

# HOLISTIC VERIFICATION OF BLOCKCHAIN CONSENSUS

Nathalie Bertrand, Vincent Gramoli, Igor Konnov, Marijana Lazic,  
Pierre Tholoniati, Josef Widder



THE UNIVERSITY OF  
SYDNEY



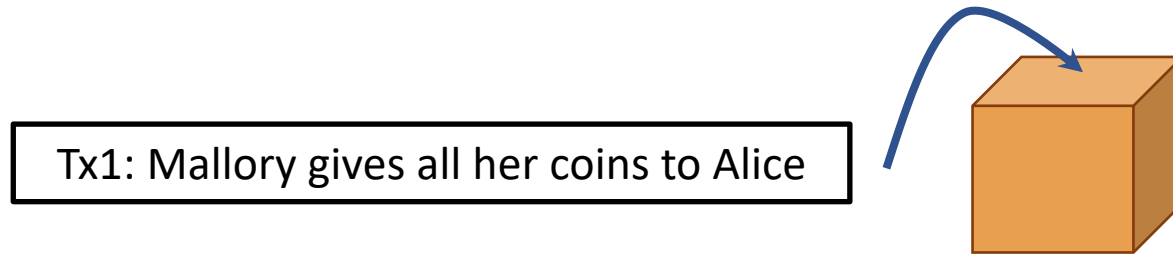
Technische  
Universität  
München

informat



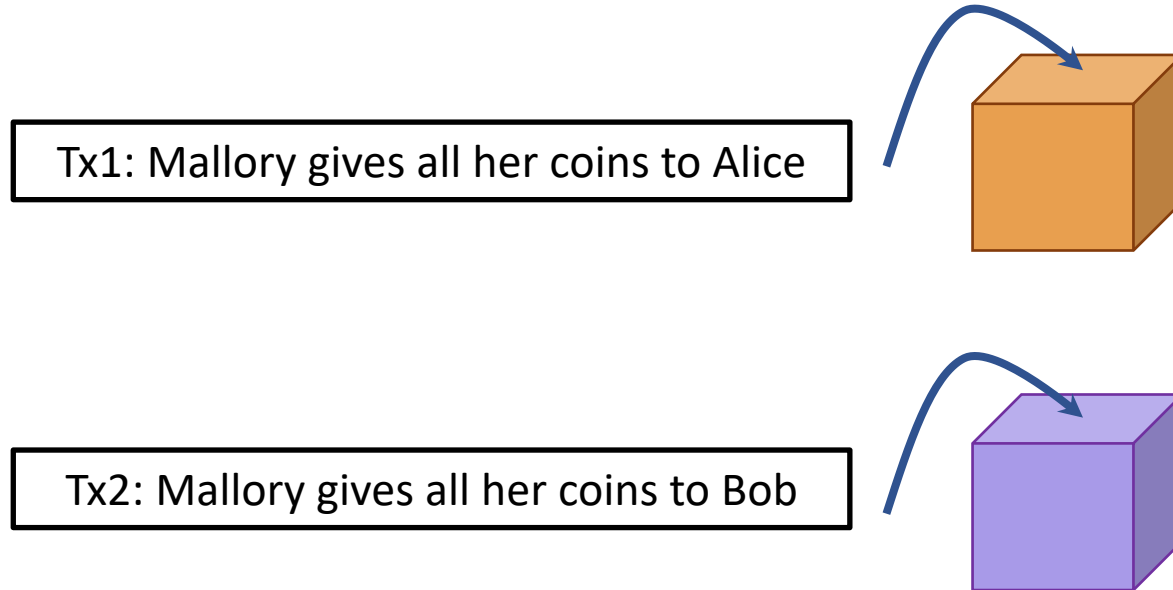
EPFL

# Blockchain

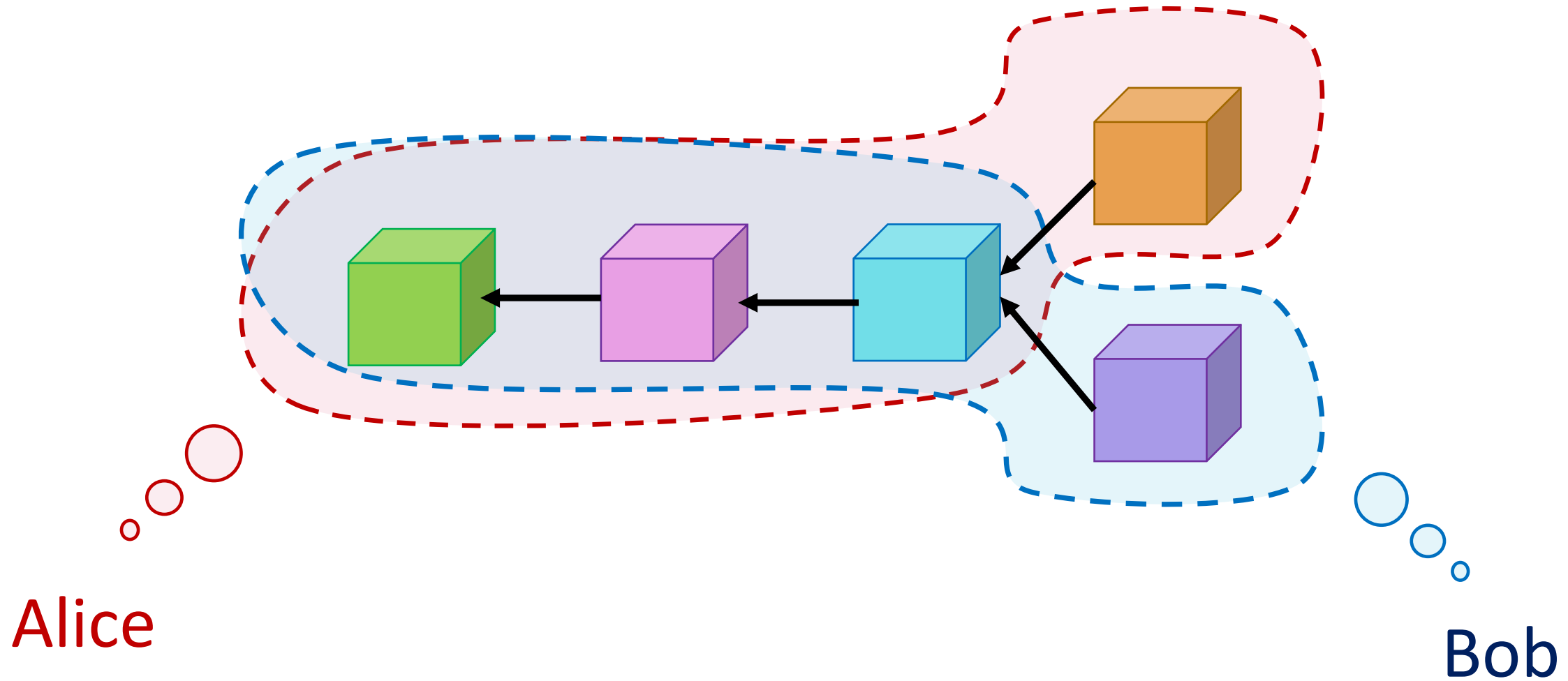


# Double Spending

Mallory



# Loss of Assets



Loss of A

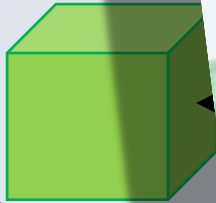
Bitcoin.com

Aug 6, 2020

SECURITY

by Terence Zimwara

# **\$5.6 Million Double Spent: ETC Team Finally Acknowledges the 51% Attack on Network**



Alice



# Loss of A

## Alice

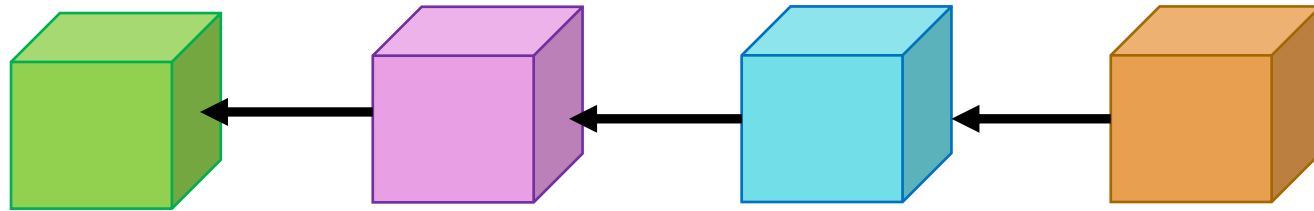
Vincent Gramoli

The image shows a screenshot of a Bitcoin.com article. The article is titled "Bitcoin Gold Gets \$18 Million Haircut" and is dated August 6, 2020. The author is Terence Zimwara. The article discusses a security issue where an unknown party with access to very large amounts of hashpower is trying to use '51% attacks'. Bitcoin Gold forum poster Mental Nomad announced a week ago, "to perform 'double spend' attacks to steal money from Exchanges. We have been advising all exchanges to increase confirmations and carefully review large deposits." The article also mentions that a founding economic principle of bitcoin was its alleviation of the double spend problem. It was a main stumbling block in the historical race to create a viable cryptographic monetary form – foiling a great many coders along the way. Satoshi Nakamoto solved it through a decentralized, distributed ledger confirmation process (blockchain). Going as far back as its genesis block from early 2009, users can be confident transactions aren't rebroadcast. Like clockwork, 6 times an hour, blocks are added – copied to nodes within the universal network.

At the bottom of the article, there is a QR code and a table with the following data:

Transactions	76	1001 BTG

# The Need for Byzantine Consensus



Alice

Bob

# Impact of Human Errors





# Red Belly Blockchain [IEEE S&P'21]

- Builds upon DBFT, a partially synchronous consensus protocol
- Can scale, its performance increasing with the system size
- Peaks at 660,000 transactions per second
- Deployed on 1000 machines across 4 continents
- Achieves 3 second finality (commit time)

# Holistic Verification of DBFT

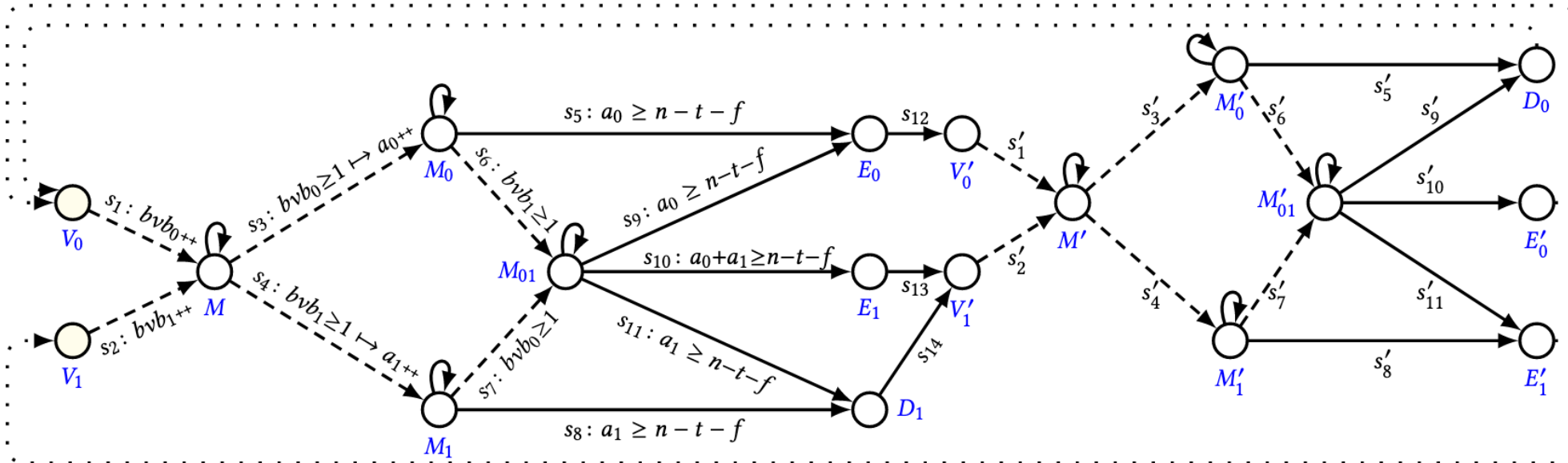
DBFT  
[NCA'18]

1. Fairness

```

1: Global scope variable:
2:  $contestants \subseteq \{0, 1\}$ , set of binary values, initially 0.
3: propose( $est$ ):
4:    $r \leftarrow 0$ 
5:   repeat:
6:     bv-broadcast( $est$ ,  $\langle est, i \rangle$ )
7:     wait until ( $contestants \neq 0$ )
8:     broadcast( $AUX$ ,  $\langle contestants, i \rangle$ )  $\rightarrow$  favorites
9:     wait until  $\exists c_1, \dots, c_{n-t} : \forall 1 \leq j \leq n-t$  favorites[ $c_j$ ]  $\neq 0$ 
10:     $\wedge$  ( $qualifiers \leftarrow \cup_{1 \leq j \leq n-t} favorites[c_j] \subseteq contestants$ )
11:    if  $qualifiers = \{v\}$  then
12:       $est \leftarrow v$ 
13:      if  $v = (r \bmod 2)$  then decide( $v$ )
14:    else  $est \leftarrow (r \bmod 2)$ 
15:     $r \leftarrow r + 1$ 

```



2. Alg. to TA  
[PODIS'17]

3. DBFT TA  
Specification

3. LTL Problem  
Specification

ByMC  
[POPL 2017]

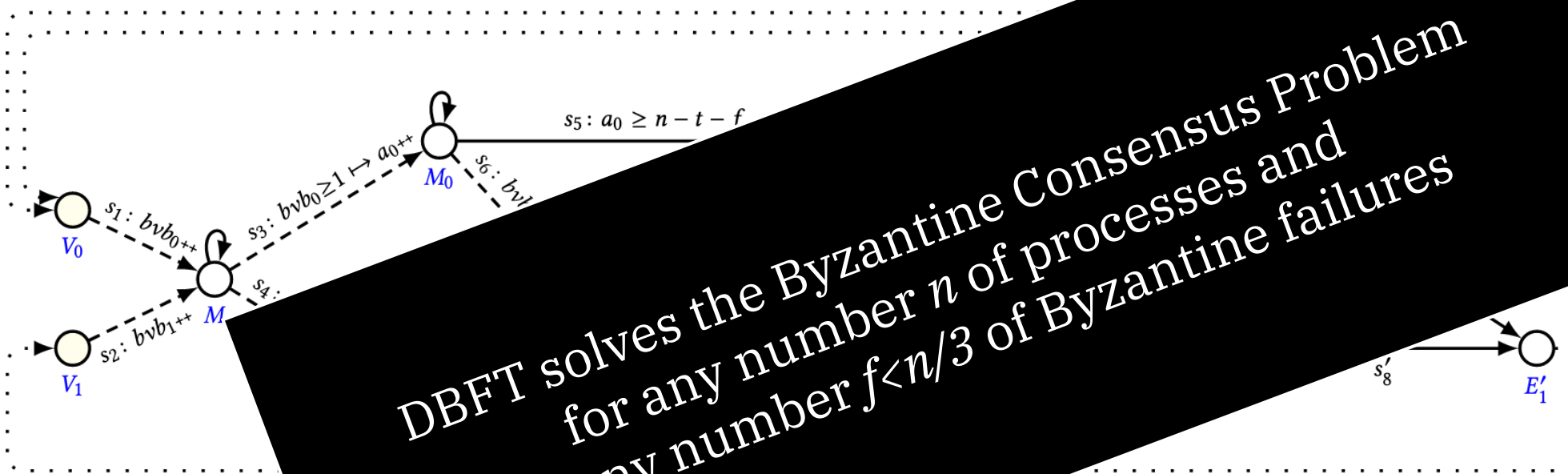
## Byzantine Consensus Problem

$\forall R \in \mathbb{N}, \forall R' \in \mathbb{N} \left( \Diamond \kappa[D_v, R] \neq 0 \Rightarrow \Box \kappa[D_{1-v}, R'] = 0 \right)$  ( $Agree_v$ )

$\forall R \in \mathbb{N} \left( \kappa[V_v, 1] = 0 \Rightarrow \Box \kappa[D_v, R] = 0 \right)$  ( $Valid_v$ )

$\exists R \in \mathbb{N} \left( \Box (\kappa[E_0, R] = 0 \wedge \kappa[E_1, R] = 0) \vee \Box (\kappa[E'_0, R] = 0 \wedge \kappa[E'_1, R] = 0) \right)$  ( $Term$ )

# Holistic Verification of DBFT



DBFT solves the Byzantine Consensus Problem  
for any number  $n$  of processes and  
for any number  $f < n/3$  of Byzantine failures

[NCA']

```

1: Global scope variable:  

2:    $\text{constants} \subseteq \{0, 1\}$ , set of binary values, initially  $\emptyset$ .  

3: propose( $\text{est}$ ):  

4:    $r \leftarrow 0$   

5:   repeat:  

6:      $\text{bv-broadcast}(\text{est}, \langle \text{est}, t \rangle)$   

7:     wait until  $\langle \text{constants} \neq \emptyset \rangle$   

8:      $\text{broadcast}(\text{AUX}, \langle \text{constants}, t \rangle) \rightarrow \text{favorites}$   

9:     wait until  $\exists c_1, \dots, c_{n-t} : \forall 1 \leq j \leq n-t \text{ favorites}[c_j] \neq \emptyset$   

10:     $\wedge (\text{qualifiers} \leftarrow \cup_{1 \leq j \leq n-t} \text{favorites}[c_j]) \subseteq \text{constants}$   

11:    if  $\text{qualifiers} = \{v\}$  then  

12:       $\text{est} \leftarrow v$   

13:      if  $v = (r \bmod 2)$  then  $\text{decide}(v)$   

14:    else  $\text{est} \leftarrow (r \bmod 2)$   

15:   $r \leftarrow r + 1$ 

```

2. Alg. to TA  
[LOPDIS'17]

### 3. DBFT TA Specification

### 3. LTL Problem Specification

ByMC  
[POPL 2017]

Byzantine C	
$\forall R \in \mathbb{N}, \forall R' \in \mathbb{N} \left( \kappa[V_v, R] \wedge \kappa[V_v, R'] = 0 \Rightarrow \kappa[D_{1-v}, R'] = 0 \right)$	$(Agree_v)$
$\forall R \in \mathbb{N} \left( \kappa[V_v, 1] \wedge \kappa[V_v, R] = 0 \Rightarrow \kappa[D_v, R] = 0 \right)$	$(Valid_v)$
$\exists R \in \mathbb{N} \left( \square \left( \kappa[E_0, R] \wedge \kappa[E_1, R] = 0 \right) \vee \square \left( \kappa[E'_0, R] = 0 \wedge \kappa[E'_1, R] = 0 \right) \right)$	$(Term)$

# Holistic Verification

## Holistic Verification of Blockchain Consensus

Nathalie Bertrand ✉  
INRIA Rennes, France

Vincent Gramoli ✉  
University of Sydney, Australia

Igor Konnov ✉  
Informal Systems, Vienna, Austria

Marijana Lazić ✉  
TU Munich, Germany

Pierre Tholoniati ✉  
Columbia University, New York, USA

Josef Widder ✉  
Informal Systems, Vienna, Austria

### Abstract

Blockchain has recently attracted the attention of the industry due, in part, to its ability to automate asset transfers. It requires distributed participants to reach a consensus on a block despite the presence of malicious (a.k.a. Byzantine) participants. Malicious participants exploit regularly weaknesses of these blockchain consensus algorithms, with sometimes devastating consequences. In fact, these weaknesses are quite common and are well illustrated by the flaw in various blockchain consensus algorithms. Paradoxically, until now, no blockchain consensus has been holistically verified using model checking.

In this paper, we remedy this paradox by model checking for the first time a blockchain consensus used in industry. We propose a holistic approach to verify the consensus algorithm of the Red Belly Blockchain [20], for any number  $n$  of processes and any number  $f < n/3$  of Byzantine processes. We decompose directly the algorithm pseudocode in two parts—an inner broadcast algorithm and an outer decision algorithm—each modelled as a threshold automaton [36], and we formalize their expected properties in linear-time temporal logic. We then automatically check the inner algorithm, under a carefully identified fairness assumption. For the verification of the outer algorithm, we simplify the model of the inner algorithm by relying on its checked properties. We finally verify not only the safety properties of the Red Belly Blockchain consensus algorithm, but also its liveness, in less than 70 seconds.

Computing methodologies → Distributed algorithms

fairness

2. Pseudocode  
to TA

Specification

```
1: Global scope variable:
2: contestants  $\subseteq \{0, 1\}$ , set of binary values, initially 0.

3: propose(est):
4:    $r \leftarrow 0$ 
5:   repeat:
6:     bv-broadcast(est, (est, i))
7:     wait until (contestants  $\neq \emptyset$ )
8:     broadcast(AUX, (contestants, i))  $\rightarrow$  favorites
9:     wait until  $\exists c_1, \dots, c_{n-t} : \forall 1 \leq j \leq n-t$  favorites[cj]  $\neq \emptyset$ 
10:         $\wedge$  (qualifiers  $\leftarrow \cup_{1 \leq j \leq n-t}$  favorites[cj])  $\subseteq$  contestants
11:   if qualifiers = {v} then
12:     est  $\leftarrow v$ 
13:     if  $v = (r \bmod 2)$  then decide(v)
14:   else est  $\leftarrow (r \bmod 2)$ 
15:    $r \leftarrow r + 1$ 
```

ByMC,  
POPL 2017]



Vincent.Gramoli@sydney.edu.au



THE UNIVERSITY OF  
**SYDNEY**



# Relater Work

- Theorem prover were used to prove parts of Stellar (not its quorum system) [DISC'19]
- The safety of Byzantine Paxos was proved by refinements (not its liveness) [DISC'11]
- Symbolic model checkers proved algorithms for fixed number of processes (n=10) [Dist. Comp. 2011]
- TLA+ model checker TLC was used for consensus with benign faults [RP'19]
- Bosco is a fast-path for consensus whose safety was proven with ByMC, but it needs to rely on another Byzantine consensus protocol [OPODIS'17]