给一群可以答应一笔达到状态 C 的交易*

S 实现: *达到 S 的报价状态 C*

Script 会支持合约的定时锁闭(CHECKLOCKTIMEVERIFY)，*S 报价*指定一个有效的 *S 实现*需要在一个特定的时间范围 *tw* 内被区块链核准。若未达成，Via 币的 *S 报价*便不会再继续对其他人显示报价请求。

哈希 **OP_RIPEDMD160** 状态 C 已被 *S 报价*指定。*S 实现*需在哈希功能下包含一个 y 的原象。H 是 **RIPEMD160。***S 实现*状态 $H(x) = y$

签署状态 **OP_CHECKSIGVERIFY** ，状态 C 已在 *S 报价*中遭指定，*S 实现*需一个被 PK 核准的签名，若这个状态是与 PK 对应的密钥签署的，Via 币需要的签名将会使用椭圆曲线数字签名算法。

2.2 Styx RSA 原子性交易

A 给予 S 一枚 Via 币然后 S 将 RSA 的乘幕演算为密钥 *sk*。这个协议允许用户绑定签名与解密。

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

输入为 y 及 A 决定，sk 是个 RSA 密钥及 N RSA 参数。RSA 确认则是:

$$fRSA(x, pk, N) = x^{pk} \bmod N$$ *pk 是 S 公共 RSA 密钥*

3.1 零知识证明或有付款

从 A 交换 Via 币来以 S 计算 A 选择的任何可公开验证的功能 *f*。在 S 计算 *f*（y）的结果后，它将以随机选择的密钥将结果进行加密来得到 c 并加密 h = H（k），然后 S 发送 c、h 以及零知识证明给 A，在没有 S 向 A 显示 f（y）的 k 的情况下以及在接收 A 的 Via 币付款之前。 在经过 A 的证明后，A 会在条件 OP_RIPEDMD160 下发布一个 *S 报价*的讯号并传送一枚 Via 币，S 则会发布包含 k 的交易 *S 实现*来领取 Via 币。 A 将使用 k 为 f（y）解密 c。 这是一个原子性交易，因为如果 S 在时间窗口内没有发布一个有效的 *S 实现*，那么提供的硬币将会归还给 A。

3.2 零知识证明介绍

$\frac{-1}{RSA}$是陷门函数，A 可以在没有 F 原子性 RSA 交互的情况下学习随机值的原象。 A 可以“计算”mod N，但是会让 A 知道 $y^{sk} \bmod N = x$。为了使这不可能，用户将不得不首先盲化他们的 F 原子性 RSA。

S 将被要求提供 n + m 对。 A 通过揭开那些用于创建每一个 n 对的随机选择密钥 k i 来请求 S 打开这些中的 n 个对。要使怀有恶意的 S 成功攻击 A，S 必须正确识别所有 n 个*挑战对*并正确将它们生成，并同时对所有未打开的对进行畸形，才可以在不提供盲签名的状态下从 A 使用者去得一个 Via 币。 S 不能预测哪一对是 A 要求打开的，它只能以非常低成功率达到目标。若 A 收到（c，h）对的开头，A 将能够在不支付 Via 币的情况下恢复盲签名。序号是具有 Via 币价值的随机值 sn。假价值会用来指定 n（c，h），用来让 A 请求 S 解密伪价值，而不是盲签名。如果 S 打开，A 必须证明 n 值是假的。

$\delta_i = (\rho i)^{pk} \bmod N$ 用于随机选择 $\rho_i \leftarrow R Z^*_N$。 A 必须提供 ρ_i 到 S 输入的值。当在假输入上的 S 评估 $f(1/RSA)$ 时，δ_i

$(\delta)^{sk} = ((\rho i)^{pk})^{sk} = \rho_i \bmod N$

A 知道 S 将打开（c，h）对来对应假值，而 A 必须能够证明它们是假的（c，h）值。A 提出 *S 报价*提供一个 Via 币给 k 来打开所有 m 个真实（c，h）值。 *S 实现*包含哈希原象。

A 将获得 m 个盲签名而不是仅 1 个，为了使其公平，将增加一个额外的步骤: A 提出 *S 报价*，A 证明 S 所有 m 值具有相同的输入，如果经 S 验证后，A 则发送 *S 实现*以包含来打开实际（c，h）对 k 中 m 个值。 A 发送到 S 的真实输入:

dj = y（rj）pk mod N d_j 是在不同绑定下绑定的 RSA $R_j \leftarrow R Z^*_n$，当 S 写入$\frac{-1}{RSA}$的实际输入 d_j

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

在 A 的 *S 报价*被区块链确认后，A 证明 S 对应于可显示所有绑定 r_j 到 S 的 d_j A 的实际输入，S 将 Via 币赎回，需要包含 k 的 *S 实现*来开启(c,h)对，

S 将其从 A，*S 实现*中兑换为 Via 币，其中包含开放（c，h）对所需的 k。 只要其一的（c，h）有效地形成并由 *S 实现*的 k 值打开，A 就可以取得在输入 y 上 f_{-1}的输出。

4.1 Styx 零知识安全凭证

我们来介绍一个只能以 Via 币签名条件使用的凭证。 *S 实现*必须透过 PK 验证的签名来签署。 凭证是 Via 币椭圆曲线数字签名算法在 Secp256k1 椭圆曲线（ECDSASecp256k1 签名）上签署的交易 *S 实现*。 签名将在 S 选择的临时公钥 SK（eph / S），PK（eph / S）下计算。

S 发布一个时间锁定的 *S 报价*，提供 1 个 Via 币，以从 B 换取有效的凭证。它必须由 ECDSASecp256k1 进行数字签名，并经由 PK（eph / S）& PK B 来进行验证。 如果凭证由 B 提交，S 则可透过 B 赎回凭证。

B 让 S 签署凭证 z = fRSA（ε，pk，N）

B 也可能在将 Z 发送给 A 之前将其盲化。如果 A 不关心匿名，B 仍就会是匿名的。 ε 让 B 打开并加密 c e 到有效的凭证。 ECDSASecp256k1σe 在交易 *S 实现*上。

B 为 l = 1 ...μ+ n 创建集合 μ+ n 哈希值 B e。 通过排列真实和假的散列，B e 会被发送给 S。 S 签署每个 B e 以获得 ECDSASecp256k1σe，然后隐藏在 C e 内部，这可以用密钥 εe 解密。 S 通过使用 RSA 陷门演算 $z\,e = f\,\mathrm{RSA}(\in_e, pk, \mathrm{N})$ 隐藏每个加密密钥 εe 。

k 对将被发送回 B。 B 必须决定给予 A 哪个加密 εe。因为解密 B 知道哪个 C e Z e 对有效形成。这可以通过连结对应于实际值的 εe 值来完成。 S 提供 B 与 μ-1 商数。

q2=...，qμ=εjμ/εjμ1mod N 其中 j1，...，jμ= R 等于实数值。 εj1 允许 B 恢复所有其他 εje，因为 εje =ε1* q2 * ... * q e

B 要求 A 反转 z。 A 与 S 进行交互，透过使用 RSA sk 来交换 A 的 Via 币盲目地反转 Z。没有人可以将 S 当前的交互与 A 与值 z 连结起来。

S 将无法知道谁支付给谁。

盲反转，A 将透过与 S 接触以获得 z 的原像。 A 结合 $z\ y = z * r_{pk} \bmod N$

其中 r←R Z * N，R 是随机选择的盲值。 A 将在输入 y 上执行与 S 的 RSA 原子性交换。交换 Via 币到 $y^{sk}\ y^{sk} = (z * r^{pk})^{sk} = (z)^{sk} * r = \varepsilon * r \bmod N$

结果是 z 的倒转与 r 绑定。A 去盲化

$(y)sk / r = \varepsilon * r / r = \varepsilon \bmod N$ 揭开 ε

A 向 B 发送 ε，并使用 ε 来打开包含 ECDSA-Secp256k1 签名 σ_e，且在交易 *S 实现*上的 c_e。最后，B ECDSA 在 SK_B 下签名 *S 实现*，并将结果发布到链上，从 S 索取他的 Via 币。

4.2 原子性匿名

- ➢ 完整性

B 得到 $z_i = f\,RSA(\varepsilon i, pk, N)$ 且 c_i 有效。

ECDSA-Secp2561k1 签名可用 εi 解锁。 S 发出锁定签名 c_i，只有当 A 认知 $f^{-1}_{RSA}(z_{j1}, sk, N)$ 才能解锁。

➢公平性

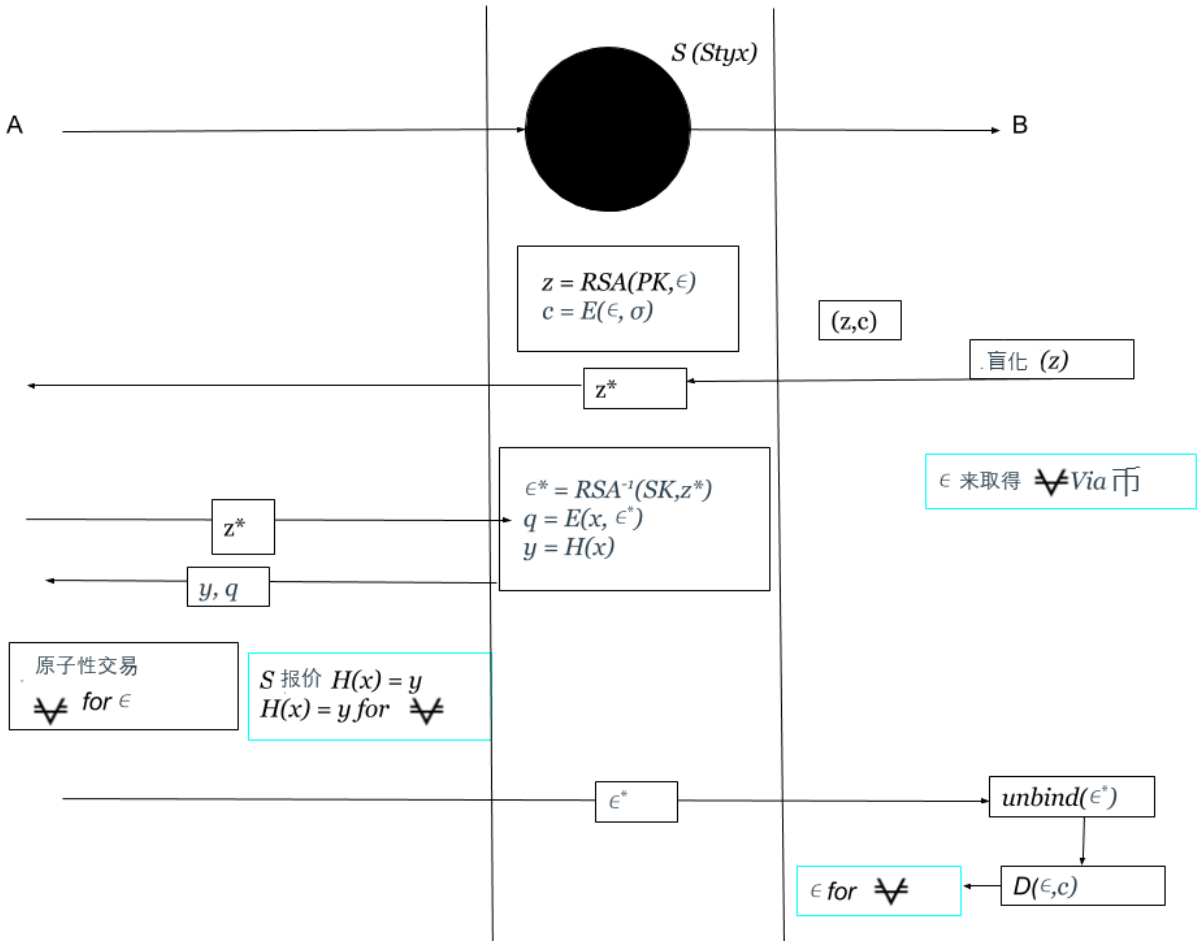
如果 S 是恶意的，原子性协议成功结束的可能性很小，但 B 无法解锁。如果 B 是恶意的，恶意 S 不接受 B 的凭证。 S 不会完成 *fRSA*。

➢匿名性

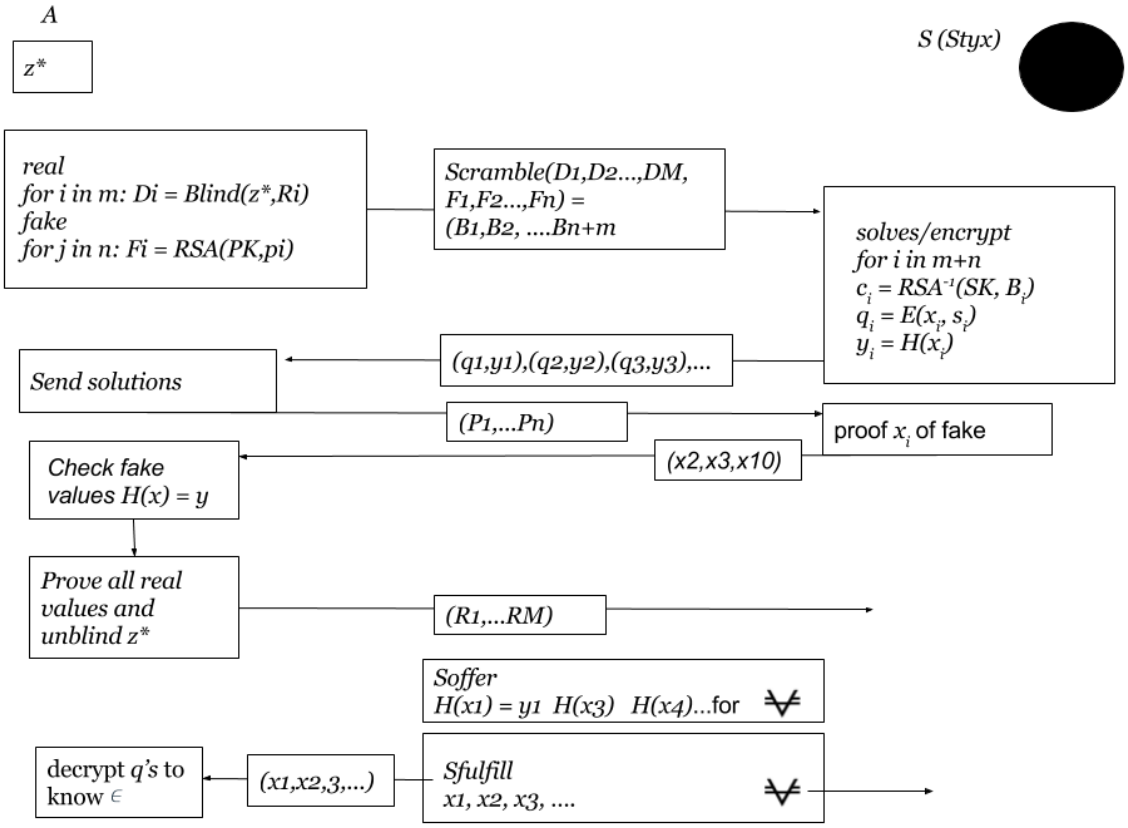
A & B 不可连结。 B 提供 z 给 A，A 提供 ε 给 B。A 和 B 之间的通信会被保存。 S 或第三方可以使用价值连接付款。在 S、A 的原子性协议之前， A 在 z 到 y 之间的盲化将产生非盲化结果 ε 及发送其到 B。

➢ Sybil 及 DoS

S 不应该信任任何人。使用匿名凭证是使 S 有可以发送给 A 的一个换取付款的盲签名。 A 能够在 S 参与之前大量购买凭单。 A 可以解除凭证，并将其发送给 B，然后又再发送给 S。这一切都是在区块链之外完成的。


S (Styx)
A
B
z = RSA(PK,∈)
c = E(∈, σ)
(z,c)
盲化 (z)
z*
∈* = RSA⁻¹(SK,z*)
q = E(x, ∈*)
y = H(x)
∈ 来取得 Via 币
z*
y, q
原子性交易
for ∈
S 报价 H(x) = y
H(x) = y for
∈*
unbind(∈*)
∈ for
D(∈,c)

5.1 概观



5.2 采用

2048 位 RSA 协议实例化。哈希函数和签名被实例化如下。

Via 币客户和矿工与 Via 币脚本模板进行交易。使用用于标准交易的 P2SH 交易模板。这是一个标准的哈希类型。

S 报价 P2SH 交易指定的兑换脚本。条件必须符合交易的履行条件。兑换脚本是散列的。散列存储在 *S 报价*中。为了支付交易款项，构建交易必须被履行。

*S 实现*包括兑换脚本和一组输入值，用于对付兑换脚本。兑换脚本可用于 *fRSA*。协议可以检查事务履行中的输入值是否包含原象，且 *S 实现*由 A 的公钥签名。公钥是 S 永久性的 Via 币地址。输入值和兑换是违反签名的。这是 S 永久 Via 地址下的签名。

RIPEMD160 实际值的哈希输出必须存储在兑换脚本（fRSA）中。 P2SH 兑换脚本限制为 520 字节。增加实际价值也会增加交易手续费。支付给（矿工）的费用可确认 Via 币在链上的交易数量。虚假的值的数量可以增大，因为它们完全处于区块链之外。实际值可以低于 20，伪值低于 300。我们可以将 RSA 计算最小化，让其值低于 50。

5.3 结论

Styx 将在 Via 币上实现，并与 Via 币的原案相容。 Styx 提供私人付款。 隐私不易被侵犯，硬币不会被盗窃。 Styx 可保证链下交易与链上交易有着等同的安全性。 它是有着匿名性与快速交易而不危害网络速度的另一种方式。

Styx 支付中心在付款阶段提供不可追踪性。 因为使用链外交易，付款可以在几秒钟内确认。 Styx 是可能的，因为存在于 Via 币中现有的操作码，让其提供隐密和可延展的付款，同时使其不可追踪以提供隐私和匿名。

6.1 感谢

我们要感谢所有在白皮书上做出贡献的人，与这篇取材引用的其他书籍作者们，并感谢 ROM 白皮书的作者们与 Tumblebit。以下是 Styx 原子性匿名支付协议白皮书所采用的参考数据。

6.2 引文

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