SUSTech Southern University of Science and Technology

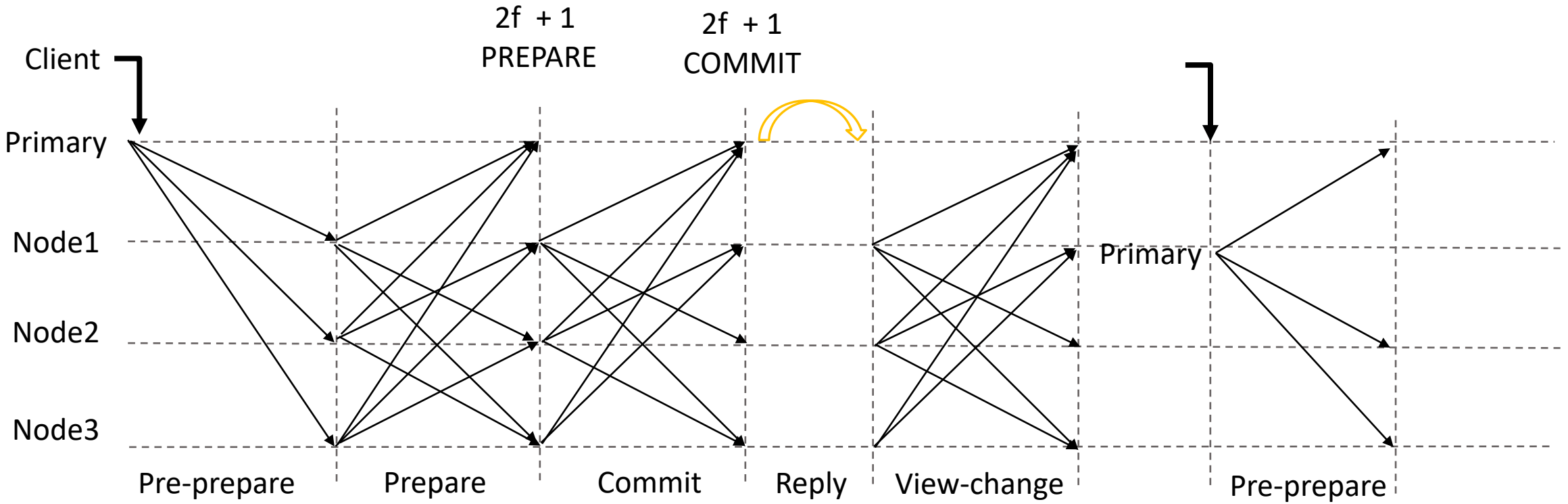
UNIVERSITY OF BIRMINGHAM | CENTRE FOR CYBER SECURITY AND PRIVACY

From PBFT to POA

RUJIA LI

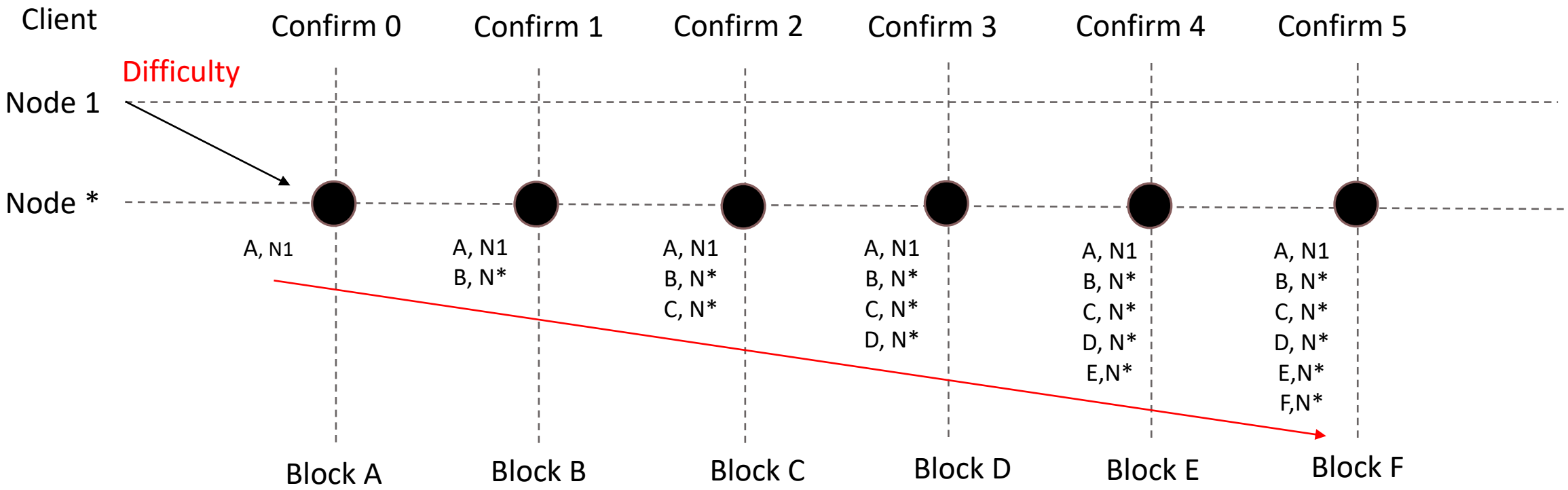
2021/12/06

Practical byzantine fault tolerance (PBFT)



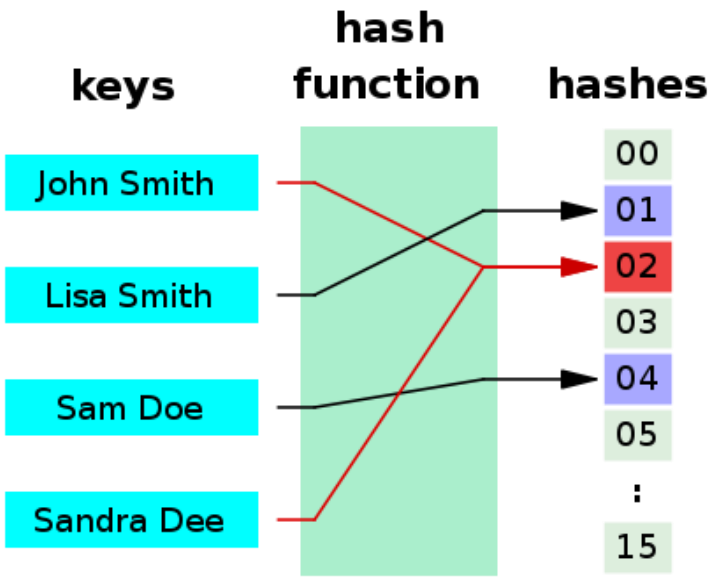
Lamport, Leslie, Robert Shostak, and Marshall Pease. (1982) "THE BYZANTINE GENERALS PROBLEM."
Castro, M., & Liskov, B. (1999, February). Practical byzantine fault tolerance. In OSDI (Vol. 99, No. 1999, pp. 173-186).

Proof of Work (POW)



Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Mining Difficulty

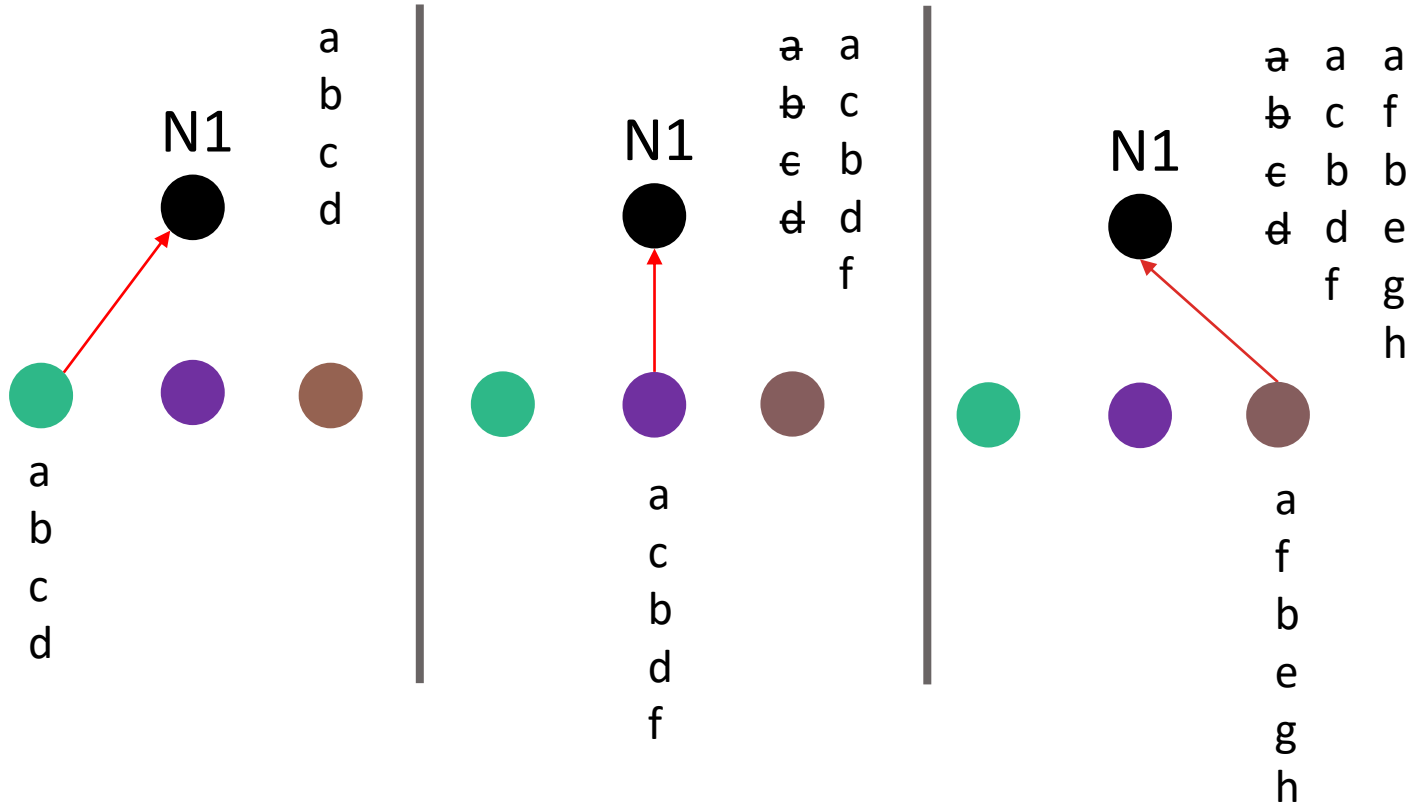


- Mining difficulty is a measurement unit used in the process of Bitcoin mining
- Difficulty indicates how difficult it is to solve a complex cryptographic puzzle
- The difficulty of mining new units increases or decreases over time, depending on the number of miners in the network
- Increases in difficulty are necessary in order to keep the target block time

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

GHOST Protocol

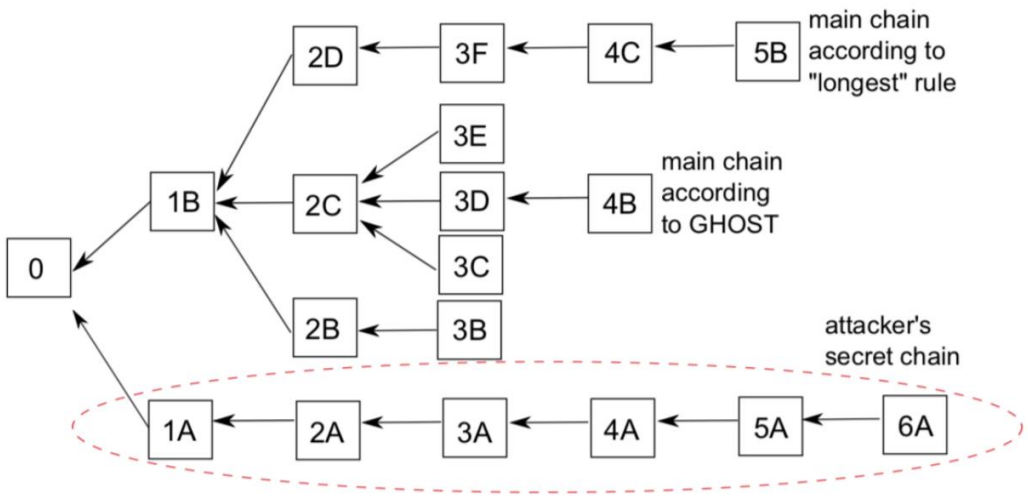
Sompolinsky, Y., & Zohar, A. (2015, January). Secure high-rate transaction processing in bitcoin. In *International Conference on Financial Cryptography and Data Security* (pp. 507-527). Springer, Berlin, Heidelberg.



GHOST Protocol

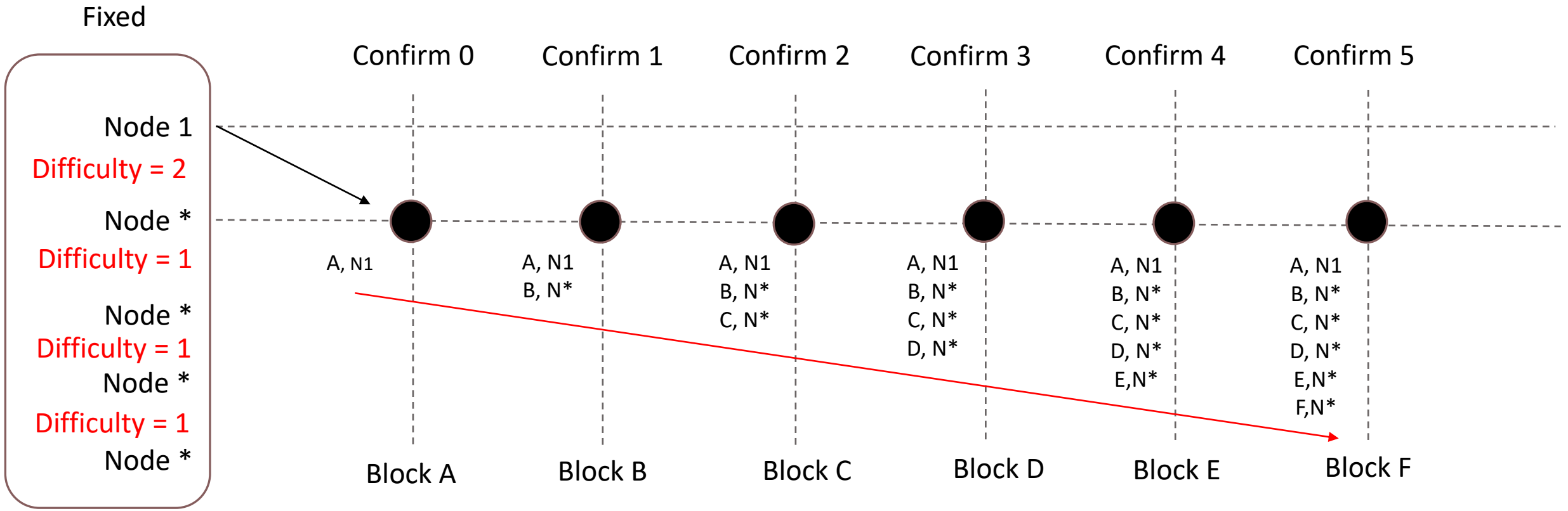
```
// If the total difficulty is higher than our known, add it to the canonical chain
// Second clause in the if statement reduces the vulnerability to selfish mining.
// Please refer to http://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf
if externTd.Cmp(localTd) > 0 || (externTd.Cmp(localTd) == 0 && mrand.Float64() < 0.5) {
        // Delete any canonical number assignments above the new head
        for i := number + 1; ; i++ {
                hash := GetCanonicalHash(hc.chainDb, i)
                if hash == (common.Hash{}) {
                        break
                }
                DeleteCanonicalHash(hc.chainDb, i)
        }
        // Overwrite any stale canonical number assignments
        var (
                headHash   = header.ParentHash
                headNumber = header.Number.Uint64() - 1
                headHeader = hc.GetHeader(headHash, headNumber)
        )
        for GetCanonicalHash(hc.chainDb, headNumber) != headHash {
                WriteCanonicalHash(hc.chainDb, headHash, headNumber)

                headHash = headHeader.ParentHash
                headNumber = headHeader.Number.Uint64() - 1
                headHeader = hc.GetHeader(headHash, headNumber)
        }
        // Extend the canonical chain with the new header
        if err := WriteCanonicalHash(hc.chainDb, hash, number); err != nil {
                log.Crit("Failed to insert header number", "err", err)
        }
```

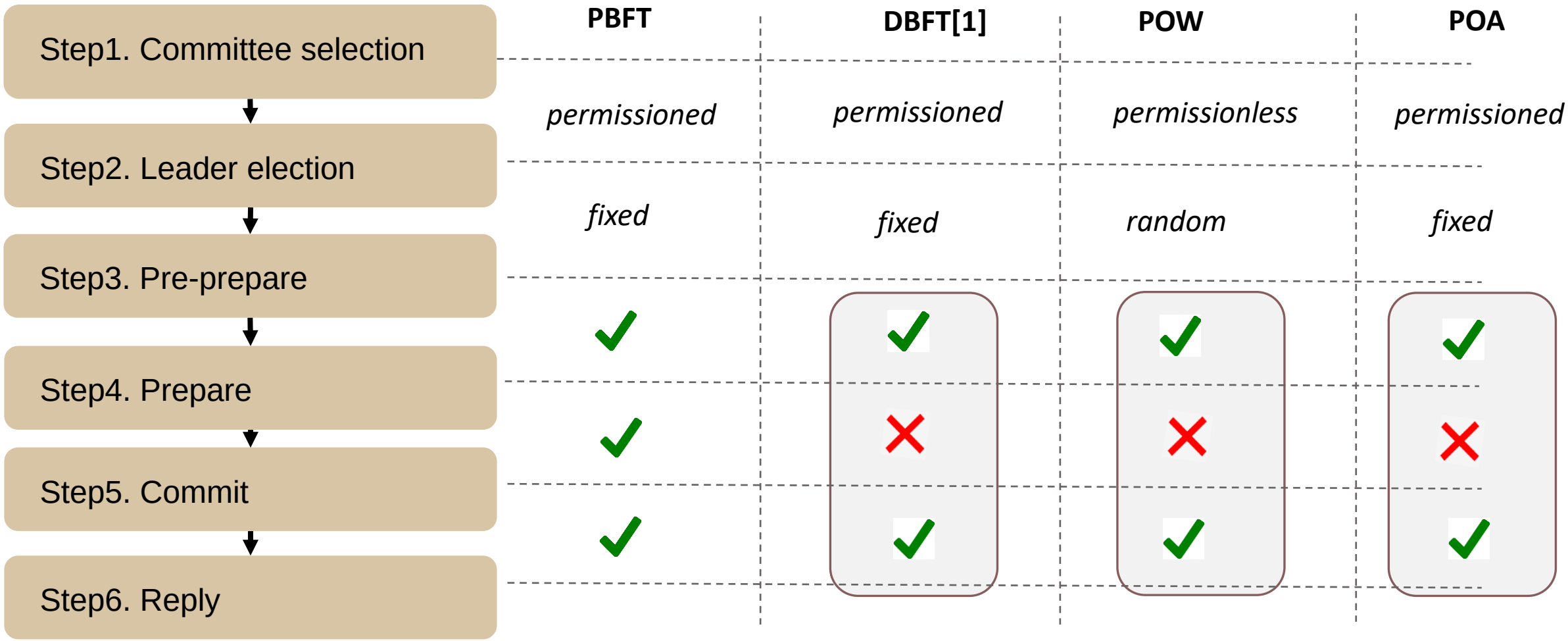


Sompolinsky, Y., & Zohar, A. (2015, January). Secure high-rate transaction processing in bitcoin. In International Conference on Financial Cryptography and Data Security (pp. 507-527). Springer, Berlin, Heidelberg.

Proof of Authority (POA Clique)



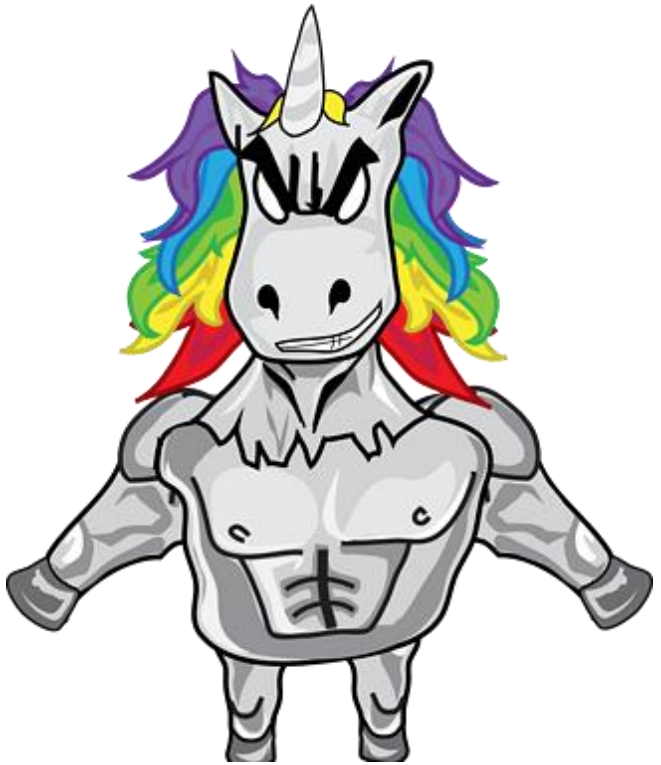
Comparison



[1] Wang, Q., Yu, . (2020, February). Security Analysis on dBFT protocol of NEO. In FC (pp. 20-31). Springer, Cham.

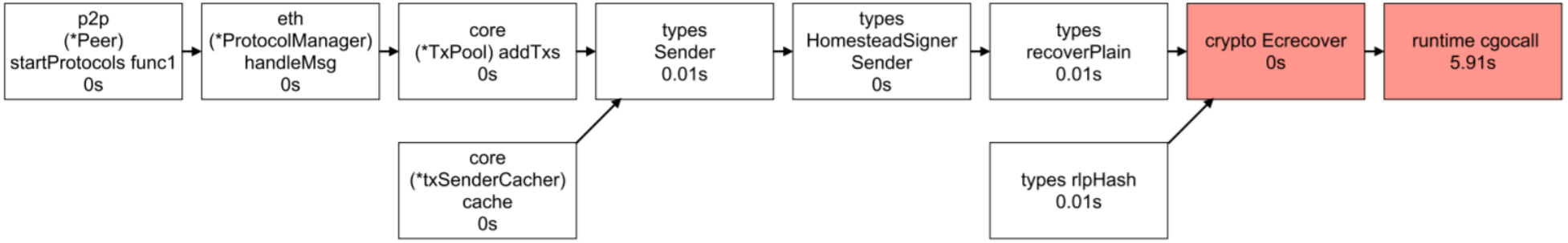
Project	*Client*	*Location*	*Market Cap*
HPB	Clique-variant	No.xxx	$ xxxx
Go-Ethereum	Clique	-	$461,322,853,481
Binance-chain	Clique-forked	No.xx	$88,094,641,014
Polygon (MATIC)	Clique-forked	No.xx	$12,428,536,101
Openethereum	Clique-forked	No.xx	$461,322,853,481
PoA network	Clique-forked	No.658	$129,657,466
Ethereum Classic	Clique-forked	No.463	$4,659,922,751
ConsenSys	Clique-forked	No.463	-
GoChain	Clique-forked	No.315	$35,899,700
Daisy	Clique-forked	No.305	-
Olecoin	Clique-forked	No.293	-
EEX	Clique-forked	-	-
AplaProject	Clique-forked	-	-
HPB	Clique-forked	No.126	$10,128,116
Tomochain	Clique-forked	No.746	$207,899,832

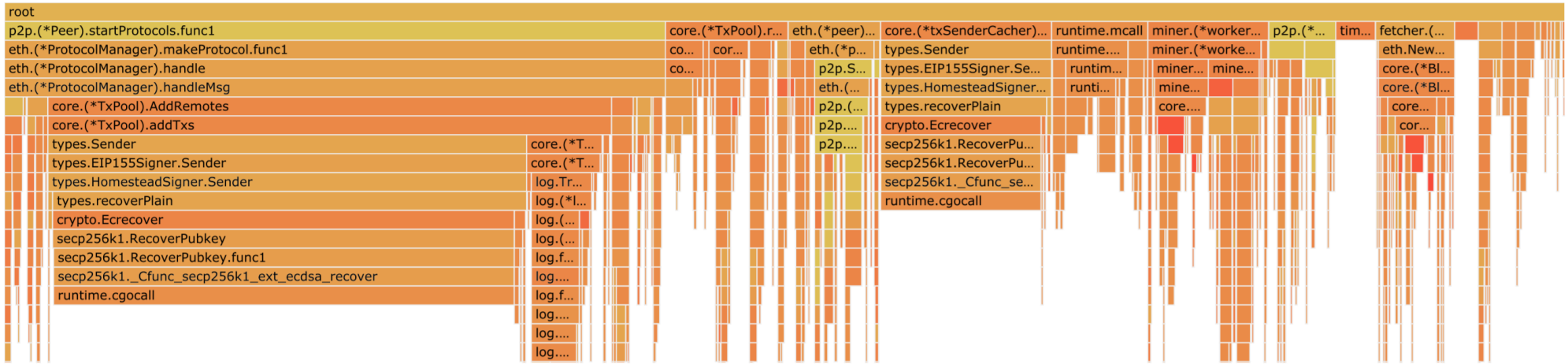
[1] Data accessed in November, 2021.
[2] Projects in Yellow are vulnerable to the attack .
[3] Projects without the background can resist the attack.



Market Capitalization

POA Performance bottleneck





Toyoda, K., Machi, K., Ohtake, Y., & Zhang, A. N. (2020). Function-level bottleneck analysis of private proof-of-authority ethereum blockchain. *IEEE Access*, *8*, 141611-141621.

Thanks!