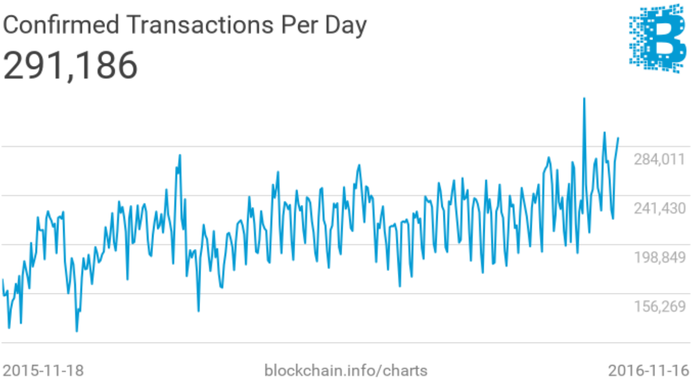
Centrally Banked Cryptocurrencies
Security Topics

Avi Weinstock
(Paper by George Danezis and Sarah Meiklejohn)

November 11, 2016

BTC scalability



[1]

$\frac{291186.0}{24*60*60} \approx 3.3702083333333333$

Theoretical max $\frac{\text{transactions}}{\text{second}}$ is 7

[1] https://api.blockchain.info/charts/previews/n-transactions.png?start=1447864475&lang=en&h=405&w=720

RSCoin overview

- Goal: Create a new cryptocurrency framework that sacrifices decentralized money generation for better performance
- "Central bank" authorises a large number of "mintettes" to make blocks via PKI
- Double-spending prevention is still moderately decentralized

Blockchain hierarchy

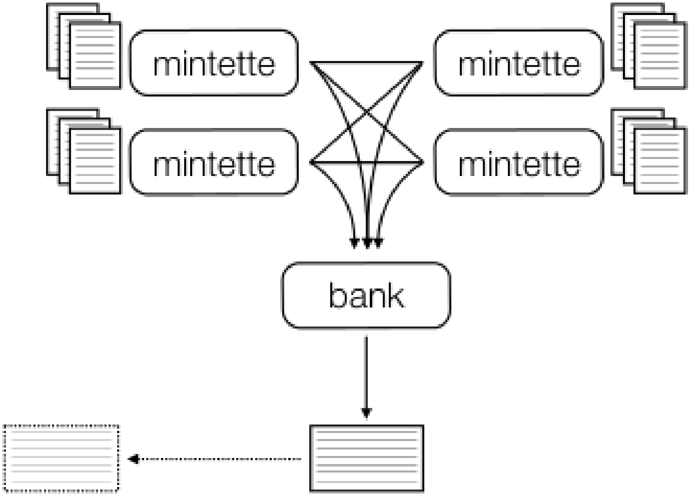


Fig. 1: The overall structure of RSCoin. Each mintettes maintains a set of lower-level blocks, and (possibly) communicates with other mintettes (either directly or indirectly). At some point, the mintettes send these blocks to the central bank, which

Interaction with mintettes

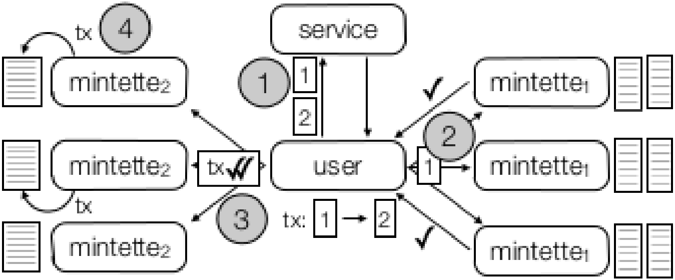


Fig. 2: The proposed protocol for validating transactions; each mintette m_i is an owner of address i. In (1), a user learns the owners of each of the addresses in its transaction. In (2), the user collects approval from a majority of the owners of the input addresses. In (3), the user sends the transaction and these approvals to the owners of the transaction identifier. In (4), some subset of these mintettes add the transaction to their blocks.

Notation (signatures and transactions)

- $(pk, sk) \leftarrow_{\$} \texttt{KeyGen}(1^{\lambda})$
  $\sigma \leftarrow_{\$} \texttt{Sign}(sk, m)$
  $\{0, 1\} \leftarrow \texttt{Verify}(pk, m, \sigma)$
- $\texttt{tx} = \{\texttt{addr}_i\} \rightarrow^{n} \{\texttt{addr}_j\}$
- $\texttt{addrid} = (\texttt{tx}, \texttt{index}_{\texttt{tx}}(\texttt{addr}), v)$
- $\texttt{uxto} : \texttt{addrid} \rightarrow \{\bot\} \cup (\texttt{addr} \times v)$
- $\texttt{pset} : \texttt{addrid} \rightarrow \texttt{tx}$

Notation (high-level blocks)

High-level blocks (produced by central bank):

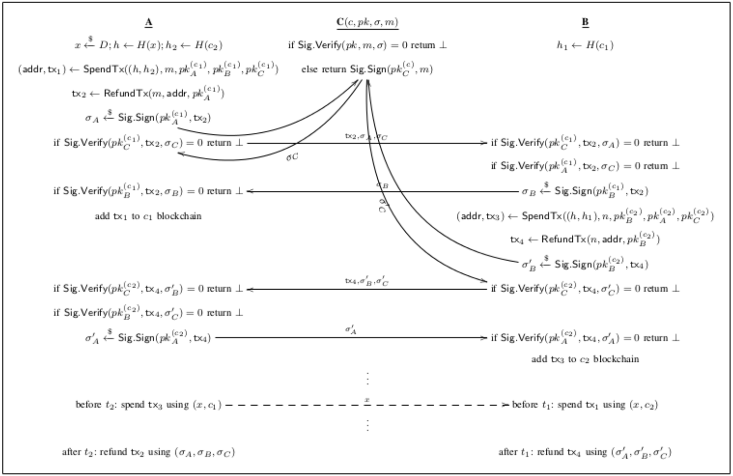
- $B^i_{\mathtt{bank}} = (h^i_{\mathtt{bank}}, \mathtt{txset}_i, \sigma^i_{\mathtt{bank}}, \mathtt{DPK}_{i+1})$
- $h^i_{\mathtt{bank}} = H(h^{i-1}_{\mathtt{bank}} || \mathtt{txset}_i)$
- $\sigma^i_{\mathtt{bank}} = \mathtt{Sign}(sk_{\mathtt{bank}}, h^i_{\mathtt{bank}})$

Notation (low-level blocks)

Low-level blocks (produced by mintettes):

- $\mathrm{b}_m^j = (h_m^j, \texttt{txset}, \sigma, \texttt{mset})$
- $h_m^j = H(h_{\texttt{bank}}^i || h_{j-1}^m || \texttt{otherblocks} || \texttt{texset})$
- $\sigma = \texttt{Sign}(sk_m, h)$
- $\exists \sigma_{\texttt{bank}}((pk_m, \sigma_{\texttt{bank}}) \in \texttt{DPK}_i$

Sidechains



A

$x \xleftarrow{\$} D; h \leftarrow H(x); h_2 \leftarrow H(c_2)$

$(\mathsf{addr}, \mathsf{tx}_1) \leftarrow \mathsf{SpendTx}((h, h_2), m, pk_A^{(c_1)}, pk_B^{(c_1)}, pk_C^{(c_1)})$

$\mathsf{tx}_2 \leftarrow \mathsf{RefundTx}(m, \mathsf{addr}, pk_A^{(c_1)})$

$\sigma_A \xleftarrow{\$} \mathsf{Sig.Sign}(pk_A^{(c_1)}, \mathsf{tx}_2)$

if $\mathsf{Sig.Verify}(pk_C^{(c_1)}, \mathsf{tx}_2, \sigma_C) = 0$ return $\perp$

if $\mathsf{Sig.Verify}(pk_B^{(c_1)}, \mathsf{tx}_2, \sigma_B) = 0$ return $\perp$

add tx_1 to c_1 blockchain

if $\mathsf{Sig.Verify}(pk_C^{(c_2)}, \mathsf{tx}_4, \sigma'_B) = 0$ return $\perp$

if $\mathsf{Sig.Verify}(pk_B^{(c_2)}, \mathsf{tx}_4, \sigma'_C) = 0$ return $\perp$

$\sigma'_A \xleftarrow{\$} \mathsf{Sig.Sign}(pk_A^{(c_2)}, \mathsf{tx}_4)$

before t_2: spend tx_3 using (x, c_1)

after t_2: refund tx_2 using $(\sigma_A, \sigma_B, \sigma_C)$

C(c, pk, σ, m)

if $\mathsf{Sig.Verify}(pk, m, \sigma) = 0$ return $\perp$

else return $\mathsf{Sig.Sign}(pk_C^{(c)}, m)$

B

$h_1 \leftarrow H(c_1)$

if $\mathsf{Sig.Verify}(pk_C^{(c_1)}, \mathsf{tx}_2, \sigma_A) = 0$ return $\perp$

if $\mathsf{Sig.Verify}(pk_A^{(c_1)}, \mathsf{tx}_2, \sigma_C) = 0$ return $\perp$

$\sigma_B \xleftarrow{\$} \mathsf{Sig.Sign}(pk_B^{(c_1)}, \mathsf{tx}_2)$

$(\mathsf{addr}, \mathsf{tx}_3) \leftarrow \mathsf{SpendTx}((h, h_1), n, pk_B^{(c_2)}, pk_A^{(c_2)}, pk_C^{(c_2)})$

$\mathsf{tx}_4 \leftarrow \mathsf{RefundTx}(n, \mathsf{addr}, pk_B^{(c_2)})$

$\sigma'_B \xleftarrow{\$} \mathsf{Sig.Sign}(pk_B^{(c_2)}, \mathsf{tx}_4)$

if $\mathsf{Sig.Verify}(pk_C^{(c_2)}, \mathsf{tx}_4, \sigma'_C) = 0$ return $\perp$

if $\mathsf{Sig.Verify}(pk_A^{(c_2)}, \mathsf{tx}_4, \sigma'_A) = 0$ return $\perp$

add tx_3 to c_2 blockchain

before t_1: spend tx_1 using (x, c_2)

after t_1: refund tx_4 using $(\sigma'_A, \sigma'_B, \sigma'_C)$

$\mathsf{tx}_2, \sigma_A, \sigma_C$
σ_C
σ_B
σ'_C
$\mathsf{tx}_4, \sigma'_B, \sigma'_C$
σ'_A
x

Fig. 5: A method for A and B to — with the approval of a third party C — exchange m units of currency c_1 for n units of currency c_2 in a fair manner; i.e., in a way such that if either A or B stops participating at any point in the interaction, the other party loses nothing.

Resources

- "Centrally Banked Cryptocurrencies" by George Danezis and Sarah Meiklejohn `https://eprint.iacr.org/2015/502`