SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Analysis of Bitcoin Backbone Protocol in the Non-Flat Model

Peifang NI[1,2,3], Hongda LI[1,2,3*] & Dongxue PAN[1,2,3]

[1]*State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing* 100093*, China;*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3]*Data Assurance and Communication Security Research Center, CAS, Beijing, China*

Appendix A The Bitcoin Backbone Protocol

In this section, we give the detailed description of bitcoin backbone protocol with parties of different computational power based on the model of [1].

Algorithm 1. This algorithm checks the validation of each block, called *chain validation*, in a given chain $\mathcal{C}$. First, it checks the validation of $X_{\mathcal{C}}$ with chain content validation predicate $V(\cdot)$, where $X_{\mathcal{C}}$ is the records of $\mathcal{C}$. Then, each block is connected properly and equipped with correct solution of puzzle with respect to hash functions $H(\cdot)$ and $G(\cdot)$. Towards to the current difficult target, each block with the proper computational power. If any of these validations fails, then $\mathcal{C}$ is discarded.

Algorithm 2. This algorithm aims to select a best local chain, called *chain comparison*, from a given set of valid chains. It is parameterized by a function $max(\cdot)$ that applies some ordering to the space of blockchains. Function $max(\mathcal{C}_1, \mathcal{C}_2)$ returns the chain with the most computational power and if $Com(\mathcal{C}_1) = Com(\mathcal{C}_2)$, then other selection criteria can be used.

Algorithm 3. This algorithm is used to extend a chain with hash functions $H(\cdot), G(\cdot)$ and target T called $proofs$-of-$work$. As described in Figure 1, party P_i with computational power C_i, best local chain $\mathcal{C}$ and input x, tries to find a solution by increasing counter ctr that satisfies $H(ctr, G(H(head(\mathcal{C}), x)) < T$.

The Backbone Protocol Π (algorithm 4). The above three algorithms allow us to describe the bitcoin backbone protocol. It is executed by the parties with different computational power, who are encouraged to solve computational puzzles and maintain a public log together. Each party maintains a local best valid chain $\mathcal{C}$ determined by algorithm 1 and algorithm 2, and tries to extend chain $\mathcal{C}$ via algorithm 3.

Algorithm A1 The $Chain\ Validation$ predicate with input chain $\mathcal{C}$, parameterized by C_i, T, two hash functions $G(\cdot), H(\cdot)$ and chain content validation predicate $V(\cdot)$.

Function validate ($\mathcal{C}$)
  b$\leftarrow$ V($X_{\mathcal{C}}$)
  if $b \wedge (\mathcal{C} \neq \epsilon)$ **then** ▷ The chain is non-empty and meaningful w.r.t. $V(\cdot)$
    $\langle h, x, ctr, TS\rangle \leftarrow head(\mathcal{C})$ and $h' = H(ctr, G(h, x))$
    repeat
      $\langle h, x, ctr, TS\rangle \leftarrow head(\mathcal{C})$
      if $validblock^T(\langle h, x, ctr, TS\rangle \leftarrow head(\mathcal{C})) \wedge (ctr \leqslant C_i) \wedge (h' < T))$
        then, $h' \leftarrow h$, $\mathcal{C} \leftarrow \mathcal{C}^{\lceil 1}$ ▷ Remove the head from $\mathcal{C}$
      else, $b \leftarrow False$
      end if
    until$(\mathcal{C} = \epsilon) \vee (b = False)$
  end if
  return b
end function

* Corresponding author (email: lihongda@iie.ac.cn)

Algorithm A2 The *Chain Comparison* predicate with input $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$, parameterized by function $max(\cdot)$.

Function $maxvalid(\{\mathcal{C}_1, ..., \mathcal{C}_k\})$
  $temp \leftarrow \epsilon$
  for $i = 1$ to k **do**
    if $validate(\mathcal{C}_i)$, **then**
      $temp \leftarrow max(\mathcal{C}_i, temp)$
    end if
  end for
  return $temp$
end function
$max(\mathcal{C}_1, \mathcal{C}_2)$: outputs the chain with the most amount of computational power

Algorithm A3 The *proof of work* function with input $(x, \mathcal{C})$, parameterized by T, C_i and two hash functions $H(\cdot), G(\cdot)$.

Function $pow(x, \mathcal{C})$
  if $\mathcal{C} = \epsilon$ **then** ▷ Determine proof of work instance
    $h \leftarrow 0$
  else
    $\langle h', x', ctr', TS' \rangle \leftarrow head(\mathcal{C})$ and $h = H(ctr', G(h', x'))$
  end if
  $b \leftarrow 1; B \leftarrow \epsilon$
  while $b \leqslant C_i$ **do** ▷ The number of queries to $H(\cdot)$
    $ctr \leftarrow b$
    if $(H(ctr, G(h, x)) < T)$ **then**
      $B \leftarrow \langle h, x, ctr, TS \rangle$, $\mathcal{C} \leftarrow \mathcal{C}B$ ▷ Extend chain
      $C_i \leftarrow C_i - b$; $b \leftarrow 1$ ▷ The party continues to mine the next block
    else $b \leftarrow b + 1$ ▷ The party continues to mine the block
  end while
  return $\mathcal{C}$
end function

Algorithm A4 The bitcoin backbone protocol in the non-flat model with local state $(h, \mathcal{C})$, parameterized by input function $I(\cdot)$ and chain reading function $R(\cdot)$.

$\mathcal{C} \leftarrow \epsilon$ **then**
$h \leftarrow 0$ and round $\leftarrow 1$
$\mathcal{C}' \leftarrow maxvalid(\mathcal{C}, the\ chains\ found\ in\ Receive())$ ▷ Choose the local best chain
  if Input() contains Read **then** ▷ Create the necessary output
    write $R(\mathcal{C}')$ to Output()
  end if
  while True **do**
    $\langle h, x \rangle \leftarrow I(h, \mathcal{C}', round, Input(), Receive())$ ▷ Determine the x-value
    $\mathcal{C}_{new} \leftarrow pow(x, \mathcal{C}')$ ▷ Extend chain
    if $\mathcal{C}' \neq \mathcal{C}_{new}$ **then**
      $\mathcal{C} \leftarrow \mathcal{C}_{new}$ and Diffuse $(\mathcal{C})$
    else, Diffuse($\perp$)
    end if
    round $\leftarrow$ round+1
  end while

References

1 Garay J , Kiayias A , Leonardos N . The Bitcoin Backbone Protocol: Analysis and Applications[J]. 2015.