

Consensus Algorithms in Wireless Blockchain System

1 Consensus Algorithm in Each Round

Algorithm 1 Utilities of Node v

```
1: function BLOCKLEADER( $seed, w_v, W$ )  $\triangleright$  Choose a Leader
2:   ( $value, proof$ ) =  $VRF(sk_v, seed)$ 
3:    $p_v = \frac{w_v}{W}$ 
4:    $ID = 0$ 
5:   if  $\frac{value}{2^{bit(value)}} \notin [\sum_{k=0}^i p_k, \sum_{k=0}^{i+1} p_k)$  then
6:      $ID++$ 
7:   end if
8:   return  $ID, value, Proof$ 
9: end function
10: function VERIFYLEADER( $pk_v, ID, value, proof, seed, w_v, W$ )  $\triangleright$  Verify the validation of
    leader
11:    $result = VerifyVRF(pk_v, value, proof)$ 
12:   if  $result == 0$  or  $\frac{value}{2^{bit(value)}} \notin [\sum_{k=0}^{ID} p_k, \sum_{k=0}^{ID+1} p_k)$  then
13:     return False
14:   else
15:     return True
16:   end if
17: end function
18: function GENERATEBLOCK( $BC_v, Txs$ )  $\triangleright$  Create a new block
19:    $B_v^{new}.Index \leftarrow Length(BC_v)$ 
20:    $B_v^{new}.Prehash \leftarrow BC_v^{latest}.Hash$ 
21:    $B_v^{new}.txs \leftarrow Txs$ 
22:    $B_v^{new}.Timestamp \leftarrow Timestamp$ 
23:    $B_v^{new}.Hash \leftarrow Hash(B_v^{new})$ 
24:   return  $B_v^{new}$ 
25: end function
26: function APPEND( $BC_v, B_v^{new}$ )  $\triangleright$  Append the new block
27:   return  $BC_v \leftarrow BC_v + B_v^{new}$ 
28: end function
29: function VALIDBLOCK( $BC_v, B_v^{new}$ )  $\triangleright$  verify the validation of new block
30:   if  $B_v^{new}.Prehash \neq BC_v^{latest}.Hash$  then
31:     return False
32:   else if  $ValidTransactions(B_v^{new}.txs) == False$  then
33:     return False
34:   else if  $VerifyLeader(pk_v, ID, value, proof, seed, w_v, W) == False$  then
35:     return False
36:   else
37:     return True
38:   end if
39: end function
```

```

40: function VALIDTRANSACTIONS( $Txs$ )
41:   for  $tx \in Txs$  do
42:     if  $tx$  is Invalid then
43:       return False
44:     end if
45:   end for
46:   return True
47: end function
48: function MSGB( $BC_v, B_v^{new}, w_v, ID, value, proof$ )
49:    $m_B.latest \leftarrow BC_v^{latest}$ 
50:    $m_B.B_v^{new} \leftarrow B_v^{new}$ 
51:    $m_B.sortition \leftarrow \{w_v, ID, value, proof\}$ 
52:   return  $m_B$ 
53: end function

```

Algorithm 2 Consensus Algorithm

```

1:  $\triangleright$  Initialization:
2:  $\triangleright$  Leader Election and Block Propocal:
3:  $(BL, value, Proof) = BlockLeader(seed)$ 
4: if  $v == BL$  then
5:    $B_v^{new} = GenerateBlock(BC_v, Txs)$ 
6:    $m_B = MSGB(BC_v, B_v^{new}, w_v, value, proof)$ 
7:    $broadcast(m_B)$ 
8: end if
9:  $\triangleright$  Block Verification and Finalization:
10: if  $isValid(B_v^{new}) \&\& VerifyLeader(pk_v, seed) == true$  then
11:    $Append(BC_v, B_v^{new})$ 
12: end if
13: function BLOCKLEADER( $seed, w_v, W$ )  $\triangleright$  Choose a Leader
14:    $(value, proof) = VRF(sk_v, seed)$ 
15:    $p_v = \frac{w_v}{W}$ 
16:    $ID = 0$ 
17:   if  $\frac{value}{2^{bit(value)}} \notin [\sum_{k=0}^i p_i, \sum_{k=0}^{i+1} p_i)$  then
18:      $ID++$ 
19:   end if
20:   return  $ID, value, Proof$ 
21: end function
22: function GENERATEBLOCK( $BC_v, Txs$ )  $\triangleright$  Create a new block
23:    $B_v^{new}.Index \leftarrow Length(BC_v)$ 
24:    $B_v^{new}.Prehash \leftarrow BC_v^{latest}.Hash$ 
25:    $B_v^{new}.txs \leftarrow Txs$ 
26:    $B_v^{new}.Timestamp \leftarrow Timestamp$ 
27:    $B_v^{new}.Hash \leftarrow Hash(B_v^{new})$ 
28:   return  $B_v^{new}$ 
29: end function
30: function APPEND( $BC_v, B_v^{new}$ )  $\triangleright$  Append the new block
31:   return  $BC_v \leftarrow BC_v + B_v^{new}$ 
32: end function
33: function VALIDBLOCK( $BC_v, B_v^{new}$ )  $\triangleright$  verify the validation of new block
34:   if  $B_v^{new}.Prehash == BC_v^{latest}.Hash \&\& valid(B_v^{new}.txs)$  then
35:     if  $VerifySortition(pk_v, value, proof, seed, w_v, W)$  then

```

```

36:         return true
37:     end if
38: end if
39: return  $BC_v \leftarrow BC_v + B_v^{new}$ 
40: end function

```
