Game Theoretical Framework for Analyzing Blockchains Robustness

Paolo Zappalà, Marianna Belotti, Maria Potop-Butucaru, Stefano Secci

Game theoretical framework for analyzing blockchains robustness

P. Zappalà [*], M. Belotti [†], M. Potop-Butucaru [‡], S. Secci[§]

February 16, 2021

Abstract

Blockchains systems evolve in complex environments that mix classical patterns of faults (e.g crash faults, transient faults, Byzantine faults, churn) with selfish, rational or irrational behaviors typical to economic systems. In this paper we propose a game theoretical framework in order to formally characterize the robustness of blockchains systems in terms of resilience to rational deviations and immunity to Byzantine behaviors. Our framework includes necessary and sufficient conditions for checking the immunity and resilience of games and an original technique for composing games that preserves the robustness of individual games. We prove the practical interest of our formal framework by characterizing the robustness of various blockchain protocols: Bitcoin (the most popular permissionless blockchain), Tendermint (the first permissioned blockchain used by the practitioners), Lightning Network, a side-chain protocol and a cross-chain swap protocol. For each one of the studied protocols we identified upper and lower bounds with respect to their resiliency and immunity (expressed as no worse payoff that the initial state) face to rational and Byzantine behaviors.

1 Introduction

Distributed Ledger Technologies (DLTs) allow sharing a ledger of transactions among multiple users forming a peer-to-peer (P2P) network. DLTs characterized by a block architecture are called "blockchains". They enable its users to transfer cryptoassets in a decentralized manner by means of modular protocols adopted by the users themselves. Beyond the traditional blockchain architectures (i.e., *layer-1 protocols*) [7, 12, 19, 27, 29, 46, 60], the literature proposes other protocols that respectively define and regulate interactions in an overlaying network (*layer-2 protocols* [30]) and interactions between different blockchains (*cross-chain protocols* [17]). Each of these protocols establishes the instructions that a user must follow in order to interact with or through a blockchain. In a Blockchain system players can be classified in three different categories as stated in [4]: (i) players who follow the prescribed protocol i.e., *altruistic*, (ii) those who act in order to maximise their own benefit i.e., *rational* and, (iii) players who may rationally deviate from the prescribed protocol i.e., *rational Byzantine*. The latest category can be redefined, according to [36], to include any possible arbitrary protocol deviation (including irrational).

Interactions among users are modeled with game theory, which is used to design incentive mechanisms aiming at preventing any possible deviation from a prescribed protocol that blockchain users need to follow [42]. As P2P systems, blockchains foresee the possibility for users to form coalition and to cooperatively deviate from a prescribed protocol. Robustness of protocols governing DLTs (e.g., consensus protocols, communication protocols and storage protocols) has been addressed in several recent works. Most of the existing literature addresses protocols characterizing specific

[*] Paolo Zappalà is with Orange Labs, 92320 Chatillon, France and LIA, Avignon Université, 84029 Avignon, France (e-mail: paolo.zappala@orange.com)

[†] Marianna Belotti is with Cedric, Cnam, 75003 Paris, France, and also with Département de la Transformation Numérique, Caisse des Dépôts, 75013 Paris, France (e-mail: marianna.belotti@caissedesdepots.fr).

[‡] Maria Potop-Butucaru is with Lip6, CNRS UMR 7606, Sorbonne University, 75005 Paris, France (e-mail: maria.potop-butucaru@lip6.fr).

[§] Stefano Secci is with Cedric, Cnam, 75003 Paris, France (e-mail: stefano.secci@cnam.fr).

blockchain implementations focusing on the agreement mechanisms necessary to validate blocks of transactions. Game theoretical frameworks are introduced in [57, 59] to analyse security aspects and incentive compatibility of Nakamoto's consensus protocol (i.e., Proof-of-Work [44]) characterizing the very first blockchain implementation known as Bitcoin. Users participating to the consensus mechanism (i.e., miners) are considered as individually rational [54] moved by the mere intention to increase their revenues i.e., the rewards earned form the mining activities [14, 55]. Authors in [15, 21, 23, 24, 25, 35, 56] adopt different utility functions for miners and pools that consider costs and relative rewards. Concerning layer-2 and cross-chain protocols, game theoretical analysis are carried out by [8, 9, 13, 32]. These analyses are strictly specific to the particular deployment context than to a generic blockchain. Most of the game theoretical models adopted to design secure and robust blockchain protocols, surveyed in [41], (i) address protocols characterizing specific blockchain implementations, (ii) analyze miners' behaviours in the consensus phase and (iii) adopt Nash Equilibria as solution concept.

In the literature, analysis of systems robustness with respect to participating actors can be classified according to the agents' nature [4]. Concerning rational agents, the robustness analysis include the study of the equilibria and the evaluation of their properties. The most studied and adopted solution concept in the literature is Nash Equilibrium, i.e., a strategy profile in which no player has interest in individually deviating from her own strategy. A first approach to the analysis of robustness is to compare Nash Equilibria, through indices such *Price of Byzantine Anarchy*, *Price of Malice* [43], and *Price of Anarchy* [38, 53]. This approach summarizes the outcomes of the games representing protocols, but it does not show explicitly the implementation risks of such systems. A second approach is to select those Nash Equilibria that fulfill some properties. In [2] equilibria with weakly dominated strategies are excluded, defining the category of *practical* strategy profiles. Again in [2] a strategy profile is defined as *k-resilient* if there is no coalition with at most k players having an incentive to deviate from the prescribed protocol. Other definitions [2, 3, 49] take probability into account and extend the concept of Nash Equilibrium. In [36], *virtual utility* – alternative to the classical game utility – is introduced to capture the blockchain agreement structure. The analysis of robustness with respect to Byzantine agents can be developed in different ways. On one hand, Byzantine behaviours can be modeled with a Bayesian game, in which being altruistic or Byzantine corresponds to two different types [5]. This method allows making forecasts on the expected outcomes of a game, but it does not provide a comprehensive analysis of the risks. On the other hand, properties of the strategy profile that corresponds to the protocol can be analyzed. Authors in [2] introduce the concept of *t-immunity* (generalised in [18]), i.e., no player gets a lower outcome if there are at most t Byzantine players that can play any possible strategy. A third approach is to model the problem with a *two-player zero-sum game* [58], in which the designer of a protocol can simulate the response of an attacker to an ideal functionality by defining its utility [10, 26].

Our contribution. This paper presents a game theoretical framework aiming at characterizing blockchain protocols, modeled as games, in terms of robustness i.e., resilience to rational deviations and immunity to Byzantine behaviors. Robustness analysis of blockchain protocols were performed in [2] by adopting the concept of *mechanism* (i.e., a pair game-prescribed strategy). In order to characterize the robustness of a distributed system the authors introduce the notions of (i) k-resilience, (ii) practicality and (iii) t-immunity. More precisely, k-resilience and practicality analyze the robustness with respect to rational agents, while t-immunity deals with Byzantine agents. In this paper we use the concept of mechanism proposed in [2] to model different types of blockchain protocols and we define a set properties to be satisfied in terms of robustness. Since the property of t-immunity is often impossible to be satisfied by practical systems [1], we introduce the concept of *t-weak-immunity.* A mechanism is t-weak-immune if any altruistic player receives no worse payoff than the initial state, no matter how any set of t players deviate from the prescribed protocol. We further extend the framework in [2] by proving some necessary and sufficient conditions for a mechanism to be optimal resilient and t-weak-immune. In order to make the method scalable to any modular protocol, we define a new operator for mechanism composition and prove that it preserves the robustness properties of the individual games. Using our framework we studied the properties of a set of layer-1, layer-2 and cross-chain protocols: Bitcoin [44], Tendermint [40], Lightning Network protocol [51], the side-chain protocol [50] and the very first implementation of a cross-chain swap protocol proposed in [47] and formalized in [32]. Thanks to the analysis of protocol robustness we spotted the weakness of the Lightning Network protocol to Byzantine behaviour and therefore we propose and further analyze an alternative version of the protocol. Our results are reported in Table 1. The paper is structured as follows. Section 3 is devoted to the definition of mechanism, (k,t)-weak-robustness, necessary and sufficient conditions for optimal resilience and weak immunity and composition of mechanisms. We apply in Section 4 the methodology developed in Section 3 to prove the robustness of the protocols presented in [40, 44, 47, 50, 51]. Section 5 concludes the paper.

Table 1: Immunity and resilience properties for Tendermint [40], Bitcoin [44], Lightning Network [51], a side-chain protocol [50] and a cross-chain swap protocol [32, 47] with respect to the number of rational deviating agents (k) and the number of Byzantine deviating agents (t) where n is the total number of players in the game.

Protocol	k-Resilience	t-Immunity	t-Weak Immunity	Results
Tendermint	**Yes, k** $< n/3$	**No**	**Yes, t** $< n/3$	Theorem 5, 6
Bitcoin	**Yes, k** $< 3n/20$	**No**	**No**	Theorem 7, 8
Lightning Network	**Yes, k** $< 3n/20$	**No**	**No**	Theorem 9
Closing module	Yes	No	No	Theorem 12
(Alternative closing module)	(Yes)	(No)	(Yes)	Theorem 12
Other modules	Yes	No	Yes	Theorem 10, 11, 15, 18, 19
Side-chain (Platypus)	**Yes, k** $< n/3$	**No**	**Yes, t** $< n/3$	Theorem 20
Cross-chain Swap	**Yes**	**No**	**Yes**	Theorem 23

2 Games, mechanisms and robustness

2.1 Preliminaries on games

Throughout the text we consider processes in which multiple decision-makers are involved. We introduce game theoretical concepts in order to study the optimal decision-making process. The basic idea of a game is to capture a set of players which act in sequence. Its graphic representation is called *game tree*. Formally, the theoretical concept which models this situation is the *extensive form game* [39].

Definition 1 (extensive form game)**.** An extensive form game with perfect information is a tuple $\Gamma = \langle N, T, P, (A_h)_{h\in V}, (u_i)_{i\in N}\rangle$, where:

- N is the set of players.
- $T = (V, E)$ is a directed rooted tree.
- $Z \subset V$ is the set of terminal nodes.
- $P : V \setminus Z \to N$ is a function assigning to each non-end node a player in N. The function P identifies at which nodes a player acts.
- $A_h = \{(x_h, x_i) \in E\}$ for each node $h \in V \setminus Z$ is the set of edges going from node h to some other nodes and represents the set of actions at node h of the tree T.
- $\Omega_i = \{s_i : V \setminus Z \to A_1 \times A_2 \times \dots A_h \times \cdots \times A_H, h : P(h) = i\}$ is the set of pure strategies of player i. Every pure strategy of player i is a function that assigns an action $a \in A_h$ to every node $h \in V \setminus Z$ in which player i is involved (formally, $h : P(h) = i$).
- $\mathscr{S}_i = \{\sigma_i : \Omega_i \to [0,1], \sum_{s\in\Omega_i} \sigma_i(s) = 1\}$ is the set of mixed strategies of player i. A mixed strategy is a probabily distribution over the set of pure strategies of player i.
- $u_i : Z \to \mathbb{R}$ is the utility function for player $i \in N$.

Fig. 1 represents a game in extensive form Γ with players $N = \{A, B\}$ and non-terminal nodes $V \setminus Z = \{a, b, c\}$. The structure and the notation of a game in extensive form is not practical for the purpose of the analysis. Every game in extensive form can be rewritten in a more compact way, called *normal form* representation [39], as shown in Fig. 2.

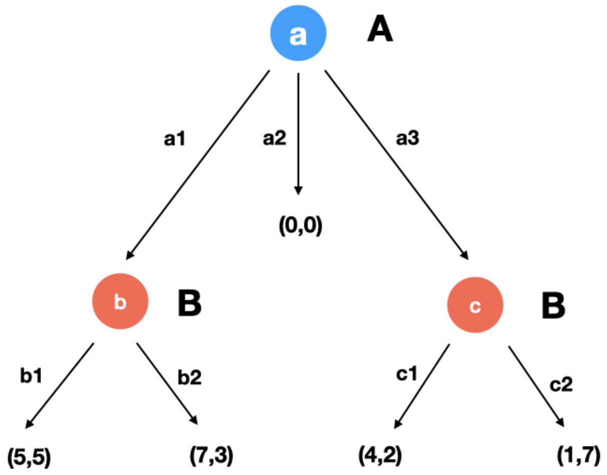
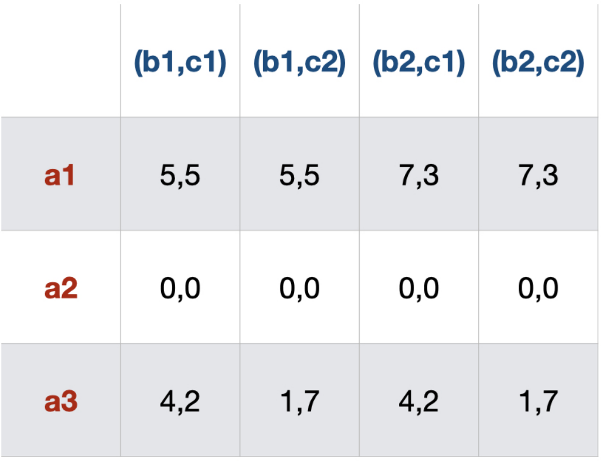


Figure 1: Game Γ in extensive form.

Definition 2 (normal form game)**.** A game in a normal form representation is identified by a tuple $\Gamma = \langle N, \mathscr{S}, u \rangle$, where N is a finite set of n players, $\mathscr{S} = \mathscr{S}_1 \times \mathscr{S}_2 \times \cdots \times \mathscr{S}_n$ where $\mathscr{S}_i$ is the set of strategies of player i and $u : \mathscr{S} \to \mathbb{R}^n$ is the utility function of the players.

Every player i has available a set of strategies $\mathscr{S}_i$. Let us suppose that every player picks a strategy $\sigma_i \in \mathscr{S}_i$; then it is possible to compute the utility for a player i: $u_i(\sigma_1, \sigma_2, \ldots, \sigma_n)$, which is the i-th component of the function u. Since they are rational agents, the goal of the players is to maximize their utility by choosing their strategy. Usually there is no strategy that allows every player to maximize their utility, therefore we have to consider strategy profiles $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$. Each player i chooses a strategy σ_i and the outcome $u(\sigma)$ pleases every player, so that they do not want to change their strategy. We introduce some *solution concepts* of a game, that consists of sets of strategy profiles.



	(b1,c1)	(b1,c2)	(b2,c1)	(b2,c2)
a1	5,5	5,5	7,3	7,3
a2	0,0	0,0	0,0	0,0
a3	4,2	1,7	4,2	1,7

Figure 2: Game Γ in normal form.

Definition 3 (Nash Equilibrium)**.** A strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is a Nash equilibrium if:

$$u_i(\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \geq u_i(\sigma_1, \sigma_2, \ldots, \tau_i, \ldots, \sigma_n)$$

for every player i and for every $\tau_i \in S_i$.

The definition of Nash equilibrium is based on the concept of *best response*, i.e., the strategy σ_i that maximizes the utility of a player i, given the strategies of the other players σ_{-i}. In a Nash equilibrium no player has an incentive to unilaterally change its strategy since utilities do not increase. Nash [45] proves that every game in normal form admits at least one Nash equilibrium. *Nash equilibria* are reasonable solution concepts since they represent a scenario in which nobody is tempted to unilaterally change her own strategy. However, the set of Nash equilibria is not always a singleton, it might happen indeed that there is more than one equilibrium. Here below some properties of Nash equilibria are introduced.

Definition 4 (strong Nash equilibrium [16]). A Nash equilibrium $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_i, \dots, \sigma_n) \in \mathscr{S}$ is said to be strong if and only if for all $C \subseteq N$, all $\tau_C \in \mathscr{S}_C$, $\exists i \in C$ such that $u_i(\sigma_C, \sigma_{-C}) \geq u_i(\tau_C, \sigma_{-C})$.

In [16] the authors prove that the outcome of every strong Nash equilibrium is Pareto efficient i.e., no player can improve her outcome without reducing the outcome of another players. Strong Nash equilibria are easy to be identified, but they do not always exist.

Definition 5 (stable Nash equilibrium [33]). A Nash equilibrium $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_i, \dots, \sigma_n) \in \mathscr{S}$ is said to be stable if it belongs to the set S which is minimal with respect to the following property: for every $\epsilon > 0$ there exists $\delta > 0$ such that any upper-hemicontinuous compact convex valued correspondence pointwise within Hausdorff distance δ of the best response correspondence of Γ has a fixed point within ϵ of S.

The concept of stable equilibria was introduced in [37] in order to exclude less meaningful Nash equilibria i.e., those equilibria that are less resilient against small changes. After [37], several other definitions of stability were introduced. We cite the definition provided in [33], which fulfills some useful properties. One of these states that there always exists a stable Nash equilibrium. Moreover, stable Nash equilibria survive after the iterated deletion of *weakly dominated strategies*, i.e., those strategies $\sigma_i \in \mathscr{S}_i$ that perform as well as or worse than another strategy $\sigma'_i \in \mathscr{S}_i$ no matter which strategy the other players choose (formally, we have that $u_i(\sigma_i, \tau_{-i}) \leq u_i(\sigma'_i, \tau_{-i})$ for all $\tau_{-i} \in \mathscr{S}_{-i}$). In the process of iterated deletion [48] weakly dominated strategies are excluded from the set of strategies available to players and the set of Nash equilibria is recomputed.

3 Games theoretical framework for proving protocols robustness

3.1 Mechanisms and Robustness

In a distributed protocol, agents who run it can either decide to follow the prescribed protocol or not. In case they do not, they deviate from the prescribed protocol by choosing a *byzantine* behaviour. We would like to model these situations and understand whether the players are incentivized to follow the given advice. In [2] the authors introduce a game theoretical framework based on the concept of mechanism and its properties. In the following we recall and extend the framework of [2].

A game is a tuple $\Gamma = \langle N, \mathscr{S}, u \rangle$ in which the set of players N corresponds to the agents involved in a protocol. We map all the possible behaviours of the players and define them as their strategies $\mathscr{S}$. Following the protocol corresponds to one and only strategy $\sigma_i \in \mathscr{S}_i$ for every player i. For the sake of simplicity we assign utility $u_i(s) = 0$ for every $s \in \mathscr{S}$ when the player i is indifferent between the outcome of the strategy profile s and the outcome of the initial state, i.e. the utility given to the players before the game is played. Analogously we assign utility $u_i(s) > 0$ when the outcome of the strategy profile s corresponds to the final state provided by the protocol and $u_i(s) \leq 0$ when the outcome of s is worse than the initial state. The value of the utility corresponds to the marginal utility with respect to the initial state. The choice of the utility function is arbitrary, once the constraints above introduced are fulfilled.

Given the strategy profile $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_i, \dots, \sigma_n) \in \mathscr{S}$ that corresponds to every player i following the protocol by playing strategy σ_i we define the mechanism (Γ, σ).

Definition 6 (mechanism [2])**.** A mechanism is a pair (Γ, σ) in which $\Gamma = \langle N, \mathscr{S}, u \rangle$ is a game and
$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is a strategy profile.

Every player is advised to play strategy $\sigma_i \in \mathscr{S}_i$. The game Γ shows all the possible strategies available to the players.

Players have a very low incentive to play weakly dominated strategies (cf. Definition 5) since they always have available a different strategy that provides no lower outcome in any scenario. A practical mechanism, formally defined below, ensures that these strategies are not included.

Definition 7 (practical mechanism [2])**.** A mechanism (Γ, σ) is practical if σ is a Nash equilibrium of the game Γ after the iterated deletion of weakly dominated strategies.

Evaluating the resilience of a distributed protocol to Byzantine behaviors corresponds to identifying the properties of the mechanism (Γ, σ). Users can decide to choose a Byzantine behaviour for two different reasons. On one hand they can cooperate in order to find a strategy profile that provides a better outcome than the one given by the protocol, i.e. that increases any of their utilities. A mechanism which is optimal resilient, i.e., *practical* (cf. Definition 7) and *strongly resilient* (cf. Definition 8), discourages these behaviours. On the other hand some agents can behave maliciously for any reason and bring other players to unpleasant scenarios. In [2] a mechanism is t-immune to this behavior if it provides not inferior utility in the case when at most t players play a strategy different from the one prescribed by the mechanism. This condition has been already identified as being too strong in practice therefore we introduce the property of *t-weak-immunity* (cf. Definition 10), which means that a player i who chooses the prescribed strategy $\sigma_i \in \mathscr{S}_i$ is never lead to a worse state than the initial one, under the hypothesis that at most t players are byzantine.

In [2] the authors introduce a generalization of Nash equilibrium, k-resilient equilibrium defined formally below. The definition is a generalization of the concept of Nash equilibrium, which can be considered as a 1-resilient equilibrium. Indeed, in a Nash equilibrium no coalition formed by a single player has an incentive to change strategy. In a k-resilient equilibrium there is no coalition of k players that have an incentive to simultaneously change strategy to get a better outcome, i.e. when any of the players identifies a larger utility. Given a coalition of rational players $C \subseteq N$ of size up to $k : 1 \leq k < |N|$, the strategy profile $\sigma \in \mathscr{S}$ and any other of their strategy profiles $\tau_C \in \mathscr{S}_C$ we can define k-resilience as follows.

Definition 8 (k-resilient equilibrium [2])**.** A strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is a k-resilient equilibrium if for all $C \subseteq N$ with $1 \leq |C| \leq k$, all $\tau_C \in \mathscr{S}_C$ and all $i \in C$, we have $u_i(\sigma_C, \sigma_{-C}) \geq u_i(\tau_C, \sigma_{-C})$.

We say that a mechanism (Γ, σ) is *k-resilient* if σ is a k-resilient equilibrium for Γ.

If every strict subset of the players has no incentive to change strategy we say that the strategy profile is *strongly resilient* (formally, if it is k-resilient for all $k \leq n-1$). We say that a mechanism (Γ, σ) is *strongly resilient* if σ is *strongly resilient.*

A mechanism (Γ, σ) is *optimal resilient* if it is practical and strongly resilient.

One of the basic assumption of game theory is that agents are rational. However, in real applications it might happen that agents behave irrationally. There are different reasons for this. Agents might have some limits that do not let them identify and choose rational behaviours. We always work under the assumptions that everything works, but there might be some technical failures that make some actions inaccessible to players. Lastly, the game might be not independent from other games. For instance, some agents might be subject to bribes which entice them to play an irrational strategy. Therefore it is interesting to study strategies that are immune to this type of behaviors. A strategy profile is t-immune if it provides not inferior utility in the case when at most t players play a strategy different from the one prescribed by the mechanism.

Definition 9 (t-immunity [2])**.** A strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is t-immune if for all $T \subseteq N$ with $|T| \leq t$, all $\tau_T \in \mathscr{S}_T$ and all $i \in N \setminus T$, we have $u_i(\sigma_{-T}, \tau_T) \geq u_i(\sigma)$. A mechanism (Γ, σ) is *t-immune* if σ is t-immune in the game Γ.

The concept of k-resilience denotes the tendency of a set of k players to cooperate to move to a equilibrium different from the one prescribed. On the other hand, the concept of t-immunity evaluates the risk of a set of t players to defect and play a different strategy that can damage the other players. The two concepts are complementary. In [2] the authors introduced the notion of (k,t)-*robust* mechanism. A mechanism is (k,t)-*robust* if it is k-resilient and t-immune.

The property of t-immunity (cf. Definition 9) is too strong and difficult to be verified in practice because it requires that the protocol provided the *best outcome* no matter which strategy a set of t players choose. In [18] the author generalizes it with the definition of (t,r)-immunity, i.e., that players receive at least $u(\sigma) - r$ no matter what the other players do. For our purposes we need a more specific definition, that is valid for all players and that is related to a threshold, that we fix equal to zero. Since zero is the utility provided to players in their initial state, the property of immunity corresponds to guaranteeing at least the value of the initial state to every player. Given a coalition of Byzantine players $T \subseteq N$ of size up to $t : 1 \leq t < |N|$, their strategy profile $\tau_T \in \mathscr{S}_T$ and the set of strategies σ_{-T} of altruistic players $i \in N \setminus T$ we can define t-weak-immunity as follows.

Definition 10 (t-weak-immunity)**.** A strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is t-weak-immune if for all $T \subseteq N$ with $|T| \leq t$, all $\tau_T \in \mathscr{S}_T$ and all $i \in N \setminus T$, we have $u_i(\sigma_{-T}, \tau_T) \geq 0$. A mechanism (Γ, σ) is *t-weak-immune* if σ is t-weak-immune in the game Γ.

A player that joins a mechanism that is t-weak-immune knows that she does not suffer any loss (i.e., outcome with negative utility) if there are at most t Byzantine players in the game. Under the assumption that a protocol provides positive outcomes, a t-immune strategy is always t-weak-immune. As the denomination might suggest, this new property is weaker. Formally, it is possible to consider it as one of its generalizations. Indeed, if we consider the equivalent game $\Gamma' = \langle N, \mathscr{S}, u' \rangle$ with $u' = u - u(\sigma)$, the definition of t-immunity and t-weak-immunity are identical. We define as *weak immune* a strategy profile that is t-weak-immune for every t.

In Section 3.2 we provide necessary and sufficient conditions to prove that a mechanism satisfies the property of optimal resilience and t-weak-immunity.

Finally, we have to take into account that players run complex protocols composed of a set of modules. We introduce in Section 3.3 the operator *composition* of games (cf. Definition 11), i.e., the game that corresponds to different games run at the same time by the same players. We prove that the properties above introduced are invariant with respect to this operator, i.e., if two protocols are independent one from another they preserve their properties when played at the same time.

3.2 Necessary and sufficient conditions for optimal resilience and weak immunity

In the following we study the necessary and sufficient conditions for mechanisms to be *optimal resilient* and *weak immune.*

According to [2] if every strict subset of players has no incentive to change their strategy we say that the strategy profile is *strongly resilient.* (Γ, σ) is a *strongly resilient mechanism* if σ is strongly resilient. A mechanism (Γ, σ) is *optimal resilient* if it is practical and strongly resilient. The concepts of k-resilience and practicality are strictly connected with the properties of Nash equilibria, which have been fully studied (see for example [16, 22, 33, 37]). Therefore, connecting these two notions, through necessary and sufficient conditions, allow us to directly exploit the properties of Nash equilibria, such as *strength* [16] and *stability* [33, 37].

Proposition 1 (strong resilience)**.** If a mechanism (Γ, σ) is strongly resilient, then $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n)$ is a strong equilibrium of Γ.

Proof. If (Γ, σ) is a strongly resilient mechanism, then for all $C \subset N$ and for all i such that $u_i(\sigma_C, \sigma_{-C}) \geq u_i(\tau_C, \sigma_{-C})$ (cf. Definition 8). Therefore, for all $C \subset N$ there always exists i such that $u_i(\sigma_C, \sigma_{-C}) \geq u_i(\tau_C, \sigma_{-C})$, which corresponds to the Definition 4 of strong equilibrium. □

Strong Nash equilibria are easy to be identified, but they are very rare; indeed, they do not always exist [16]. Therefore, the property of strongly resiliency is even more rare. We thus take into account a different concept of solution, that of stable Nash equilibrium, which tries to identify those Nash equilibria that are more likely to be played. According to definition provided in [33], stable equilibria fulfill different properties, among which they survive the iterated deletion of weakly dominated strategies. The concept of stable equilibria, which is well studied in literature [33, 37] extends the concept of practical mechanism.

Proposition 2 (practicality)**.** If $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n)$ is a stable equilibrium of Γ, then the mechanism (Γ, σ) is practical.

Proof. Stable equilibria survive after the iterated deletion of weakly dominated strategies, therefore the mechanism is practical. □

In [33] the authors proves that there always exists at least one stable Nash equilibrium, that leads us to the following corollary.

Corollary 1. For any game Γ there is always at least one $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ such that the mechanism (Γ, σ) is practical.

Indeed, since for every game $\Gamma = \langle N, \mathscr{S}, u \rangle$ there always exists a stable equilibrium $\sigma \in \mathscr{S}$, from Proposition 2 we have that (Γ, σ) is practical.

We now know prove that the properties of strongly resiliency and practicality are independent, and therefore both of them have to be studied. This result comes from the fact that also strength and stability are independent [37].

Proposition 3. The properties of strongly resiliency and practicality are independent.

Proof. In order to prove the independence we have to identify 4 examples of mechanism with the following properties: (i) strongly resilient and practical, (ii) strongly resilient and not practical, (iii) not strongly resilient and practical, (iv) not strongly resilient and not practical:

1. We define the mechanism (Γ, σ) such that for all i we have that $u_i(\sigma) = 1$ and $u_i(\tau) = 0$ for all $\tau \neq \sigma$. The mechanism is strongly resilient. The strategy profile σ is the only Nash equilibrium. Since there always exists a stable Nash equilibrium, σ is stable and thanks to Proposition 2 we have that (Γ, σ) is practical.

2. Let us consider the mechanism (Γ, σ), in which Γ has two players, for all i and for all $\tau \in \mathscr{S}$ we have that $u_i(\tau) = 1$, but for $\overline{\tau} = (\sigma_1, \overline{\tau}_2)$ with $\overline{\tau}_2 \neq \sigma_2$ which provides utility $u_i(\overline{\tau}) = (0, 1)$. The mechanism (Γ, σ) is strongly resilient, because $u_i(\sigma) \geq u_i(\tau)$ for all i and all $\tau \in \mathscr{S}$. However, it is not practical, as player 1 would not consider $\sigma_1 \in \mathscr{S}_1$, but a different strategy that always provides utility equal to 1.

3. Since strength and stability are independent [37], there always exists a game Γ in which an equilibrium σ is stable, but not strong. The mechanism (Γ, σ) is not strongly resilient (thanks to Proposition 1 and it is practical, since it is stable (Proposition 2).

4. It is enough to define a mechanism (Γ, σ) such that σ is not a Nash equilibrium of Γ.

□

The following proposition provides a necessary and sufficient condition to determine if a mechanism is weak immune.

Proposition 4 (weak immunity)**.** A strategy profile $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_i, \ldots, \sigma_n) \in \mathscr{S}$ is weak immune if and only if for all $i \in N$ in the game $\Gamma_i = \langle N', \mathscr{S}', u' \rangle$ with $N' = \{i, j\}$, $\mathscr{S}'_i = \mathscr{S}_i$, $\mathscr{S}'_j = \mathscr{S}_1 \times \mathscr{S}_2 \times \cdots \times \mathscr{S}_{i-1} \times \mathscr{S}_{i+1} \times \cdots \times \mathscr{S}_n$, $u'_i = u_i$ and $u'_j = -u_i$ the best response $\tau'_j \in S'_j$ to u'_i gives outcome $u'_i(\sigma_i, \tau'_j) \geq 0$.

Proof. Let us prove the *if* part. Since τ'_j is a best response to σ_i, by definition $u'_j(\sigma_i, \tau'_j) \geq u'_j(\sigma_i, \tau')$ for all $\tau' \in \mathscr{S}_j$. Therefore $u'_i(\sigma_i, \tau'_j) \leq u'_i(\sigma_i, \tau')$ and so for all $\tau' \in \mathscr{S}_j$ we have that $u'_i(\sigma_i, \tau') \geq 0$. By construction for every $\tau_{-i} \in \mathscr{S}_{-i}$ there is one and only one $\tau' \in \mathscr{S}'_j$, which gives $u_i(\sigma_i, \tau_{-i}) = u'_i(\sigma_i, \tau'_j)$. Hence we have that $u_i(\sigma_i, \tau_{-i}) \geq 0$ for all $\tau_{-i} \in \mathscr{S}_{-i}$. The proof for the *only if* part is analogous, since we can find a one-to-one correspondence among strategies in $\mathscr{S}$ and $\mathscr{S}'$. □

The principle is to fix one player $i \in N$ at a time and consider all the other players as a unique adversarial player j that sets her strategy in order to reduce the utility of player i. The game Γ_i in which player i faces an adversarial player j belongs to a specific class of games, called *two-player zero-sum games* [58], whose Nash equilibria are always in the form $(v, -v)$ with $v \in \mathbb{R}$. The term v is called *value of the game* and corresponds to the minimum value that player i is able to achieve. Proposition 4 states that a strategy profile is weak immune if and only if the *best response* (i.e., the strategy producing the most favorable outcome) for the adversarial player j assigns to player i a positive outcome $v \geq 0$. This condition allows us to check the weak immunity property by looking at only N outcomes from N games, which is more efficient than considering all the possible outcomes of the game Γ. We see in Section 4.5 how this condition allows us to verify the weak immunity of a mechanism.

3.3 Composition of Games and Mechanisms

Blockchains systems are complex protocols designed in a modular way. In order to study the robustness of such complex protocols we analyze the robustness of the individual modules and infer the properties of the system by composition.

We introduce therefore the notion of *composition of games*. Given two different games A and B, the game $A \odot B$ corresponds to players picking a strategy from each game and receiving as utility the sum of the utilities of the two games. The games are intended to be played separately and independently.

Definition 11. Given $A = \langle N, \mathscr{S}_A, u_A \rangle$ and $B = \langle N, \mathscr{S}_B, u_B \rangle$ two games in normal form with the same set of players N, two different sets of strategies $\mathscr{S}_A = \{\mathscr{S}_{Ai} : i \in N\}$ and $\mathscr{S}_B = \{\mathscr{S}_{Bi} : i \in N\}$ and two different utility functions: $u_A : \mathscr{S}_A \to \mathbb{R}^N$ and $u_B : \mathscr{S}_B \to \mathbb{R}^N$ then, it is possible to define a new game $C = A \odot B$, called composition of A and B, which is characterized as follows. $C = \langle N, \mathscr{S}_C, u_C \rangle$, where:

- N is the set of the players,
- $\mathscr{S}_C := \{(s_{Ai}, s_{Bi}), s_{Ai} \in \mathscr{S}_{Ai}, s_{Bi} \in \mathscr{S}_{Bi}, \forall i \in N\}$ is the set of strategies,
- $u_C(\{(\sigma_{Ai}, \sigma_{Bi})\}) := u_A(\{\sigma_{Ai}\}) + u_B(\{\sigma_{Bi}\})$ is the utility function.

In the context of non-cooperative games linear transformations of utility functions ($u'_i = a \cdot u_i + b$ with $a \in \mathbb{R}^+$ and $b \in R$) are considered invariant transformations since they preserve the main properties of the game [31]. Therefore, defining the utility function of the composition of games as the sum of the utility functions is equivalent to defining it for any linear combination. It is possible to extend the definition of composition of games to pairs of games in which different sets of players are involved. Indeed, for instance if a player i is involved in game A but not in game B, it is possible to extend game $B = \langle N, \mathscr{S}_B, u_B \rangle$ to $B = \langle N', \mathscr{S}'_B, u'_B \rangle$ in which player i is added ($N' = N \cup \{i\}$) and she is assigned a "null" strategy ($\mathscr{S}'_B = \mathscr{S}_B \times \{\sigma_\emptyset\}$) not influencing the utilities of the outcomes. Formally, for all $s \in \mathscr{S}_B$ and for all $j \in N' \setminus \{i\}$, $u'_j(s, \sigma_\emptyset) = u_j(s)$, while for $i \in N'$ we have that $u_i(s, \sigma_\emptyset) = 0$. Intuitively it is possible to extend the definition of composition of games to more than two games. In Section 4.8 we use the notation $A \odot B \odot C$ to represent either game $A \odot (B \odot C)$ or $(A \odot B) \odot C$. We do not prove the associative property of this operator, but it is intuitive that the two games are the same, except for a different strategy labelling.

The following theorems allow us to model the building blocks of complex protocols, study the properties of the subsequent mechanisms and finally, through the composition of mechanisms, deduce the properties of the composed protocol.

Theorem 1. *Let $A = \langle N, \mathscr{S}_A, u_A \rangle$ and $B = \langle N, \mathscr{S}_B, u_B \rangle$ be two games in normal form representation. Then, $\{(\sigma_{Ai}, \sigma_{Bi})\}$ is a Nash equilibrium for $A \odot B$ if and only if $\{\sigma_{Ai}\}$ and $\{\sigma_{Bi}\}$ are Nash equilibria respectively for A and B.*

Proof. Let us prove the *if* part. If $\{\sigma_{Ai}\}$ and $\{\sigma_{Bi}\}$ are Nash equilibria for A and B, then $\forall j$ and for any other pair of strategies for player j, σ'_{Aj} and σ'_{Bj} we have that:

$$u_A(\{\sigma_{Aj}, \sigma_{A-j}\}) \geq u_A(\{\sigma'_{Aj}, \sigma_{A-j}\}) \text{ and } u_B(\{\sigma_{Bj}, \sigma_{B-j}\}) \geq u_B(\{\sigma'_{Bj}, \sigma_{B-j}\})$$

where $-j := \{i \in N : i \neq j\}$. Hence, for any other $\{(\sigma'_{Aj}, \sigma'_{Bj}), (\sigma_{A-j}, \sigma_{B-j})\}$ it is possible to deduce that:

$$u_{A\odot B}(\{(\sigma_{Ai}, \sigma_{Bi})\}) := u_A(\{\sigma_{Ai}\}) + u_B(\{\sigma_{Bi}\}) \geq$$
$$\geq u_A(\{\sigma'_{Aj}, \sigma_{A-j}\}) + u_B(\{\sigma'_{Bj}, \sigma_{B-j}\}) =: u_{A\odot B}(\{(\sigma'_{Aj}, \sigma'_{Bj}), (\sigma_{A-j}, \sigma_{B-j})\})$$

that is, $\{(\sigma_{Ai}, \sigma_{Bi})\}$ is a Nash equilibrium for $A \odot B$.
Let us prove the *only if* part by contradiction, i.e., $\exists\{(\sigma_{Ai}, \sigma_{Bi})\}$ that is a Nash equilibrium for $A \odot B$ but at least one among $\{\sigma_{Ai}\}$ and $\{\sigma_{Bi}\}$ is not a Nash equilibrium for A or B. Let us suppose that $\{\sigma_{Ai}\}$ is not a Nash equilibrium for A: $\exists j, \exists \sigma'_A : u_A(\{\sigma_{Aj}, \sigma_{A-j}\}) < u_A(\{\sigma'_{Aj}, \sigma_{A-j}\})$ then,

$$u_{A\odot B}(\{(\sigma_{Ai}, \sigma_{Bi})\}) := u_A(\{\sigma_{Ai}\}) + u_B(\{\sigma_{Bi}\}) <$$
$$< u_A(\{\sigma'_{Aj}, \sigma_{A-j}\}) + u_B(\{\sigma_{Bj}, \sigma_{B-j}\}) =: u_{A\odot B}(\{(\sigma'_{Aj}, \sigma_{Bj}), (\sigma_{A-j}, \sigma_{B-j})\}$$

which contradicts the hypothesis that $\{(\sigma_{Ai}, \sigma_{Bi})\}$ is a Nash equilibrium for $A \odot B$. □

The Nash equilibria can be identified by selecting equilibria within the single games. It is not possible to create other Nash equilibria nor to lose them in the process of composition of the games.

Concerning robustness properties for composition of games, we can state the following results on resiliency and weak immunity for two composed games. The results can be generalized for the composition of multiple games.

Theorem 2. *Let $A = \langle N, \mathscr{S}_A, u_A\rangle$ and $B = \langle N, \mathscr{S}_B, u_B\rangle$ be two games, (A, σ_A) and (B, σ_B) two practical mechanisms. Then, $(A \odot B, \{\sigma_{Ai}, \sigma_{Bi}\})$ is a practical mechanism.*

Proof. Thanks to Theorem 1 we have that $\{\sigma_{Ai}, \sigma_{Bi}\}$ is a Nash equilibrium for $A \odot B$. It is sufficient to prove that it survives the iterated deletion of weakly dominated strategy. Indeed, every strategy in the form (τ^*_{Ai}, τ_{Bi}) or (τ_{Ai}, τ^*_{Bi}), where τ^*_A is weakly dominated in A and τ^*_B is weakly dominated in B for some player i, is weakly dominated by another Nash equilibrium in $A \odot B$ for the very same player i. The strategy profile $\{\sigma_{Ai}, \sigma_{Bi}\}$ survives the iterated deletion of these weakly dominated strategies. It is now sufficient to prove that there is no other weakly dominated strategy. By contradiction we assume that there is a player i such that there exists $(\bar{\sigma}_{Ai}, \bar{\sigma}_{Bi}) \in \mathscr{S}_{A\odot B}$ that weakly dominates $(\sigma_{Ai}, \sigma_{Bi})$. Therefore, considering the utility u for the player i, for every $(\tau_{A,-i}, \tau_{B,-i}) \in \mathscr{S}_{A\odot B,-i}$ we have that:

$$u_{A\odot B}(\{(\bar{\sigma}_{Ai}, \bar{\sigma}_{Bi}), (\tau_{A,-i}, \tau_{B,-i})\}) \geq u_{A\odot B}(\{(\sigma_{Ai}, \sigma_{Bi}), (\tau_{A,-i}, \tau_{B,-i})\}).$$

Since σ_{Ai} is not dominated by $\bar{\sigma}_{Ai}$ in the game A, there exists $\bar{\tau}_{A,-i} \in \mathscr{S}_{A,-i}$ such that $u_A(\bar{\sigma}_{Ai}, \bar{\tau}_{A,-i}) < u_A(\sigma_{Ai}, \bar{\tau}_{A,-i})$.
Analogously there exists $\bar{\tau}_{B,-i} \in \mathscr{S}_{B,-i}$ such that $u_B(\bar{\sigma}_{Bi}, \bar{\tau}_{B,-i}) < u_B(\sigma_{Bi}, \bar{\tau}_{B,-i})$.
Therefore we have that:

$$u_{A\odot B}(\{(\bar{\sigma}_{Ai}, \bar{\sigma}_{Bi}), (\bar{\tau}_{A,-i}, \bar{\tau}_{B,-i})\}) < u_{A\odot B}(\{(\sigma_{Ai}, \sigma_{Bi}), (\bar{\tau}_{A,-i}, \bar{\tau}_{B,-i})\}),$$

which contradicts the assumption. □

Theorem 2 formalizes the intuition that if two mechanisms are practical then, playing both selected strategy profiles is still a practical mechanism. Following propositions prove the resilience and immunity of the games composition.

Theorem 3. *Let $A = \langle N, \mathscr{S}_A, u_A\rangle$ and $B = \langle N, \mathscr{S}_B, u_B\rangle$ be two games, (A, σ_A) and (B, σ_B) two mechanisms respectively k-resilient and k'-resilient. Then, $(A \odot B, \{\sigma_{Ai}, \sigma_{Bi}\})$ is a $\min(k, k')$-resilient mechanism.*

Proof. We know that for all $C \subseteq N$ with $1 \leq |C| \leq k$, all $\tau_{A,C} \in \mathscr{S}_{A,C}$ and all $i \in C$, we have $u_{Ai}(\sigma_{A,C}, \sigma_{A,-C}) \geq u_i(\tau_{A,C}, \sigma_{A,-C})$. Analogously, for all $C' \subseteq N$ with $1 \leq |C'| \leq k'$, all $\tau_{B,C'} \in \mathscr{S}_{B,C'}$ and all $i \in C'$, we have $u_{Bi}(\sigma_{B,C'}, \sigma_{B,-C'}) \geq u_i(\tau_{B,C'}, \sigma_{B,-C'})$. Hence, we have that for all $S \subseteq N$ with $1 \leq |S| \leq \min(k, k')$, all $(\tau_{A,S}, \tau_{B,S}) \in \mathscr{S}_{A,S} \times \mathscr{S}_{B,S}$ and all $i \in S$:

$$u_{Ai}(\sigma_{A,S}, \sigma_{A,-S}) + u_{Bi}(\sigma_{B,S}, \sigma_{B,-S}) \geq u_i(\tau_{A,S}, \sigma_{A,-S}) + u_i(\tau_{B,S}, \sigma_{B,-S}).$$

We recall that $\mathscr{S}_{A\odot B,S} = \mathscr{S}_{A,S} \times \mathscr{S}_{B,S}$, thus for all $S \subseteq N$ with $1 \leq |S| \leq \min(k,k')$, all $(\tau_{A,S}, \tau_{B,S}) \in \mathscr{S}_{A\odot B,S}$ and all $i \in S$:

$$u_{A\odot B,i}(\{\sigma_{A,S}, \sigma_{B,S}\}, \{\sigma_{A,-S}, \sigma_{B,-S}\}) \geq u_{A\odot B,i}(\{\tau_{A,S}, \tau_{B,S}\}, \{\sigma_{A,-S}, \sigma_{B,-S}\}).$$

□

If a mechanism is k-resilient, then the protocol is followed by every player whenever at most k rational players are allowed. If there is more than one mechanism, the threshold on the maximum number of rational players allowed is the minimum among the rational player numbers k, k' in the individual mechanisms.

Theorem 4. *Let $A = \langle N, \mathscr{S}_A, u_A\rangle$ and $B = \langle N, \mathscr{S}_B, u_B\rangle$ be two games, (A, σ_A) and (B, σ_B) two mechanisms respectively t-weak-immune and t'-weak-immune. Then, $(A \odot B, \{\sigma_{Ai}, \sigma_{Bi}\})$ is a $\min(t,t')$-weak-immune mechanism.*

Proof. In game A, for all $T \subseteq N$ with $|T| \leq t$, all $\tau_{A,T} \in \mathscr{S}_{A,T}$ and all $i \in N \setminus T$, we have $u_{Ai}(\sigma_{A,-T}, \tau_{A,T}) \geq 0$. In game B, for all $T \subseteq N$ with $|T| \leq t'$, all $\tau_{B,T} \in \mathscr{S}_{B,T}$ and all $i \in N \setminus T$, we have $u_{Bi}(\sigma_{B,-T}, \tau_{B,T}) \geq 0$. Therefore we have that for all $T \subseteq N$ with $1 \leq |T| \leq \min(t,t')$, all $(\tau_{A,T}, \tau_{B,T}) \in \mathscr{S}_{A,T} \times \mathscr{S}_{B,T}$ and all $i \in N \setminus T$:

$$u_{A\odot B,i}(\{\sigma_{A,T}, \sigma_{B,T}\}, \{\tau_{A,-T}, \tau_{B,-T}\}) = u_{Ai}(\sigma_{A,T}, \tau_{A,-T}) + u_{Bi}(\sigma_{B,S}, \tau_{B,-S}) \geq 0$$

□

If a player combines two mechanisms which are weak immune for respectively at most t and t' Byzantine players, then it means that she is considering a mechanism which can provide non-negative outcomes if there are at most a number of Byzantine users equal to $\min(t,t)'$.

The following corollaries generalize the results reported in Theorem 3 and Theorem 4.

Corollary 2. Let $A_1, A_2, \ldots A_n$ with $n \in N$ be games and let $(A_1, \sigma_{A1}), (A_2, \sigma_{A2}), \ldots (A_n, \sigma_{An})$ be the corresponding mechanisms respectively $k_1, k_2, \ldots k_n$-resilient. Then, $(A_1 \odot A_2 \odot \cdots \odot A_n, \{\sigma_{A_1}, \sigma_{A_2}, \ldots, \sigma_{A_n}\})$ is a $\min(k_1, k_2, \ldots, k_n)$-resilient mechanism.

Corollary 3. Let $A_1, A_2, \ldots A_n$ with $n \in N$ be games and let $(A_1, \sigma_{A1}), (A_2, \sigma_{A2}), \ldots (A_n, \sigma_{An})$ be the corresponding mechanisms respectively $t_1, t_2, \ldots t_n$-weak-immune. Then, $(A_1 \odot A_2 \odot \cdots \odot A_n, \{\sigma_{A_1}, \sigma_{A_2}, \ldots, \sigma_{A_n}\})$ is a $\min(t_1, t_2, \ldots, t_n)$-weak-immune mechanism.

4 Applications

In this section we prove the effectiveness of our framework by analyzing the robustness of different protocols from blockchains systems. In Section 4.1 and 4.2 we analyse Tendermint and Bitcoin. Section 4.3 and 4.9 addresses layer-2 protocols; in the first one we analyze Lightning Network [51], a protocol on top of the Bitcoin blockchain while the second section presents analysis on the side-chain protocol Platypus [50]. In Section 4.10 we analyze a cross-chain swap protocol [47], which allows two users to exchange cryptoassets living in two different blockchains. The names of the variables in the following sections are consistent with the notation used in the papers where protocols are introduced.

4.1 Tendermint

Tendermint's consensus protocol (i.e., Tendermint-core [6, 40]) is split into three parts: the Pre-Propose round, the Propose round and the Vote round. During the Pre-Propose round, the proposer presents a block, to the other participants. During the Propose round, each participant chooses whether to accept or not the block and broadcasts her decision. If the votes for the proposal exceed a predetermined threshold ν then participants start the Vote phase. If the block receives more than ν votes, it is validated. Tendermint's consensus algorithm sets $\nu = n - f = \frac{2}{3}n$; the threshold representing the number of non-faulty actors (as n denotes the total number of nodes and f the total number of faulty nodes) is set to $\frac{2}{3}$ of the network participants.

The set of actions available to participants is described in Figure 3.

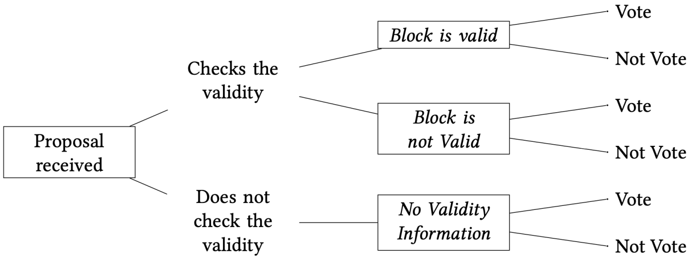


Figure 3: Strategies available to participants [6].

Definition 12. The *Tendermint game* is a mechanism $(\Gamma^{tc}, \sigma^{tc})$ such that the game Γ^{tc} represents the decision-making problem and the strategy σ^{tc} is the prescribed consensus protocol. Once a proposal v is received, N players choose either to check or not to check the validity of the value, then they can choose either to Vote or Not to Vote for it. At the very first stage of the game (stage a) a player can choose either to check (C) the validity or not check (NC). If she checks it, she can choose to Vote or Not Vote for it, in case value v is valid (stage b) or not (stage c). If she does not check it (stage d), she can choose to Vote (V) or Not Vote (NV) for it. Every strategy τ is represented by a vector (a, b, c, d) in which $a \in \{C, NC\}$, $b, c, d \in \{V, NV\}$. The utility for player i is $u_i(\tau) = 1$ if a valid block is approved or a non-valid block is not approved, $u_i(\tau) = 0$ if a valid block is not approved and $u_i(\tau) < 0$ if a non-valid block is approved.

The strategy prescribed by Tendermint's consensus protocol is $\sigma^{tc} = (C, V, NV, NV)$ i.e., to check for the validity of the proposal and then if the block is valid to vote for it, otherwise not vote for it. If the number of rational or byzantine players allowed is $f < \frac{1}{3}n$, the other players have the necessary threshold to validate a block. Indeed, they can veto any validation of blocks proposed by malicious nodes. The mechanism $(\Gamma^{tc}, \sigma^{tc})$ is thus not f-weak-immune for any $f \geq \frac{1}{3}n$ and we can state the following results.

Theorem 5. *The mechanism* $(\Gamma^{tc}, \sigma^{tc})$ *is* (f, f)*-robust for any* $f < \frac{1}{3}n$.

Proof. First, let's consider the case in which the proposer puts forward a non-valid block. If $f < \frac{1}{3}n$ is the number of players who deviate, then at most $\frac{1}{3}n$ will vote for the non-valid block, which is

less then the threshold $\nu = \frac{2}{3}n$ asked by the consensus algorithm to validate the block.
Let's thus consider the case of the proposer putting forward a valid block. The $n-f$ altruistic player will vote in favour of validating the block. Since $n - f \geq \frac{2/3}{n} + 1$, the threshold ν is overcome. □

If there are at least $f \geq \frac{1}{3}n$ byzantine players, it is possible to this set of players to veto any validation of blocks. The mechanism $(\Gamma^{tc}, \sigma^{tc})$ is thus not f-weak-immune for any $f \geq \frac{1}{3}n$. From now on, we exclude the case that no blocks are validated.

Theorem 6. *The mechanism* $(\Gamma^{tc}, \sigma^{tc})$ *is not f-weak-immune for any* $f \geq \frac{1}{3}n + 2$.

Proof. It is enough to prove that if there are $f = \frac{1}{3}n + 2$ byzantine players, a non-valid block is approved. Let us suppose that the players are split in 3 sets: altruistic players are divided in two set A and A' of dimension $\frac{1}{3}n - 1$, while byzantine players are part of the third set B. Let us suppose that the proposer is a byzantine player. She sends two different incompatible values v and v' to the players respectively in A and A'. Then, during the Propose and Vote phase, all the players in B broadcast to player A and A' respectively their vote in favour of v and v'. Both players in A and A' are satisfied, as the threshold of $\frac{n}{3} + 1$ votes is met, so they both broadcast the values v and v' which are however incompatible. □

We considered the case for a generic n, which cannot give results about weak immunity for values $\frac{n}{3}$ and $\frac{1}{3}n + 1$. Tendermint [40] considers only the case of a specific $n = 3f + 1$ number of players, with threshold $\nu = 2f+1$. In this specific case, Theorem 5 consists in setting t, the number of byzantine players, the following condition for weak immunity: $t < \frac{n}{3} < f + 1$. On the other hand, we can state from Theorem 6 that we do not have weak immunity for $t \geq \frac{1}{3}n + 2 > f + 2$. With a similar argument to the one proposed in Theorem 6 it is possible to prove that for $t = f+1$ or $t = f + 2$ Tendermint's protocol does not fulfill weak immunity.

4.2 Bitcoin

Bitcoin is a permissionless blockchain based on a Proof-of-Work mechanism [44] where every user has a chance to publish a new block in the distributed ledger. The user probability to publish/mine a new block is proportional to her computational power α. Bitcoin's protocol [28] requires that once a block is mined, it should be broadcast to every other user. In case two or more blocks are mined at the same moment, the players split equally their effort to mine from any of the blocks (i.e., a *fork* is generated). Hence, published blocks are not automatically validated; they are considered as valid when belonging to the *longest chain* i.e., the longest branch of the ledger.

As for Tendermint, Bitcoin's protocol can be represented by a mechanism $(\Gamma^{btc}, \sigma^{btc})$. We take into account the worst-case scenario, in which the byzantine users coordinate, thus they are represented by a single player i. The altruistic users act in the same way and can therefore be represented by a second player j. The strategies of the players correspond to choosing (i) where in the chain add a new block and (ii) when to publish the mined blocks. Player j plays only one strategy defined by σ^{btc} i.e., she follows the protocol by mining on the main chain (the longest one) or splitting her effort if there is more than one chain of the same length available. Since the game is stochastic, we group all the states of the game that are equivalent in the same class. We consider two states as equivalent if they have the same configuration (i.e., the difference between the number of mined blocks by the i and j is the same) independently from the precise position in the chain. In the Bitcoin blockchain a best practice is to consider a block as valid if belonging to a chain where at least B (usually, $B = 6$) blocks have been published afterwards, because it is presumably considered impossible to create a longer chain that does not include it. This block is invalidated if a fork is made at the previous block and more than $B + 1$ blocks are published starting from it. In this way, the block does not belong to the longest chain anymore and it is not considered valid.

Definition 13. The *Bitcoin game* is a mechanism $(\Gamma^{btc}, \sigma^{btc})$ such that the game Γ^{btc} represents the decision-making problem and the strategy σ^{btc} is the prescribed protocol. The game Γ^{btc} is characterized by two players i and j, who have respectively mining power α and $1 - \alpha$ and every state of the game can be represented by the state class $\{x_k\}_{k\in\{0,1,\dots,B+1\}}$, where x_k is the number of blocks mined, yet not published, at level k by player i. The block at level $k = 0$ is the only one to be published. The initial state of the game is $\{x_k = 0\}$ $\forall k \in \{0, 1, \dots, B + 1\}$, while the final state of the game is state class with value $x_{B+1} \geq 1$. While player j has only one possible strategy σ^{btc}, while player i can choose which branches to mine from (i.e. at which level k add the block). The utility of the players corresponds to the number of bitcoins they own.

The game theoretical framework let us state the following results on Bitcoin's mechanism robustness. Any subset of players T with $|T| = t$ having mining power $\alpha > 0$ have a small probability, not negligible, to perform a successful attack, by building a longer chain which does not include a block which was already considered valid (Theorem 7).

Proposition 5. The probability for a player with mining power α to find n blocks before any other player can find at most m blocks is

$$P(n, m) = \sum_{i=n}^{n+m-1} \binom{n+m-1}{i} \alpha^i (1-\alpha)^{n+m-1-i}.$$

Proof. It's enough to compute that among $n + m - 1$ blocks at least n are being mined by the player with mining power α.

$$P(n, m) = \sum_{i=n}^{n+m-1} \binom{n+m-1}{i} \alpha^i (1-\alpha)^{n+m-1-i}.$$

□

If $m = 1$ we have that $P(n, 1) = \alpha^n$.

Theorem 7. *The Bitcoin mechanism $(\Gamma^{btc}, \sigma^{btc})$ is not t-weak-immune for any t.*

Proof. Let us suppose that a player with mining power $\alpha > 0$ plays the strategy of block withholding if she has mined more blocks than the main chain. When she reaches more than B blocks than the main chain, she publishes all of them, thus invalidating the others. Due to Proposition 5, at every new block she has approximately probability $\alpha^B > 0$ to perform the attack. When the number of attempts goes to ∞, the probability to perform the attack goes to 1. Thus it is almost sure that any subset of players T of cardinality $|T| = t$ with total mining power α, with $\alpha > 0$, the attack will be successfully performed. □

Theorem 8. *The Bitcoin mechanism* $(\Gamma^{btc}, \sigma^{btc})$ *is k-resilient if k players have at most* $\alpha \leq \frac{3}{20}$ *as total mining power.*

Proof. Let us suppose that i is a rational player and j is an altruistic player, i.e. it follows the protocol. The goal of player i is to maximise the number of bitcoins owned by her. A combination of attacks can make her owning more bitcoins than the ones she would receive by following the protocol. Specifically, she can gain bitcoins by double spending them. She validates a block in which she spends M bitcoins, then she creates a fork before the block which creates a longer chain from. Performing such selfish mining attack [25] makes the player i lose some bitcoins. Indeed, creating forks includes the risk of creating blocks which can eventually not belong to the longest chain; thus, the reward R given by these blocks would get lost. Let us define $N(\alpha)$, the average number of blocks lost in the attack by player i, who has computation power α. The player i chooses to perform the attack if $M > R \cdot N(\alpha)$, i.e. if the double spended bitcoins are greater than the average reward lost in the attack. The values of M and R are parameters, while $N(\alpha)$ depends on the strategy chosen by i. Since i is a rational player, she chooses the optimal strategy, i.e. the strategy that minimises $N(\alpha)$.

We thus have to identify the optimal strategy. The probability for player i to add a new block depends on how the other players are split in mining the other blocks. If there are m forks, there is $\alpha' := \frac{\alpha}{\alpha + \frac{1-\alpha}{m+1}}$ chance for player i to mine the next block and $1 - \alpha' = \frac{\frac{1-\alpha}{m+1}}{\alpha + \frac{1-\alpha}{m+1}}$ chance for player j to mine the next block. Player i has probability α' to add a new block at the level k that she chooses, i.e. to add 1 to the value of any x_k. Player j has probability $1 - \alpha'$ to mine a block, which is added at level $k = 0$. The chain is increased by one level, i.e. the number of forks x_1 created at level $k = 1$ are published. The states are moved by one position, i.e. $x_k \to x_{k-1}$. From every state $s = \{x_k\}_{k \in \{0,1,\ldots,B+1\}}$ player i has to mine on average N_s blocks before getting to the final state, following the optimal strategy, i.e. the strategy that minimizes the number of blocks to be mined. The problem has infinite states, because at any level k player i can create $x_k \in \mathscr{N}$ forks. We thus fix a maximum number of blocks $L \in \mathscr{N}$ that can be mined at the same level (i.e. we set $x_k \leq L$) and consider the equivalent problem with a finite number of states. In order to find the optimal solution we compute the optimal Bellman operator [11], which provides the solution in close form. We find out that even increasing L, for significant (> 0.05) values of α the optimal strategy is to perform a selfish mining attack and create only one fork at every level. Figure 4 shows the average number of blocks that player i has to mine in order to perform an attack.

On average every block contains transactions for $M = 10000$ BTC. Mining a block is worth $R = 6.25$ BTC. Therefore an attack is rationally chosen by player i if $N(\alpha) < \frac{M}{R} = \frac{10000}{6.25} = 1600$. Since $N(0.15) = 2347 > 1600$, we have the proof. □

The Bitcoin's mechanism can be made more resilient by reducing the number of bitcoins exchanged in a block M, the reward of a block R and the number of blocks B needed for validation.

On the long run the majority of users ($\alpha \geq \frac{1}{2}$) produce the longer chain. However, on the short run a minority of users ($\alpha < \frac{1}{2}$) can make a fork the longer chain with positive probability. The following theorem provides the value of this probability.

Theorem 9. *The probability for a byzantine player with computation power* α*, with* $\alpha < \frac{1}{2}$*, to prevent a transaction to be published within* Δ *blocks is:*

$$\Phi_\Delta(\alpha) = \frac{\alpha}{1-\alpha} - \sum_{k=1}^{n-1} (1 - \Phi_{n-k}(\alpha)) \cdot \alpha^k \cdot (1-\alpha)^k \cdot M(k),$$

where $M(k)$ *is a function defined in [34] that maps natural numbers to the sequence* $1, 1, 2, 5, 13, 42 \ldots$.

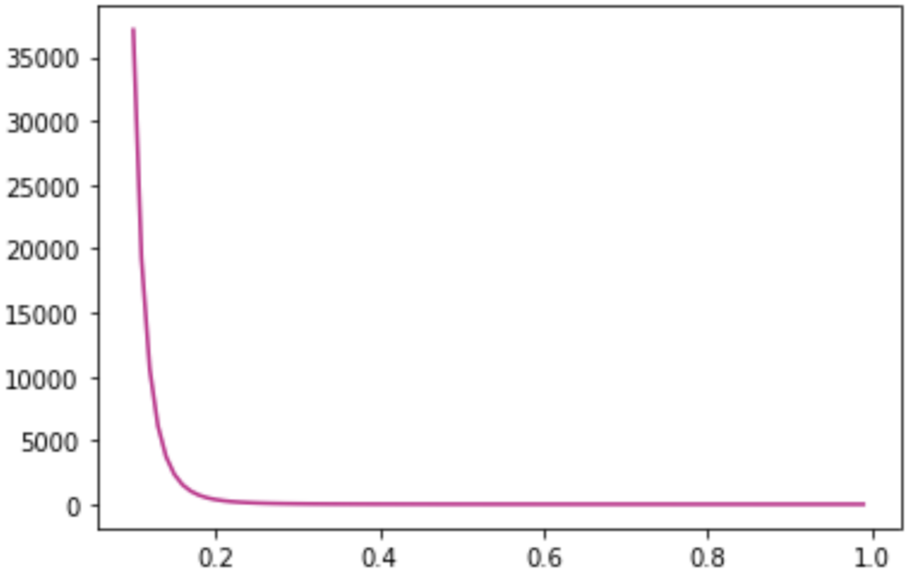


Figure 4: Number of blocks $N(\alpha)$ to be mined by player i with computational power α under the optimal attacking strategy.

Proof. Let us suppose that a transaction is published on the main chain on the next block, unless the byzantine player succeeds in publishing a block which does not include this transaction. First, let us consider the case $n = 1$; the byzantine player succeeds if she publishes the block before any other player. If she fails, her best strategy is to mine blocks on an alternative chain until it gets to be the longest one. If $\alpha \geq \frac{1}{2}$ the byzantine player will almost surely succeed. Otherwise if $\alpha < \frac{1}{2}$, we have that $\Phi_1(\alpha) = \frac{\alpha}{1-\alpha}$. Indeed, we can model the problem with a Markov chain with states $m \in \mathbb{Z} \cap (-\infty, +1]$, in which $+1$ is the only absorbing state, it is possible to move from state n to state $n+1$ with probability α and from state n to state $n-1$ with probability $1-\alpha$. It is a reformulation of the gambler's ruin Markov chain. The state m represents how many blocks the private chain is ahead of the main one. It is enough for the private chain to be one block ahead to succeed.

In case $n > 1$ we can make a similar argument, but excluding the cases in which the state $+1$ is achieved too early. We have that: $\Phi_n(\alpha) = \frac{\alpha}{1-\alpha} - \sum_{k=1}^{n-1} \mathbb{P}(\text{not achieving } +1 \text{ with } n-k \text{ blocks left}) \cdot \mathbb{P}(\text{getting to } +1 \text{ in } 2k \text{ steps})$, which gets us to the formula: $\Phi_n(\alpha) = \frac{\alpha}{1-\alpha} - \sum_{k=1}^{n-1}(1 - \Phi_{n-k}(\alpha)) \cdot \alpha^k \cdot (1-\alpha)^k \cdot M(k)$, where $M(k)$ is a function defined in [34] that maps natural numbers to the sequence $1, 1, 2, 5, 13, 42 \ldots$. □

4.3 Lightning Network

In the Bitcoin blockchain transactions are collected in blocks, validated and published on the distributed ledger [44]. The most known of them, Bitcoin, is based on a Proof-of-Work system that validates blocks of transactions and chains them one to another [44]. Bitcoin faces a problem of scalability, in terms of speed, volume and value of the transactions. A transaction is confirmed only once the block to which it belongs is part of a chain with at least D blocks in front of it (under the convention set by the Bitcoin protocol $D = 6$). On average a new block is validated every T minutes (within Bitcoin, $T = 10$), thus it takes around $T \cdot D = 60$ minutes for a transaction to be confirmed, a value that cannot be reduced. Moreover, the number of transactions in a block is limited. Bitcoin cannot bear a sudden upsurge in volume of transactions. Since not all the requests for transactions can be included in a block, some of them are prioritised. The criterion used to order the transactions is the value of the *fee* that a user pays to the mining pool who validates the block. Therefore performing a lot of transactions on the network can be expensive, since a lot of fees have to be paid.

In order to overcome these issues authors in [51] introduce a layer-2 class of protocols called Lightning Network. The latter allows users to create bidirectional payment *channels* to handle unlimited transactions in a private manner i.e., off-chain without involving the Bitcoin blockchain. Two users A and B open a channel by publishing on the Bitcoin blockchain two transactions towards a fund F. The amounts of the transactions form the initial balance of the channel. In Section 4.4 we analyze the protocolar module to open a channel. The fund F can send or receive cryptoassets via blockchain transactions only if both users sign them.

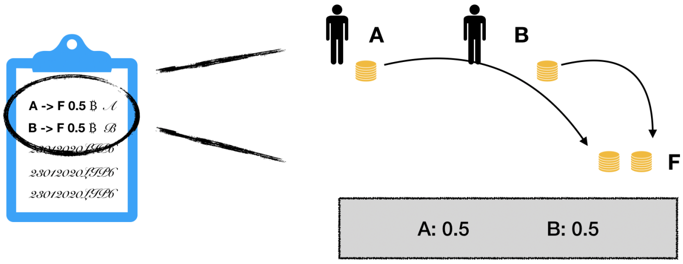


Figure 5: A and B open a channel.

Once the channel is opened, users can exchange by simply privately updating the balance of the channel. The protocol to update the balance is discussed in Section 4.6.

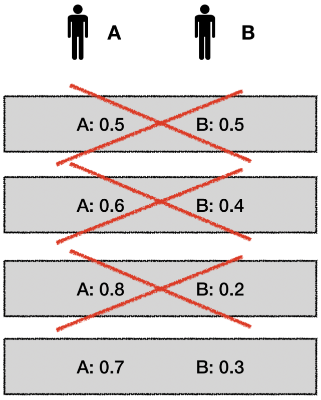


Figure 6: A and B privately update the balance of the channel.

A further construction, called *Hashed Timelock Contract* (HTLC), allows users to create transactions within the channel that can be triggered at will. The structure of the protocol is similar to the one used to update the balance (cf. Section 4.7).

When the users are no more interested in exchanging bitcoins they decide to close the channel. Two transactions are published on the Bitcoin blockchain: one from F to A and another one from F to B. The value of the transactions corresponds to the ones of the latest balance (cf. Section 4.5). The protocol to close the channel is presented in Section 4.5.

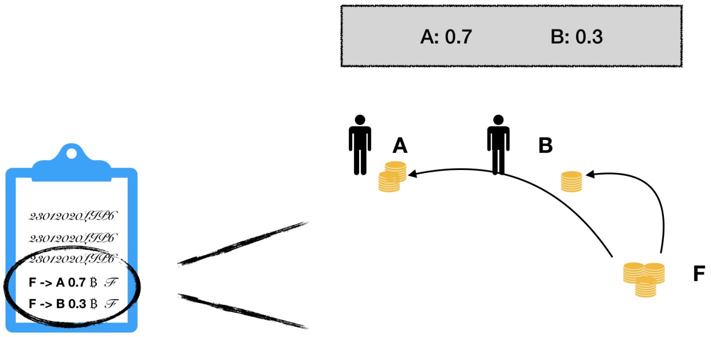


Figure 7: A and B close a channel.

Lightning Network allows transactions also between users who have not opened a common channel (i.e., *routed payment*). Indeed, two users can perform a transaction through a path of open channels, using other users as intermediate nodes. This protocol is analyzed in Section 4.8.

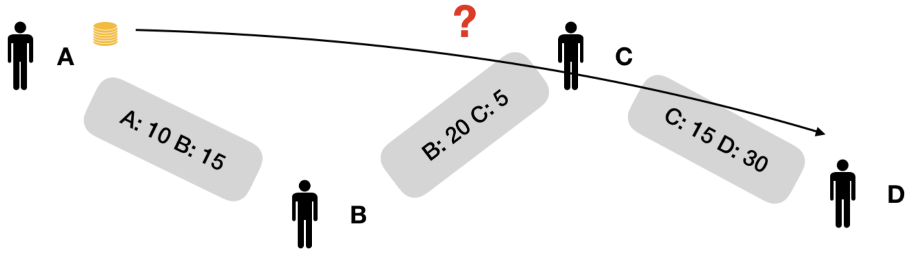


Figure 8: A path of channels between users A and D.

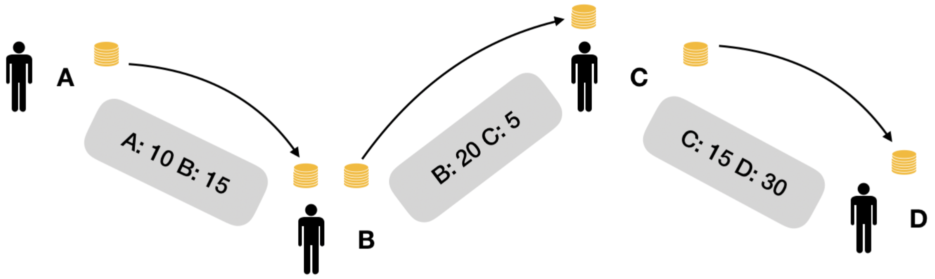


Figure 9: A sends 5 Ƀ to D through nodes B and C.

In the public Bitcoin blockchain every transaction is signed by the sender. In the Lightning Network every operation is identified by a commitment $\mathscr{C}$ which must be signed by two users, let us say A and B. In the following sections we use the following notations: $\mathscr{C}_{..}$ when the commitment is signed by nobody; $\mathscr{C}_{A.}$ when the commitment is signed only by user A; $\mathscr{C}_{.B}$ when the commitment

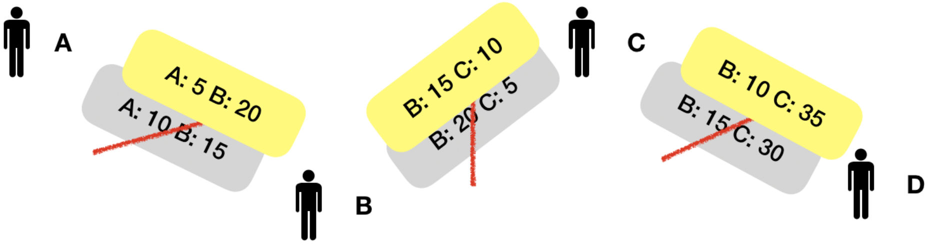


Figure 10: All the balances are updated.

is signed only by user B; $\mathscr{C}_{AB}$ when the commitment is signed by both users, this is the only case in which the commitment $\mathscr{C}$ is valid.

In practice, the channel consists of a user, let us say F. Every transaction from and to F must be signed by both users A and B.

4.4 Opening module

Informally, the protocol asks the users to fund the channel F with two different transactions, respectively valued x_A and x_B, and to create two different commitments that allow them to publish a transaction that makes them close the channel unilaterally. Formally, in order to open a channel the users perform a transaction Tx towards F signed by both of them and they create two different commitments that let them close the channel unilaterally. The protocol involves the following steps (cf. Fig. 11):

1. A creates a transaction *C1b* that allows F to send x_A to A and to send x_B to B. B is able to spend x_B only after that Δ blocks are validated (in [51] $\Delta = 1000$). A signs *C1b* and sends it to B.

2. B creates a transaction *C1a* that allows F to send x_A to A and to send x_B to B. A is able to spend x_A only after that Δ blocks are validated. B signs *C1a* and sends it to B.

3. A creates a transaction Tx that makes A send x_A to F and B send x_B to F. A signs Tx and sends it to B.

4. B signs Tx and publishes it on the Bitcoin blockchain.

If a user decides to close the channel unilaterally, she receives her part of funds after a certain interval of time, while the other user receives it immediately. We formalize the protocol with a game in extensive form Γ^{op} (cf. Definition 14), represented by its game tree (cf. Fig. 12). At every node of the tree (i.e., decision step) the player involved in the protocol has two actions available: either following it by signing the commitment required or not following it. The *initial state* corresponds to having no channel opened, while the final state corresponds to having the channel opened. We assign *null* utility to the initial state and positive utility (by convention fixed to 1) to the final state. If at any step the players do not follow the protocol, they get back to the initial state, with outcome $(0, 0)$. If they do follow at every step, they are able to open the channel, with outcome $(1, 1)$. We denote by $\sigma^{op} = (\{C1b_{A\cdot}, Tx_{A\cdot}\}(\{C1a_{\cdot B}, Tx_{AB}\})$ the strategy profile that corresponds to following the protocol at every node.

Definition 14. The *opening game* Γ^{op} is a game in extensive form, with two players $N = \{A, B\}$ and 4 nodes, labeled by a number (1 is the vertex):

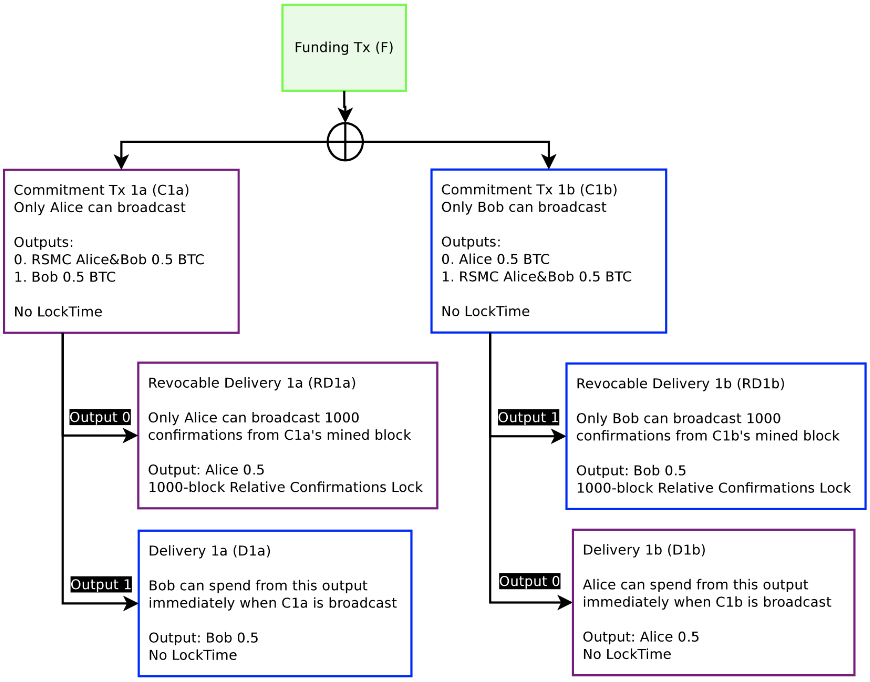
1. A has two actions available: $C1b_{\cdot\cdot}$, which provides outcome $(0, 0)$; $C1b_{A\cdot}$, which leads to node 2.

2. B has two actions available: $C1a_{\cdot\cdot}$, which provides outcome $(0, 0)$; $C1a_{\cdot B}$, which leads to node 3.

3. A has two actions available: $Tx_{..}$, which provides outcome $(0,0)$; $Tx_{A.}$, which leads to node 4.

4. B has two actions available: $Tx_{A.}$, which provides outcome $(0,0)$; Tx_{AB}, which provides outcome $(1,1)$.

At every node the player involved in the protocol have two actions available: either follow it or not follow it. If at any step they do not follow it, they get back to the initial state, with outcome $(0,0)$. If they do at every step, they are able to open the channel, with outcome $(0,0)$. The strategy profile recommended by the protocol is $\sigma^{op} = (\{C1b_{A.}, Tx_{A.}\}, (\{C1a_{.B}, Tx_{AB}\})$, in which the actions are played respectively at nodes $(\{1,3\},\{2,4\})$. The protocol is thus represented by the mechanism $(\Gamma^{op}, \sigma^{op})$, whose properties we analyze in the sequel.



Funding Tx (F)
Commitment Tx 1a (C1a)
Only Alice can broadcast
Outputs:
0. RSMC Alice&Bob 0.5 BTC
1. Bob 0.5 BTC
No LockTime
Commitment Tx 1b (C1b)
Only Bob can broadcast
Outputs:
0. Alice 0.5 BTC
1. RSMC Alice&Bob 0.5 BTC
No LockTime
Output 0
Revocable Delivery 1a (RD1a)
Only Alice can broadcast 1000
confirmations from C1a's mined block
Output: Alice 0.5
1000-block Relative Confirmations Lock
Output 1
Revocable Delivery 1b (RD1b)
Only Bob can broadcast 1000
confirmations from C1b's mined block
Output: Bob 0.5
1000-block Relative Confirmations Lock
Output 1
Delivery 1a (D1a)
Bob can spend from this output
immediately when C1a is broadcast
Output: Bob 0.5
No LockTime
Output 0
Delivery 1b (D1b)
Alice can spend from this output
immediately when C1b is broadcast
Output: Alice 0.5
No LockTime

Figure 11: Scheme of the commitments for the opening of a channel [51].

Theorem 10. *The mechanism $(\Gamma^{op}, \sigma^{op})$ is not immune.*

Proof. Since we are in a two-player setting, a mechanism is immune (cf. Definition 9) if it is 1-immune, i.e. if both players receive no lower payoff than $u(\sigma^{op}) = (1,1)$, no matter what the other player chooses. A counterexample is B deviating from $\sigma^{op}_B = \{C1a_{.B}, Tx_{AB}\}$ to $\tau_B = \{C1a_{..}, Tx_{AB}\}$, i.e. B refusing to signing $C1a$ at step 2. For player A the outcome of $u_A(\sigma^{op}_A, \tau_B) = 0 < 1 = u(\sigma^{op})$. □

Theorem 11. *The mechanism $(\Gamma^{op}, \sigma^{op})$ is optimal resilient and weak immune.*

Proof. The strategy profile σ^{op} provides the best outcome for both players $(1,1)$. Therefore, the mechanism $(\Gamma^{op}, \sigma^{op})$ is strongly resilient.
Both σ^{op}_A and σ^{op}_B are dominant strategies respectively for A and B, because they always get a better outcome, no matter what the other player does. Therefore σ^{op} survives after the iterated deletion of weakly dominated strategies: the mechanism is practical. The players never receive negative payoff therefore, if they play σ^{op}_A and σ^{op}_B they always get a non-negative payoff. This corresponds to the Definition 10 of weak immunity. □

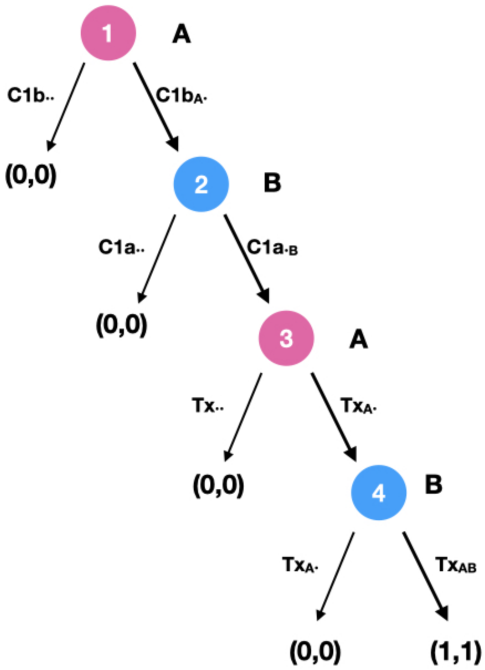


Figure 12: The game tree of Γ^{op}

4.5 Classical and alternative closing modules

As described in Section 4.4, both users A and B have a copy of a transaction that allows them to close the channel unilaterally. Indeed, A and B own respectively two commitments $C1a_{\cdot B}$ and $C1b_{A\cdot}$ signed by the other part. If they add their signature, respectively $C1a_{AB}$ and $C1b_{AB}$, they can unilaterally publish a transaction that returns the values stuck in the fund x_A and x_B back to their owners. If a user decides to unilaterally close the channel, she receives her part of the fund after that Δ blocks are validated on the Bitcoin blockchain, while the other user receives it immediately. The protocol recommends to close the channel by creating a new transaction, namely ES, that let the players receive their cryptoassets immediately. We model the situation with the following game in normal form.

Definition 15. The *closing game* $\Gamma^{cl} = \langle N, \mathscr{S}, u \rangle$ of the channel (x_A, x_B) with $x_A, x_B > 0$ is a game in normal form, with two players $N = \{A, B\}$ who have available three different pure strategies each: $\mathscr{S}_A = \{C1a_{AB}, DN, ES\}$ and $\mathscr{S}_B = \{C1b_{AB}, DN, ES\}$. The value of the utility can be found in the following payoff table.

			B	
		$C1b_{AB}$	DN	ES
	$C1a_{AB}$	$(\frac{1}{2}, \frac{1}{2})$	$(0,1)$	$(0,1)$
A	DN	$(1,0)$	$(-1,-1)$	$(-1,-1)$
	ES	$(1,0)$	$(-1,-1)$	$(1,1)$

First, we assume that the channel (x_A, x_B) is funded by both players i.e., $x_A, x_B > 0$. If one of the two players has no asset involved in the channel, we have to model the situation with a degenerate game, in which she can play any possible strategy. We recommend users to never unilaterally fund the channel. Indeed, if we drop the assumption that both players fund the channel, we have to consider a different modelisation. For instance, if B does not fund the channel we have that

$x_B = 0$. No matter what her strategy chooses, she gets nothing. We fix the utility of any outcome to 1 because it corresponds to the outcome of closing the channel. The payoff matrix of the game is the following:

$$\begin{array}{cc|c|c|c|}
 & & & B & \\
 & & C1b_{AB} & DN & ES \\ \hline
 & C1a_{AB} & (\frac{1}{2},1) & (0,1) & (0,1) \\ \hline
A & DN & (1,1) & (-1,1) & (-1,1) \\ \hline
 & ES & (1,1) & (-1,1) & (1,1) \\ \hline
\end{array}$$

This is a case of degenerate game, in which player B can theoretically choose any possible strategy, even doing nothing DN.

The players have three different strategies: publishing their commitment, seeking a deal to create a new transaction ES or just doing nothing DN. We assign null utility to players who receive their asset after Δ blocks, positive utility (normalized to 1) if they receive it immediately, negative utility if they cannot redeem their cryptoassets. The players receive null payoffs if they get their asset within Δ blocks, because they return to the initial state. For instance, this is case for player A if the strategy profile chosen by the players is $(C1a_{AB}, ES)$, i.e. if B seeks a deal but A unilaterally closes the channel. The players receive a positive outcome (normalised to 1) if they receive their asset immediately, as for instance if they reach a deal (ES, ES). The players receive a negative outcome (normalised to -1) if their asset is stuck in the channel, such as in the case in which A seeks a deal but B does nothing (DN, ES). In case both users decide to unilaterally close the channel $(C1a_{AB}, C2a_{AB})$, only one between $C1a$ and $C1b$ can be published. They have the same chance $(\frac{1}{2})$ for their transaction to published, leading to any of the state $(0,1)$ and $(1,0)$ with equivalent probability. Therefore the utility can be computed as a weighted average: $\frac{1}{2}(0,1) + \frac{1}{2}(1,0) = (\frac{1}{2}, \frac{1}{2})$.

The protocol recommends the strategy profile $\sigma^{cl} = (ES, ES)$ i.e., that both players seek a deal. In the following we analyze the properties of the mechanism $(\Gamma^{cl}, \sigma^{cl})$.

Theorem 12. *Under the assumption $x_A > 0a$ or $x_B > 0$, the mechanism $(\Gamma^{cl}, \sigma^{cl})$ is optimal resilient, but not weak immune.*

Proof. The utility $u(\sigma^{cl}) = (1,1)$ cannot be increased by any other strategy profile, therefore the mechanism $(\Gamma^{cl}, \sigma^{cl})$ is strongly resilient.
For both player the strategy DN is weakly dominated by the strategy ES. Indeed, no matter what the other player does, the ES always provides the same or even a better utility than DN. If we exclude both strategies DN the players have available only two strategies: $\{C1a_{AB}, ES\}$ and $\{C1b_{AB}, ES\}$. Once again, ES dominates the other strategy by providing a better outcome. The only strategy that survives the iterated deletion of weakly dominated strategies for both players is ES. Therefore the only stable Nash equilibrium is $\sigma^{cl} = (ES, ES)$. Thanks to Proposition 2 we can say that a stable equilibrium provides a practical mechanism.
To prove that the mechanism is not weak immune it is enough to show a counterexample. Indeed, if A chooses ES as required by the protocol and B chooses the Byzantine strategy DN, player A receives a negative outcome $u_A(\sigma^{cl}_A, DN) = u_A(ES, DN) = -1$. □

Since the mechanism is not weak immune, it is not immune either. We thus provide an alternative protocol that can satisfy the property of weak immunity.

Theorem 13. *Under the assumption $x_A > 0$ or $x_B > 0$, the only weak immune mechanism is (Γ^{cl}, σ^*) with $\sigma^* = (C1a_{AB}, C2a_{AB})$.*

Proof. In order to identify weak immune mechanisms we apply Proposition **??**. We consider player A and the game Γ^{cl}_A in which B is the adversarial player whose utility is the opposite of player A's. The payoff matrix of the game Γ^{cl}_A is the following.

$$\begin{array}{cc|c|c|c|}
 & & & B & \\
 & & C1b_{AB} & DN & ES \\ \hline
 & C1a_{AB} & (\frac{1}{2},-\frac{1}{2}) & (0,0) & (0,0) \\ \hline
A & DN & (1,-1) & (-1,1) & (-1,1) \\ \hline
 & ES & (1,-1) & (-1,1) & (1,-1) \\ \hline
\end{array}$$

The only Nash equilibria of the game in pure strategies is $(C1a_{AB}, DN)$, which provides outcome $(0, 0)$. Since this is a zero-sum game, all the Nash equilibria provide the same outcome (v, v) where $v = 0$ is the value of the game. Since the value of the game is non-negative, player A has always a strategy to get at least 0. This strategy is $C1a_{AB}$, which thus is the only one that player A can choose in a weak immune mechanism.
Analogously we can define the game Γ^{cl}_{B} in which A is the adversarial player, which lets us prove that $C1b_{AB}$ is the only weak immune strategy for player B. Therefore, $(C1a_{AB}, C1b_{AB})$ is the only strategy profile that provides a weak immune mechanism. □

We believe that Lightning Network should include the alternative protocol (Γ^{cl}, σ^*) as default. In the case in which the channel is unilaterally funded, one of the player is already forced to follow the mechanism (Γ^{cl}, σ^*). Listing all the possible strategies we have determined the only protocol which can be modeled as a weak immune mechanism. It is not possible to create any other protocol that can satisfy this property.

4.6 Updating module

Performing a transaction within a channel consists in updating its balance. Technically, the previous commitments ($C1a$ and $C1b$) with balance (x_A, x_B) are replaced by two new commitments ($C2a$ and $C2b$) with different balance (x'_A, x'_B). In order to prevent players from publishing old commitments, they sign two Breach Remedy Transactions ($BR1a$ and $BR1b$), that can invalidate $C1a$ and $C2b$. Indeed, if any party publishes an outdated commitment the other one can retrieve all the cryptoassets in the fund. If, for instance, user A publishes the outdated commitment $C1a$, she can retrieve her fund x_A unless user B publishes $BR1a$ before Δ blocks are validated. Briefly speaking, if any part publishes an outdated commitment the other part can retrieve all the assets in the fund. In practice the players have an incentive to delete outdated commitments to limit the risk of an unintentional leak, that could provoke their publication and thus the loss of all the assets stored in the channel. The protocol to update the balance (cf. Fig. 13) requires the players to sign the commitments in a specific order. The protocol involves the following steps:

1. A creates a transaction *C2b* that allows F to send x'_A to A and to send x'_B to B. B is able to spend x'_B only after that Δ blocks are validated. A signs *C2b* and sends it to B.

2. B creates a transaction *C2a* that allows F to send x'_A to A and to send x'_B to B. A is able to spend x'_A only after that Δ blocks are validated. B signs *C2a* and sends it to B.

3. A creates a transaction *BR1a* that lets B retrieve x_A in case A publishes *C1a* and B publishes *BR1a* within the following Δ blocks. Then A sends *BR1a* to B.

4. B creates a transaction *BR1b* that lets A retrieve x_B in case B publishes *C1b* and A publishes *BR1b* within the following Δ blocks. Then B sends *BR1b* to A.

We formalize the protocol with a game in extensive form Γ^{up} (cf. Definition 16), represented by the tree in Fig. 14. The initial state corresponds to the previous balance (with thus null utility), the final state to the updated balance (with utility equal to 1). One may question that with the updated balance one of the two party is receiving a smaller cryptoasset however, this does not consist in receiving a lower utility since updating the balance guarantees the exchange of a different cryptoasset which is more valuable than the one stored in the channel. We assign a negative value to the states in which players lose their cryptoassets or part of them.

Definition 16. The *updating game* Γ^{up} is a game in extensive form, with two players $N = \{A, B\}$ and 5 nodes, labeled by a number (1 is the vertex):

1. A has two actions available: $C2b_{..}$, which provides outcome $(0, 0)$; $C2b_{A.}$, which leads to node 2.

2. B has three actions available: $C2a_{..}$, which provides outcome $(0, 0)$; $C2b_{AB}$, which provides outcome $(1, 1)$; $C2a_{.B}$, which leads to node 3.

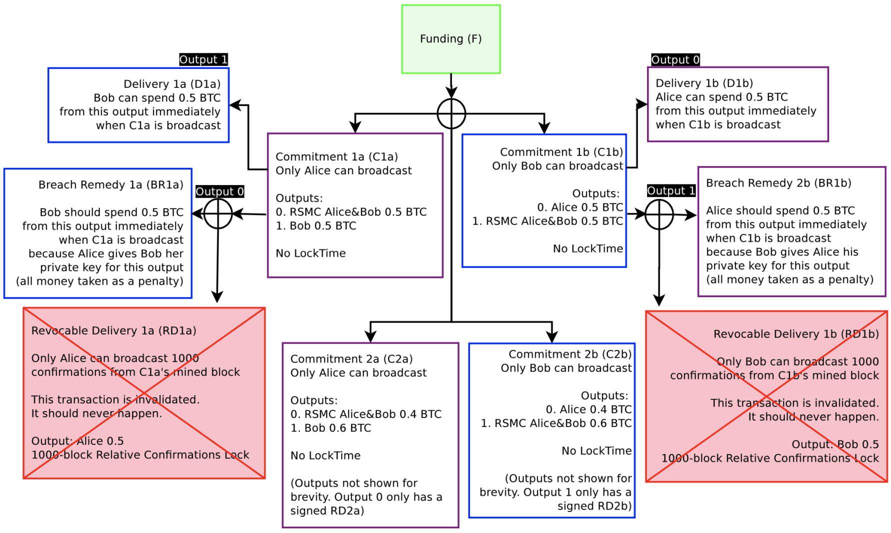

Funding (F)
Output 1
Delivery 1a (D1a)
Bob can spend 0.5 BTC
from this output immediately
when C1a is broadcast
Commitment 1a (C1a)
Only Alice can broadcast
Outputs:
0. RSMC Alice&Bob 0.5 BTC
1. Bob 0.5 BTC
No LockTime
Breach Remedy 1a (BR1a)
Bob should spend 0.5 BTC
from this output immediately
when C1a is broadcast
because Alice gives Bob her
private key for this output
(all money taken as a penalty)
Output 0
Revocable Delivery 1a (RD1a)
Only Alice can broadcast 1000
confirmations from C1a's mined block
This transaction is invalidated.
It should never happen.
Output: Alice 0.5
1000-block Relative Confirmations Lock
Commitment 2a (C2a)
Only Alice can broadcast
Outputs:
0. RSMC Alice&Bob 0.4 BTC
1. Bob 0.6 BTC
No LockTime
(Outputs not shown for
brevity. Output 0 only has a
signed RD2a)
Output 0
Delivery 1b (D1b)
Alice can spend 0.5 BTC
from this output immediately
when C1b is broadcast
Commitment 1b (C1b)
Only Bob can broadcast
Outputs:
0. Alice 0.5 BTC
1. RSMC Alice&Bob 0.5 BTC
No LockTime
Output 1
Breach Remedy 2b (BR1b)
Alice should spend 0.5 BTC
from this output immediately
when C1b is broadcast
because Bob gives Alice his
private key for this output
(all money taken as a penalty)
Commitment 2b (C2b)
Only Bob can broadcast
Outputs:
0. Alice 0.4 BTC
1. RSMC Alice&Bob 0.6 BTC
No LockTime
(Outputs not shown for
brevity. Output 1 only has a
signed RD2b)
Revocable Delivery 1b (RD1b)
Only Bob can broadcast 1000
confirmations from C1b's mined block
This transaction is invalidated.
It should never happen.
Output: Bob 0.5
1000-block Relative Confirmations Lock

Figure 13: Scheme of the commitments to update the balance of the channel [51].

3. A has three actions available: $BR1a_{..}$, which provides outcome $(0,0)$; $C2a_{AB}$, which provides outcome $(1,1)$; $BR1a_{A.}$, which leads to node 4.

4. B has two actions available: $BR1b_{.B}$, which provides outcome $(1,1)$; $BR1b_{..}$, which leads to node 5.

5. A has two actions available: $C1a_{AB}$, which provides outcome $(-1,1)$; $C2a_{AB}$, which provides outcome $(1,1)$.

The protocol recommends to sign all the commitments and it is thus represented by the strategy profile $\sigma^{up} = (\{C2b_{A.}, BR1a_{A.}, C2a_{AB}\}, \{C2a_{.B}, BR1b_{.B}\})$, in which the actions are played respectively at nodes $(\{1,3,5\},\{2,4\})$. At nodes 2 and 3 respectively users B and A can enforce the new commitments by publishing them on the Bitcoin blockchain and thus closing the channel. At node 4 user B can refuse to provide the breach remedy transaction to user A, who at node 5 can then publish the new commitment enforcing the closure of the channel. If at node 5 user A publishes the old commitment $C1a$, user B can retrieve all the funds by publishing the breach remedy transaction $BR1a$.
We analyze the properties of the mechanism $(\Gamma^{up}, \sigma^{up})$ under the assumption that it is always possible to publish a transaction within Δ blocks, otherwise it is not possible to validate the breach remedy transactions in time. The mechanism is not immune, indeed if any user refuses to sign a commitment the players return to the original balance that provides lower payoff than the final balance. However, the mechanism satisfies the properties of optimal resilience and weak immunity.

Theorem 14. *The mechanism $(\Gamma^{up}, \sigma^{up})$ is not immune.*

Proof. Since we are considering a game with only two players, a mechanism is immune if it is 1-immune. A mechanism is 1-immune (cf. Definition 9) if any player receives the same outcome by playing the recommended strategy, no matter which strategy the other player chooses. This is not the case of the mechanism $(\Gamma^{up}, \sigma^{up})$, indeed if player A chooses σ^{up}_A and player B chooses $\{C2a_{..}, BR1b_{.B}\} \neq \sigma^{up}_B$ the payoff for player A is $u_A(\sigma^{up}_A, \{C2a_{..}, BR1b_{.B}\}) = 0 < 1 = u_A(\sigma^{up}_A, \sigma^{up}_B)$. □

The property of immunity is too strong in this case, therefore we consider other weaker properties.

Theorem 15. *The mechanism $(\Gamma^{up}, \sigma^{up})$ is optimal resilient and weak immune with probability $1 - \Phi_{\Delta}(\alpha)$, but it is not immune.*

Proof. We analyze the mechanism $(\Gamma^{up}, \sigma^{up})$ under the assumption that it is always possible to publish a transaction within Δ blocks, otherwise it is not possible to validate the breach remedy transactions in time. The probability that this happens when a byzantine agent with computational power α attacks the Bitcoin blockchain is $1 - \Phi_{\Delta}(\alpha)$ (cf. Theorem 9).
The outcome for the strategy profile σ^{up} is $(1, 1)$, which cannot be increased by any other strategy profile. Therefore, the mechanism $(\Gamma^{up}, \sigma^{up})$ is strongly resilient.
In order to prove that the mechanism is resilient, we have to exclude weakly dominated strategies. Since it is cumbersome to list all the strategies, we proceed by excluding all the actions that are included in a weakly dominated strategy. At node 1 A receives always a better outcome by picking action $C2b_{A\cdot}$ rather than $C2b_{\cdot\cdot}$, thus $C2b_{\cdot\cdot}$ is never included in a practical mechanism. At node 2 B never plays the action $C2a_{\cdot\cdot}$, at node 3 A never plays $BR1a_{\cdot\cdot}$ and at node 5 A never plays $C1a_{AB}$. The remaining strategy profiles, included σ^{up}, provide outcome $(1, 1)$. Since they all survive the iterated deletion of weakly dominated strategies, they are all practical mechanisms. Thanks to Corollary 1 we know that there always exists at least one practical mechanism. However, the reader should keep in mind that this might not be unique.
In order to prove that the mechanism is weak immune we apply Proposition 4. We consider one player i at a time and we make the other player j adversarial, by fixing her outcome as the opposite of player i (cf. Fig. 15). Then we prove that the best response of player j to player i never leads her to a negative outcome. We take $i = A$ and we consider the game Γ_A^{up} in which player $j = B$ has utility opposite to player i. The best response of player j to the strategy σ_A^{up} picked by player i is the strategy $\{C2a_{\cdot\cdot}, BR1b_{\cdot\cdot}\}$, i.e. at node 2 to avoid to reach a deal by not signing $C2a$. The payoff for player A is $u_A(\sigma_A^{up}, \{C2a_{\cdot\cdot}, BR1b_{\cdot\cdot}\}) = 0$, which is non-negative. Analogously we consider the game Γ_B^{up} in which $i = B$ is the picked player and $j = A$ is the adversarial player, with utility opposite to player i. The best response for j to strategy σ_B^{up} is $\{C2b_{\cdot\cdot}, BR1a_{\cdot\cdot}, x\}$ with x any possible action at node 5, which provides a non-negative payoff $u_B(\{C2b_{\cdot\cdot}, BR1a_{\cdot\cdot}, x\}, \sigma_B^{up}) = 0$. Since both adversarial games provide non-negative payoff, thanks to Proposition 4 we get that the mechanism is weak immune. □

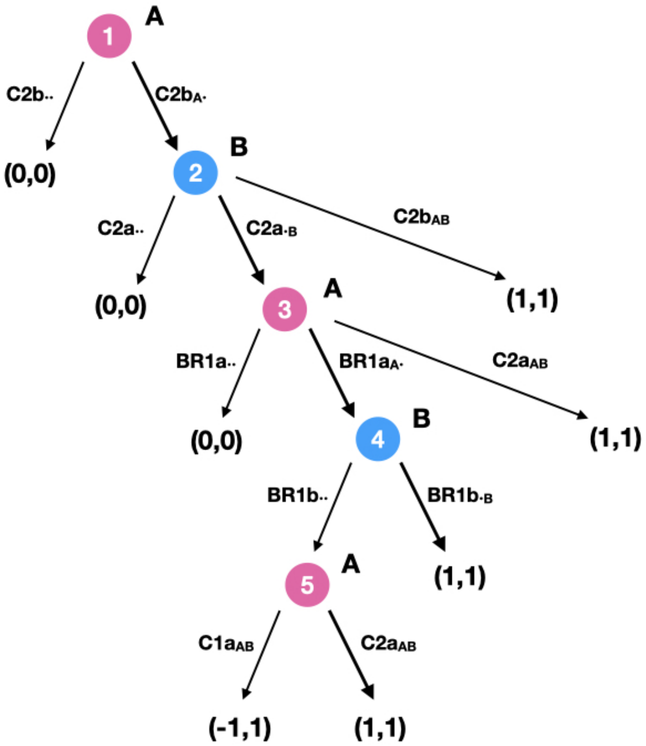


Figure 14: The game tree of Γ^{up}

4.7 Hashed Timelock Contract module

A bidirectional payment channel only allows transactions inside a channel. In order to perform transactions through a network of channels Lightning Network introduces an additional construction, called Hashed Timelock Contract (HTLC). The HTLC allows to create transactions that can be triggered at will. The HTLC makes use of the *hash function*, a deterministic caotic function that maps any input x to a fixed-length string $y = hash(x)$. It is not possible to retrieve x given y in a faster way than trying with a bruce-force method to randomly guess x. Hence if x is chosen among strings of considerable length, it is almost impossible to identify x given by $y = hash(x)$ in a reasonable time. Let us suppose that users A and B open a channel with balance (x_A, x_B) and A wants to send a payment through HTLC to B so that the new balance would be (x'_A, x'_B), with $x_A < x'_A$. A creates a random data R and then computes $H = hash(R)$. Then she sends an update of the contract to B, with a specific characteristic: if B publishes it, she can retrieve the difference $x'_B - x_B$ only if she proves to know x such that $H = hash(x)$ within Δ blocks (in [51] $\Delta = 1000$). A can trigger the contract by providing R to B. If she does not do it, B cannot find $x = R$ and thus has no incentive to publish the contract. The HTLC protocol works as follows (cf. Fig. 16):

1. A creates a commitment *C2b* that allows F to send x'_A to A, x_B to B after Δ blocks and $x'_B - x_B$ to B if she publishes x such that $H = hash(x)$ to the Bitcoin blockchain within Δ blocks. A signs it and sends it to B.

2. Analogously, B creates a set of commitment *C2a* that allows F to send x'_B to B, x_A to A after Δ blocks and $x'_B - x_B$ to B if she publishes x such that $H = hash(x)$ to the Bitcoin blockchain within Δ blocks. B signs it and sends it to A.

3. A creates a transaction *BR1a* that lets B retrieve x_A in case A publishes *C1a* and B publishes *BR1a* within the following Δ blocks. Then A sends *BR1a* to B.

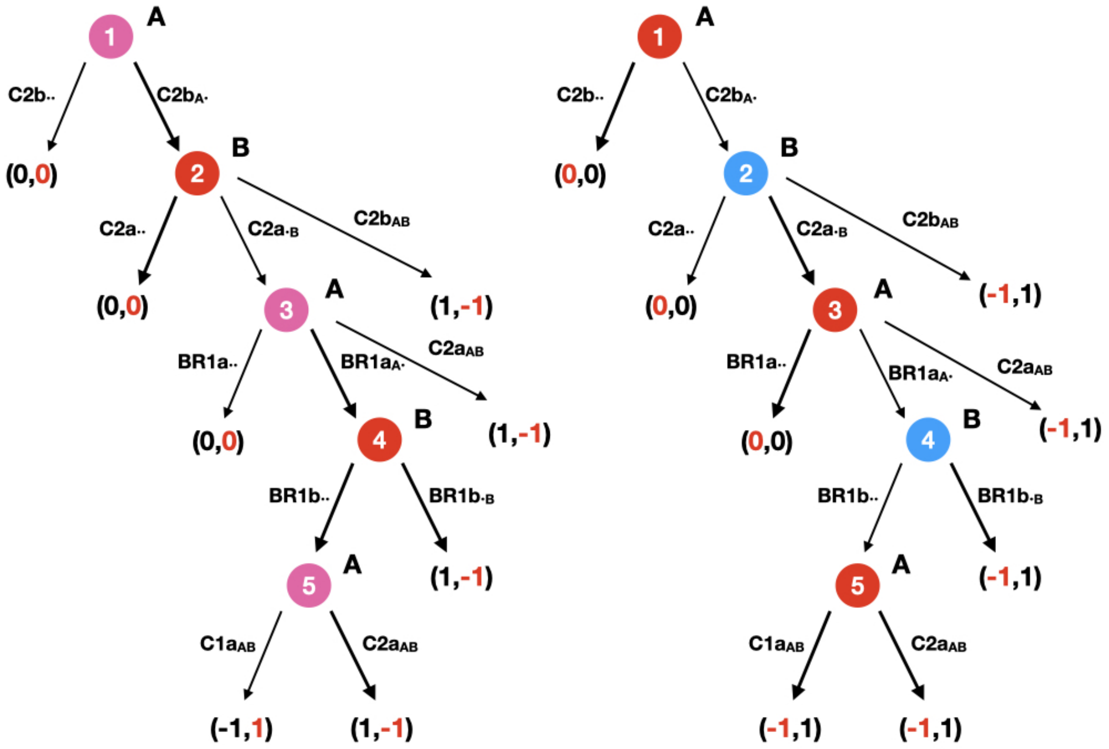


Figure 15: The game trees of Γ_A^{up} and Γ_B^{up}

4. B creates a transaction *BR1b* that lets A retrieve x_B in case B publishes *C1b* and A publishes *BR1b* within the following Δ blocks. Then B sends *BR1b* to A.

The protocol for the HTLC corresponds to the protocol for updating a channel, with the only difference that the new commitments *C2a* and *C2b* provide a different output. Under the assumption that a transaction (or just the key R) can be published within Δ blocks, we can define a game Γ^{htlc} with the very same structure as Γ^{up} (cf. Definition 16 and Fig. 14). Following the protocol corresponds to the strategy profile σ^{htlc}. Hence we can introduce the following theorem.

Theorem 16. *The mechanism* $(\Gamma^{htlc}, \sigma^{htlc})$ *is optimal resilient and weak immune, but not immune, with probability* $1 - \Phi_\Delta(\alpha)$*.*

Proof. Since the mechanisms $(\Gamma^{htlc}, \sigma^{htlc})$ and $(\Gamma^{up}, \sigma^{up})$ follow the very same structure, we can apply Theorem 15. □

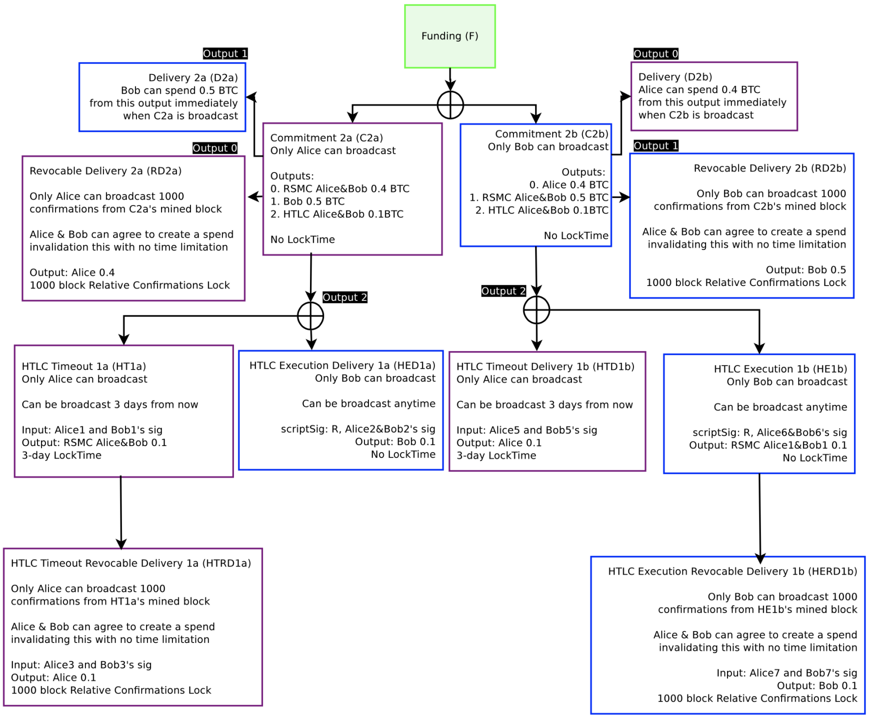

Funding (F)
Output 1
Delivery 2a (D2a)
Bob can spend 0.5 BTC
from this output immediately
when C2a is broadcast
Output 0
Revocable Delivery 2a (RD2a)
Only Alice can broadcast 1000
confirmations from C2a's mined block
Alice & Bob can agree to create a spend
invalidation this with no time limitation
Output: Alice 0.4
1000 block Relative Confirmations Lock
Commitment 2a (C2a)
Only Alice can broadcast
Outputs:
0. RSMC Alice&Bob 0.4 BTC
1. Bob 0.5 BTC
2. HTLC Alice&Bob 0.1BTC
No LockTime
Commitment 2b (C2b)
Only Bob can broadcast
Outputs:
0. Alice 0.4 BTC
1. RSMC Alice&Bob 0.5 BTC
2. HTLC Alice&Bob 0.1BTC
No LockTime
Output 0
Delivery (D2b)
Alice can spend 0.4 BTC
from this output immediately
when C2b is broadcast
Output 1
Revocable Delivery 2b (RD2b)
Only Bob can broadcast 1000
confirmations from C2b's mined block
Alice & Bob can agree to create a spend
invalidating this with no time limitation
Output: Bob 0.5
1000 block Relative Confirmations Lock
Output 2
Output 2
HTLC Timeout 1a (HT1a)
Only Alice can broadcast
Can be broadcast 3 days from now
Input: Alice1 and Bob1's sig
Output: RSMC Alice&Bob 0.1
3-day LockTime
HTLC Execution Delivery 1a (HED1a)
Only Bob can broadcast
Can be broadcast anytime
scriptSig: R, Alice2&Bob2's sig
Output: Bob 0.1
No LockTime
HTLC Timeout Delivery 1b (HTD1b)
Only Alice can broadcast
Can be broadcast 3 days from now
Input: Alice5 and Bob5's sig
Output: Alice 0.1
3-day LockTime
HTLC Execution 1b (HE1b)
Only Bob can broadcast
Can be broadcast anytime
scriptSig: R, Alice6&Bob6's sig
Output: RSMC Alice1&Bob1 0.1
No LockTime
HTLC Timeout Revocable Delivery 1a (HTRD1a)
Only Alice can broadcast 1000
confirmations from HT1a's mined block
Alice & Bob can agree to create a spend
invalidating this with no time limitation
Input: Alice3 and Bob3's sig
Output: Alice 0.1
1000 block Relative Confirmations Lock
HTLC Execution Revocable Delivery 1b (HERD1b)
Only Bob can broadcast 1000
confirmations from HE1b's mined block
Alice & Bob can agree to create a spend
invalidating this with no time limitation
Input: Alice7 and Bob7's sig
Output: Bob 0.1
1000 block Relative Confirmations Lock

Figure 16: Scheme of the commitments of the HTLC [51].

4.8 Routing module

The *Hashtime Locked Contract* (HTLC) allows to create transactions that can be triggered at will. Summing up what presented in Section 4.7 for technical details, the protocol for the HTLC works as follows. User A creates a pair (H, R), where H is public and R is its private key. She shares with user B a commitment together with the string H. Once this commitment is published on the Bitcoin blockchain, user B can receive the transaction only if she can provide the private key R within Δ blocks. It is easy to check that R is the private key of H, but it is almost impossible to retrieve R, given H. In this way, user A can trigger the transaction whenever she wants by disclosing R to user B. The protocol is represented by the mechanism $(\Gamma^{htlc}, \sigma^{htlc})$, that has the very same structure of the updating module (cf. Section 4.6) and thus satisfies optimal resilience and weak immunity, but not immunity.

Lightning Network allows payments also between two users, namely A and C, who do not share a channel. The requirement for a *routed payment* is to find a path of channels between the two users, i.e. a sequence of users who two-by-two share a channel. For instance, let us suppose that users A and C have both opened a separate channel with a third user B. In the *routed payment* user B is the intermediate node. The HTLC is implicated in the protocol that allows users to perform routed payments, which works as follows. Let us consider the case of a single intermediate node, namely B: users A and B have an opened channel with balance (x_A, x_B), while B and C have opened a different channel with balance (y_B, y_C). Let us suppose that A wishes to send δ to C. Informally, A sends $\delta + \epsilon$ to B and B sends δ to C, where $\epsilon \geq 0$ is the fee given to the intermediate node B. Since the channel are opened the two payments consists in updating the balance of the two channels: $(x_A, x_B) \to (x_A - \delta - \epsilon, x_B + \delta + \epsilon)$ and $(y_B, y_C) \to (y_B - \delta, y_C + \delta)$. The protocol for routed payments lets the receiver C trigger both payments at the same moment:

1. C creates a random data R and hashes it: $H = hash(R)$. Then, she sends H to A.

2. A creates a HTLC, namely H^{AB} of value $\delta + \epsilon$ locked with H and sends it to B.

3. B creates a HTLC, namely H^{BC} of value δ locked with H and sends it to C.

4. C discloses R to B, hence validating H^{BC}.

5. B discloses R to A, thus validating H^{AB}.

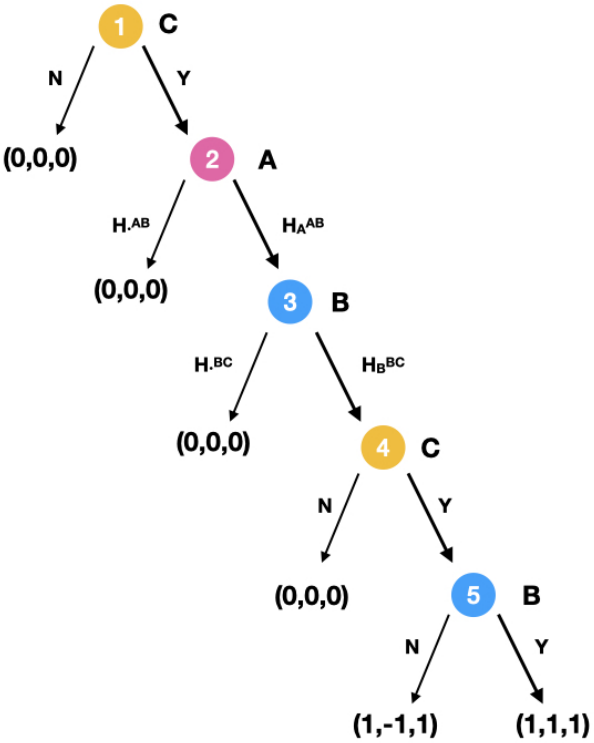


Figure 17: The game tree of Γ^{rout}

We formalize the protocol with a game in extensive form Γ^{rout}, whose tree is displayed in Fig. 17. The initial state consists in the initial balance and it is assigned null utility. The final state corresponds for A and C to fulfill the payment, for B to receive the fee ϵ. The final state has positive payoff, normalised to 1. Any state that consists in a loss of assets is assigned negative payoff. The strategy profile recommended by the protocol is denoted by $\sigma^{rout} = (\{H_A^{AB}\}, \{H_B^{BC}, Y\}, \{Y, Y\})$.

Definition 17. The *routing game* Γ^{rout} is a game in extensive form, with three players $N = \{A, B, C\}$ and 5 nodes, labeled by a number (1 is the vertex):

1. C has two actions available: either N, not sending H to A, which provides outcome $(0, 0, 0)$, or Y, sending H to A, which leads to node 2.

2. A has two actions available: either $H_{\cdot}^{AB}$, which provides outcome $(0, 0, 0)$, or H_A^{AB}, which leads to node 3.

3. B has two actions available: either $H_{\cdot}^{BC}$, which provides outcome $(0, 0, 0)$, or H_B^{BC}, which leads to node 4.

4. C has two actions available: either N, not disclosing R to B, which provides outcome $(0, 0, 0)$, or Y, disclosing R to B, which leads to node 5.

5. B has two actions available: either N, not disclosing R to A, which provides outcome $(1, -1, 1)$ or Y, disclosing R to A, which provides outcome $(1, 1, 1)$.

At node 1 C creates the lock H and its key R. At node 2 and 3 the two HTLCs are created. At node 4 C triggers the payment in the channel that she shares with B. At node 5 B triggers the payment in the channel that she shares with A. If at step 5 B does not trigger the payment, A and

C reach the final state, because C has received the payment, also if A has not paid for it.
The recommended strategy profile is $\sigma^{rout} = (\{H_A^{AB}\}, \{H_B^{BC}, Y\}, \{Y, Y\})$, respectively played at nodes $(\{2\}, \{3, 5\}, \{1, 4\})$. The payoff are as shown only under the assumption that in both HTLCs the transactions can be triggered. We analyze the protocol under this assumption.

The following theorems state that the mechanism corresponding to the routed payment protocol is not immune but is weak immune and optimal resilient.

Theorem 17. *$(\Gamma^{rout}, \sigma^{rout})$ is not immune.*

Proof. Since the game Γ^{rout} has three players, the mechanism is immune if it is 1-immune and 2-immune. To prove that the mechanism is not immune, it is enough to prove that it is not 1-immune. A mechanism is 1-immune (cf. Definition 9) if any player who chooses the recommended strategy receives the same outcome, no matter what any Byzantine player can choose. This property is not fulfilled. Indeed, if A picks the strategy $H_{\cdot}^{AB}$, the outcome for C is lower: $u_C(H_{\cdot}^{AB}, \sigma_B^{rout}, \sigma_C^{rout}) = 0 < 1 = u_C(\sigma_A^{rout}, \sigma_B^{rout}, \sigma_C^{rout}) = u_C(\sigma^{rout})$. □

The property of immunity is too strong for this protocol, therefore we consider the other properties.

Theorem 18. *Under the assumption that in both HTLCs the transactions can be triggered, $(\Gamma^{rout}, \sigma^{rout})$ is optimal resilient and weak immune.*

Proof. There is no other strategy than σ^{rout} that can improve any of its payoffs $u(\sigma^{rout}) = (1, 1, 1)$. Thus $(\Gamma^{rout}, \sigma^{rout})$ is a strongly resilient mechanism.
In order to prove that the mechanism is practical, we proceed by excluding the actions that belongs to weakly dominated strategies. At node 5 B never plays N because she would receive -1 rather than 1. Therefore at node 4 C never chooses N because she would receive 0 rather than 1. Analogously at nodes 3, 2 and 1 players do not choose alternative actions, because they would receive 0 rather than 1. The strategy profile σ^{rout} is the only one that survives the iterated deletion of weakly dominated strategies, hence the mechanism is practical.
In order to prove that the mechanism is weak immune we apply Proposition 4. We consider one player i at a time and we introduce an adversarial player j that plays at any node which is not played by i (cf. Fig. 18). We define the game Γ_i^{rout} which has the same structure, two players i and j and utility function for j opposite to the one of player i. In games Γ_A^{rout} and Γ_C^{rout} respectively A and C never receive negative payoffs. In game Γ_B^{rout} player B never receives negative payoff if she plays σ_B^{rout}. For Proposition 4, since all the adversarial games Γ_i^{rout} do not provide negative payoff if the players follow the recommended strategy σ_i^{rout}, the mechanism is weak immune. □

The HTLCs introduced in the protocol work independently from the routing protocol. We can model them with two different mechanisms: $(\Gamma^{AB}, \sigma^{AB})$ for H^{AB} and $(\Gamma^{BC}, \sigma^{BC})$ for H^{BC}. The mechanism $(\Gamma^{AB}, \sigma^{AB})$ represents the HTLC deployed on the channel A-B, while the mechanism $(\Gamma^{BC}, \sigma^{BC})$ refers to the HTLC implemented on the channel B-C. The HTLCs belong to two different channels, so they are independent one from another. The assumption from the routing protocol is that in both HTLCs the transactions can be triggered, but this is true only if every transaction can be published within Δ blocks (cf. Section 4.7). Under this assumption, the protocol for routed payments is independent from the protocol for HTLC, because it is external with respect to the channel, while the HTLCs work within the channel. The routed payment is thus represented by three independent protocols $(\Gamma^{rout}, \sigma^{rout})$, $(\Gamma^{AB}, \sigma^{AB})$, and $(\Gamma^{BC}, \sigma^{BC})$. Therefore we analyze the properties of its mechanism by defining the composition of the three games $(\Gamma^{rout} \odot \Gamma^{AB} \odot \Gamma^{BC}, \{\sigma_i^{rout}, \sigma_i^{AB}, \sigma_i^{BC}\})$.

Theorem 19. *The mechanism $(\Gamma^{rout} \odot \Gamma^{AB} \odot \Gamma^{BC}, \{\sigma_i^{rout}, \sigma_i^{AB}, \sigma_i^{BC}\})$ is optimal resilient and weak immune with probability $1 - \Phi_\Delta(\alpha)$.*

Proof. We analyze the mechanism $(\Gamma^{up}, \sigma^{up})$ under the assumption that it is always possible to publish a transaction within Δ blocks, otherwise it is not possible to validate the breach remedy transactions in time. The probability that this happens when a byzantine agent with computational power α attacks the Bitcoin blockchain is $1 - \Phi_\Delta(\alpha)$ (cf. Theorem 9). The operator composition (cf. Definition 11) is invariant with respect the properties of the mechanisms. Thanks to Theorems

16 and 18 we have that $(\Gamma^{rout}, \sigma^{rout})$, $(\Gamma^{AB}, \sigma^{AB})$ and $(\Gamma^{BC}, \sigma^{BC})$ are practical. Therefore, with Theorem 2 we have that their composition $(\Gamma^{rout} \odot \Gamma^{AB} \odot \Gamma^{BC}, \{\sigma_i^{rout}, \sigma_i^{AB}, \sigma_i^{BC}\})$ is practical. Analogously, thanks to Theorems 16 and 18 we have that every single mechanism is k-resilient for all k and t-weak-immune for all t. Theorems 3 and 4 allow us to say that the composition $(\Gamma^{rout} \odot \Gamma^{AB} \odot \Gamma^{BC}, \{\sigma_i^{rout}, \sigma_i^{AB}, \sigma_i^{BC}\})$ is k-resilient for all k and t-weak-immune for all t i.e., it is strongly resilient and weak immune. □

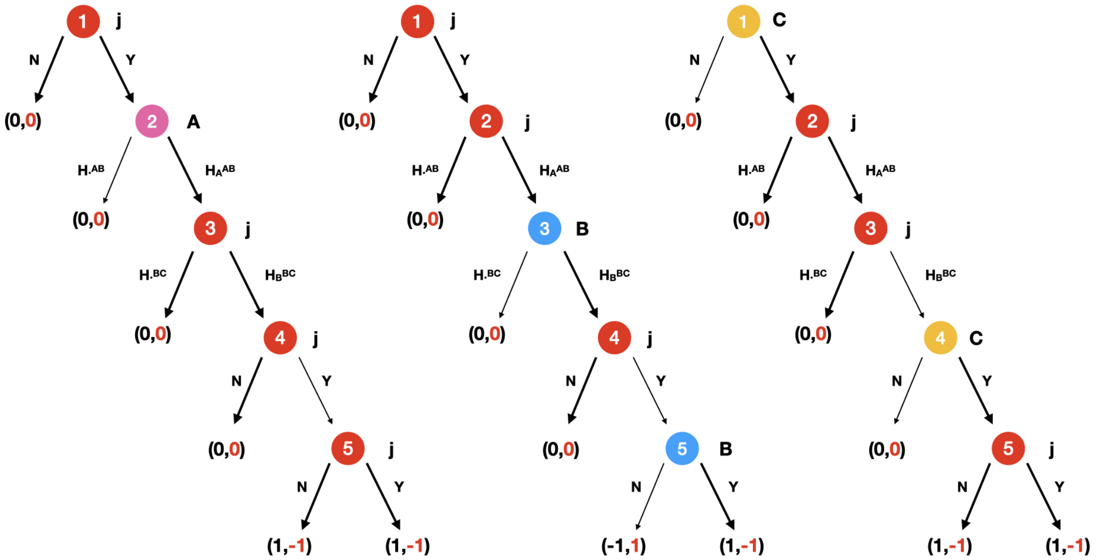


Figure 18: The game trees of Γ_A^{rout}, Γ_B^{rout} and Γ_C^{rout}

4.8.1 Recap

All the results of the Lightning Network are available in Table 1. The Lightning Network is built on top of Bitcoin blockchain. Therefore its properties depend highly on Bitcoin blockchain's ones. If we exclude the closing protocol, the Lightning Network satisfies optimal resilience and weak immunity. Therefore we can compose (cf. Definition 11) its protocols' games with Bitcoin mechanism's, which provide weaker results, and prove that the Lightning Network satisfies the same properties of the Bitcoin mechanism.

4.9 Side-chain

A different solution to overcome the scalability and privacy problems of blockchains is offered by Platypus [50], a protocol that allows a group of users to create a childchain (sidechain) that can handle off chain transactions without the need of synchrony among peers. In this section we consider the protocol to create a Platypus chain, described in Fig. 19. The protocol let the childchain validators broadcast transactions to the peers until the number of validators that have confirmed the transactions overcome a defined threshold.

It is possible to model this protocol with a game in extensive form Γ^{cr}, in which players are split into two categories: normal users (set U) and the validators (set V). Users' utility is positive if their transactions are successfully published and it is negative if a different wrong transaction is validated instead of hers. Normal users have utility 1 if their transaction is successfully published, 0 if they get back to the initial state, -1 if they lose anything in the process. The validators have utility n, with n the number of valid transactions which are broadcast. The protocol is divided into phases. Every phase consists of players acting at the same time, indeed we work under the assumption that the broadcast of any of the players involved is subsequent to the action of every other player. If this condition is not fulfilled, it would be necessary to consider different phases instead of one, with the same structure.

Definition 18. The *creation game* is a game Γ^{cr} in extensive form, where $N = U \cup V$ is the set of players, with $|N| = m_v$. Every phase corresponds to a node of the tree, at which players play at the same time.

- Phase 1; only the player p_0 is involved. The player p_0 has two actions: either complete it Y or not N. If she does not, the outcome is 0 for all players.
- Phase 2; every player within normal users play at the same time. Everyone dispose of the same two actions: broadcasting their message Y or not N. If the message is not broadcast for player i, her utility is always 0.
- Phase 3; the validators can choose within a set of actions a_u with $u \subseteq U$ i.e., they can validate all the messages for the users within the set u. The cardinality of the set of their actions is equal to $2^{|U|}$. The utility for the validators corresponds to the number of valid transactions which are broadcast.
- Phase 4; the validators can choose within a set of actions in the form $(b_t, s_{t'})$, where t and t' are any subset of transactions broadcast in Phase 3. The action b consists in broadcasting the transactions belonging to the set t until $\lfloor 2m_v/3 \rfloor + 1$ validators receive it, while s means to send the transactions in t'.

We define the mechanism $(\Gamma^{cr}, \sigma^{cr})$, where $\sigma^{cr} \in \mathscr{S}$ is the strategy of following the protocol i.e., for normal users u the strategy is $\sigma^{cr}_u = Y$, while for validators v the strategy is $\sigma^{cr}_v = (a_{u^*}, b_{t^*}, s_{t^*})$, where u^* is the set of users who send a message and t^* is the set of transactions broadcast in Phase 3. We thus analyze the properties of the mechanism.

Theorem 20. *The mechanism* $(\Gamma^{cr}, \sigma^{cr})$ *is not* t*-immune for any* t*.*

Proof. It is enough to prove that the mechanism is not 1-immune. A mechanism is 1-immune if every player does not reduce her utility if only one other player is choosing a Byzantine behaviour (cf. Definition 9). This property is not fulfilled, indeed if in Phase 1 the process p_0 chooses N rather than $\sigma^{cr}_{p_0} = Y$, the utility for every player is 0, which is lower than the utility provided by σ^{cr}. □

In [50] it is proved that no wrong transaction can be validated if there are at most $\lfloor \frac{m_v}{3} \rfloor$ corrupted players. This property cannot be expressed with the concept of immunity, which is too strong; to capture this information we exploit the definition of t-weak-immunity (cf. Definition 10). Within our model, the upper bound on the number of corrupted players means that no negative payoff is given to the players under the hypothesis that there are at most $\lfloor \frac{m_v}{3} \rfloor$ Byzantine nodes i.e., that the mechanism is $\lfloor \frac{m_v}{3} \rfloor$-weak-immune.

Algorithm 1 Platypus creation procedure

```
▷ State of the algorithm
$\Omega$, the parentchain
$\Gamma$, the Platypus protocol
$P_\Omega$, the set of processes in the parentchain
$P_\Psi \leftarrow \perp$, the set of processes in the Platypus chain
$V_\Psi \leftarrow \perp$, the set of validators in the Platypus chain
$m_v$, the amount of validators required in $\Psi$
$\mathbb{C}_i$, coins that belong to process $p_i$
$job_i$, boolean defining if $p_i$ is VALIDATOR or just USER
$plid$, the Platypus chain identifier
$msg_i = \langle \mathbb{C}_i, plid, job_i \, \sigma_i \rangle$, signed message to join.
$\sigma_i$, signature of $msg_i$ by $p_i$
$tx_{plcr} \leftarrow \perp$, the Platypus creation transaction

▷ PHASE 1: process $p_0$ initiates request
$msg_0 \leftarrow \mathsf{sign}(\langle \mathbb{C}_0, plid, \mathsf{job_0} \rangle)$
$\mathsf{multicast}(msg_0)$ to $P_\Omega$

▷ PHASE 2: Rest of processes who want to join reply
when $msg_0$ is received from $p_0$
$msg_i \leftarrow \mathsf{sign}(\langle \mathbb{C}_\mathsf{i}, plid, \mathsf{job_i} \rangle)$
$\mathsf{multicast}(msg_i)$ to $P_\Omega$

▷ PHASE 3: Validator $p_i \in V_\Psi$ gathers enough validators
when $msg_j$ is received from $p_j$ and $p_j \notin P_\Psi$
$\{P_\Psi, \mathbb{C}_{P_\Psi}\} \leftarrow \{P_\Psi \cup \{p_j\},\ \mathbb{C}_{P_\Psi} \cup msg_j.\mathbb{C}_j\}$
if $(msg_j.job_j = \mathsf{VALIDATOR}$ and $p_j \notin V_\Psi)$ then
    $\{V_\Psi, \mathbb{C}_{V_\Psi}\} \leftarrow \{V_\Psi \cup \{p_j\},\ \mathbb{C}_{V_\Psi} \cup msg_j.\mathbb{C}_j\}$
    if $(|V_\Psi| = m_v)$ then                    ▷ Enough validators to start transaction
        $tx_{plcr} \leftarrow \mathsf{createPlatypusTx}(\mathbb{C}_{P_\Psi}, \mathbb{C}_{V_\Psi}, plid)$
        $tx_{plcr} \leftarrow \mathsf{sign_i}(tx_{plcr})$
        $\mathsf{multicast}(tx_{plcr}, \{msg_k\}_{p_k \in P_\Psi})$ to $V_\Psi$

▷ PHASE 4: $p_i \in V_\Psi$ signs and broadcasts until it gets enough signatures
when $(tx_{plcr}, \{msg_j\}_{p_j \in P_\Psi})$ is received and not $\mathsf{is\_written}(\Omega, tx_{plcr}, plid)$    ▷ if
$tx_{plcr}$ with $plid$ not written in $\Omega$
if $(\mathsf{verify}(tx_{plcr}, \{msg_j\}))$ then $tx_{plcr} \leftarrow \mathsf{sign_i}(tx_{plcr})$
    if $(\mathsf{num\_signers}(tx_{plcr}) < \lfloor 2m_v/3 \rfloor + 1)$ then
        $\mathsf{multicast}(tx_{plcr}, \{msg_j\})$ to $V_\Psi$
    else $\Gamma.\mathsf{send}(\Omega, tx_{plcr})$                    ▷ enough signatures
```

Figure 19: Algorithm to create a chain in Platypus [50].

Theorem 21. *The mechanism $(\Gamma^{cr}, \sigma^{cr})$ is optimal resilient and $\lfloor \frac{m_v}{3} \rfloor$-weak-immune.*

Proof. Under the strategy profile σ^{cr} the validators consider all the processes ($u = t = U$), thus their utility reach its maximum $|U|$. The other users have only two strategies, where broadcasting their message is the only strategy played at the equilibrium. Therefore the payoffs generated by σ^{cr} cannot be increased and the mechanism Γ^{cr}, σ^{cr} is strongly resilient.
For normal users the strategy Y dominates N (the utility is 1 which is larger then 0), while for validators (a_U, b_U, s_U) dominates every other strategy: indeed, any other strategy would provide a payoff lower than $|U|$. Therefore the strategy profile σ^{cr} is the only one with weakly dominating strategies, thus thanks to Proposition **??** we get that the mechanism is practical.
In order to prove weak immunity, we apply Proposition **??**. We need to prove that every player never gets negative utility when following the protocol, when all the other players become adversarial. The validators have never negative utility, thus it is enough to prove that neither the other users do. In the worst case scenario for user $u \in U$ a wrong process is validated. To do so, another user $u' \in U$ should be publish it and the validators should approve it. Under the assumption that there at most $\lfloor \frac{m_v}{3} \rfloor$ corrupted processes, in [50] it is proved that this is not possible. The proof follows from the intuition that the Byzantine validators own less than a third of the network they cannot validate two different transactions including one which can damage the user u. Therefore users never get negative utility if there are at most $\lfloor \frac{m_v}{3} \rfloor$ Byzantine players. This corresponds to the definition of $\lfloor \frac{m_v}{3} \rfloor$-weak-immunity (cf. Definition 10). □

4.10 Cross-chain swap

In this section we analyze the protocol introduced in [47], that allows two users to swap assets belonging to two different blockchains, which do not communicate with each other. In [32] the authors introduce a theoretical framework proving that the protocol is correct for those players who are altruistic, no matter what the others do. In the following we prove that the Cross-chain Swap protocol [47] satisfies the (k, t)-weak-robustness.

In this protocol users publish two different transactions on two different blockchains (e.g., Altcoin and Bitcoin) that can be triggered with the disclosure of a single private key x. The transactions have to be published within two different time intervals, Δ_1 and Δ_2, depending on the corresponding blockchain. In [32] the relationship between Δ_1 and Δ_2 is provided for a generic cross-chain swap protocol. In the 2-players context of [47], the condition proved in [32] results in $\Delta_1 \geq 2\Delta_2$. Both works assume that the transactions can be published within the time interval $[0, \min(\Delta_1, \Delta_2)] = [0, \Delta_2]$.

More specifically, the protocol stands on the property of the *hash function*, introduced in Section 4.7. The hash function allows to map a string x to $y = hash(x)$ such that given y it is almost impossible to retrieve x. Briefly speaking, A creates a random string x, computes $y = hash(x)$, creates a transaction on the Bitcoin blockchain that sends an amount of bitcoins to B under the condition that B identifies z such that $y = hash(z)$. Then, B creates a transactions on the Altcoin blockchain that sends an amount of altcoins to A under the condition that A provides z such that $y = hash(z)$. A discloses x, thus validating both transactions.

A creates two transactions on the Bitcoin blockchain: TX1, that lets B receive an amount of bitcoins if she provides x, and TX2, that gives back the amount to A if B does not provide x within Δ_1 hours (in [47] $\Delta_1 = 48$). B creates two transactions on the Altcoin blockchain: TX3, that lets A receive an amount of altcoins if she provides x, and TX4, that gives back the amount to B if A does not provide x within Δ_2 hours (in [47] $\Delta_2 = 24$). The theoretical bounds for Δ_1 and Δ_2 are provided in [32]. In a context with two players, the condition is that $\Delta_1 \geq 2\Delta_2$. From now on we consider the assumption that Δ_1 and Δ_2 fulfill the properties set in [32], and specifically we have that $\min(\Delta_1, \Delta_2) = \Delta_2$.

Since the two blokchains are independent we model the protocol with two different mechanisms $(\mathscr{G}_1, \sigma_1)$ and $(\mathscr{G}_2, \sigma_2)$ (cf. Definitions 19 and 20), that represent the actions that the players perform in each blockchain. We set to 0 the utility of the initial state, 1 the utility of every state in which the player receive what is asked, -1 the utility of every state in which the player gives some coins without receiving any. The Bitcoin blockchain is represented by game $\mathscr{G}_1$, while the Altcoin blockchain by $\mathscr{G}_2$ (cf. Fig. 20). We work under the assumption that a transaction can be published within $\min(\Delta_1, \Delta_2) = \Delta_2$ hours.

Definition 19. The *Bitcoin game* is an extensive form game $\mathscr{G}_1$ with 2 players $N = \{A, B\}$ and 5 nodes (1 is the vertex):

1. A can either Y, pick a random string x, create TX1 and TX2, then send TX2 to B, or doing none of them N. The action Y leads to node 2, while the action N leads to the outcome $(0, 0)$.

2. B can either Y, sign TX2, that leads to node 3, or N refusing to do it, with outcome $(0, 0)$.

3. A can either do nothing N, with thus outcome $(0, 0)$, or Y publish TX1 on the Bitcoin blockchain, that leads to node 4.

4. Both A and B have available two actions: either Y publish TX2 before that x is revealed or N not. If any of the two does so, the outcome is $(0, 0)$. Otherwise, A reveals x and (N, N) leads to node 5.

5. B can either Y publish x on the Bitcoin blokchain or N not doing it. If she does, the outcome is $(1, 1)$. If she does not, the outcome is $(1, -1)$.

The strategy profile that corresponds to following the protocol is $\sigma_1 = (\{Y, Y, N\}, \{Y, N, Y\})$, respectively played at nodes $(\{1, 3, 4\}, \{2, 4, 5\})$. Until x is revealed, the transactions cannot be triggered, therefore they provide null payoff. When x is revealed on the other chain, A has received the altcoins (thus with payoff equal to 1). If at step 5 B reveals x, she triggers the contract and receives the bitcoins (payoff equal to 1). Otherwise she has lost her asset in altcoins (negative payoff -1).

Definition 20. The *Altcoin game* is an extensive form game $\mathscr{G}_2$ with 2 players $N = \{A, B\}$ and 5 nodes (1 is the vertex):

1. B can either Y, create TX3 and TX4 and send the latter to A, or doing nothing N. The action Y leads to node 2, while the action N leads to the outcome $(0, 0)$.
2. A can either Y, sign TX4, that leads to node 3, or N refusing to do it, with outcome $(0, 0)$.
3. B can either do nothing N, with thus outcome $(0, 0)$, or publish TX3 on the Altcoin blockchain (Y), that leads to node 4.
4. Both A and B have available two actions: either publish TX4 (Y) before that x is revealed or not (N). If any of the two does so, the outcome is $(0, 0)$. Otherwise, A reveals x and (N, N) leads to node 5.
5. A can either publish x on the Altcoin blockhain (Y) or not doing it (N). If she does, the outcome is $(1, 0)$. If she does not, the outcome is $(0, 0)$.

The strategy profile that corresponds to following the protocol is $\sigma_2 = (\{Y, N, Y\}, \{Y, Y, N\})$, respectively played at nodes $(\{2, 4, 5\}, \{1, 3, 4\})$. Until x is revealed, the transactions cannot be triggered, therefore they provide null payoff. When x is revealed, A receives the altcoins (thus with payoff equal to 1). B does not know if he receives the asset, hence her payoff is 0.

Since the two blockchains are independent, we consider the composition of the two games $(\mathscr{G}_1 \odot \mathscr{G}_2, \{\sigma_{1i}, \sigma_{2i}\})$ that represents the full protocol and analyze its properties.

Theorem 22. *Under the assumption that any transaction can be published within a time interval* $[0, \Delta_2]$*, the mechanism* $(\mathscr{G}_1 \odot \mathscr{G}_2, \{\sigma_{1i}, \sigma_{2i}\})$ *is not immune.*

Proof. The strategy profile $\{\sigma_{1i}, \sigma_{2i}\}$ provides outcome

$$u_{\mathscr{G}_1 \odot \mathscr{G}_2}(\{\sigma_{1i}, \sigma_{2i}\}) = u_{\mathscr{G}_1}(\sigma_1) + u_{\mathscr{G}_2}(\sigma_2) = (1, 1) + (1, 0) = (2, 1)$$

If B considers a strategy σ_B^* that lets her play action N at node 2 of the Bitcoin game and action N at node 1 of the Altcoin game, the outcome is

$$u_{\mathscr{G}_1 \odot \mathscr{G}_2}(\{\sigma_{1A}, \sigma_{2A}\}, u_B^*) = u_{\mathscr{G}_1}(\sigma_{1A}, \sigma_{1B}^*) + u_{\mathscr{G}_2}(\sigma_{2A}, \sigma_{2B}^*) = (0, 0) + (0, 0) = (0, 0)$$

thus reducing the payoff for player A. In a two-player game a mechanism is immune if it is 1-immune (cf. Definition 9), but in this case A receives a loss if B performs a specific Byzantin behaviour. □

Theorem 23. *Under the assumption that any transaction can be published within an interval of time* Δ_2*, the mechanism* $(\mathscr{G}_1 \odot \mathscr{G}_2, \{\sigma_{1i}, \sigma_{2i}\})$ *is optimal resilient and weak immune.*

Proof. It is enough to prove that the two mechanisms $(\mathscr{G}_1, \sigma_1)$ e $(\mathscr{G}_2, \sigma_2)$ satisfy the properties and then exploit the properties of the operator composition of games.
In game $\mathscr{G}_1$ the strategy profile σ_1 is the only one with outcome $(1, 1)$, which is maximal. Thus we have that $(\mathscr{G}_1, \sigma_1)$ is strongly resilient.
Every strategy different from σ_1 is weakly dominated, indeed they bring to either outcome -1 or 0, which is lower than $u_1(\sigma_1) = (1, 1)$. Thus σ_1 is a stable Nash equilibrium and for Proposition 2 we have that the mechanism $(\mathscr{G}_1, \sigma_1)$ is practical.
In order to prove weak immunity we apply Proposition 4. When following respectively strategies σ_{1A} and σ_{1B} both A and B never get negative utility. Therefore the mechanism $(\mathscr{G}_1, \sigma_1)$ is also weak immune.
In game $\mathscr{G}_2$ the strategy profile σ_2 produces an outcome $(1, 0)$ which is maximal for both players, thus we have that the mechanism $(\mathscr{G}_2, \sigma_2)$ is strongly resilient.
The strategies within σ_2 are never weakly dominated, because none of the others can provide a

better outcome. Hence the mechanism is practical.
Every outcome is non-negative, therefore the mechanism is weak immune.

Since both mechanisms are optimal resilient and weak immune, we can apply Theorems 2, 3 and 4, that ensure the invariance of the properties once the operator composition is applied. The mechanism $(\mathscr{G}_1 \odot \mathscr{G}_2, \{\sigma_{1i}, \sigma_{2i}\})$ is thus optimal resilient and weak immune. □

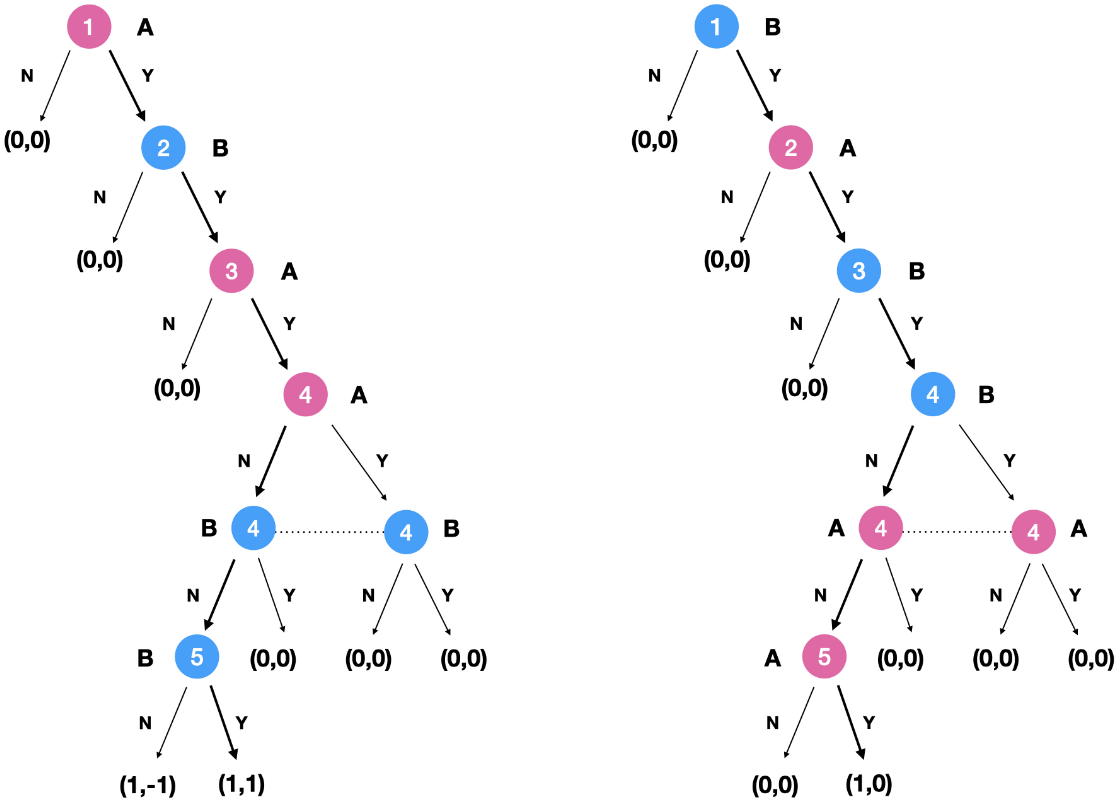


Figure 20: The game trees of $\mathscr{G}_1$ and $\mathscr{G}_2$.

The mechanism is not immune, indeed it is sufficient that one player does not create or publish a transaction to stop the protocol. Under the assumption that any transaction can be published within a time interval $[0, \Delta_2]$ the mechanism is optimal resilient and weak immune.

5 Conclusions

We propose the first generic game theoretical framework that models the robustness of blockchains towards rational and byzantine behaviors. We identify the necessary and sufficient conditions for a protocol to be robust and develop a methodology to characterize the robustness of complex protocols via the composition of simpler robust building blocks. The effectiveness of our framework is demonstrated by its capability to capture the robustness of various blockchain protocols such as Bitcoin, Tendermint, lightning networks, side-chain and cross-chain protocols. As future work we plan to investigate the resilience of other blockchain protocols such as Algorand [20] or IOTA [52].

References

[1] Abraham, I., Alvisi, L., and Halpern, J. Distributed computing meets game theory: Combining insights from two fields. *SIGACT News 42* (06 2011), 69–76.

[2] Abraham, I., Dolev, D., Gonen, R., and Halpern, J. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing* (New York, NY, USA, 2006), PODC '06, Association for Computing Machinery, pp. 53–62.

[3] Abraham, I., Dolev, D., and Halpern, J. Y. Distributed protocols for leader election: A game-theoretic perspective. *ACM Trans. Econ. Comput. 7*, 1 (Feb. 2019).

[4] Aiyer, A. S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.-P., and Porth, C. Bar fault tolerance for cooperative services. In *SOSP '05* (2005).

[5] Amoussou-Guenou, Y., Biais, B., Potop-Butucaru, M., and Tucci Piergiovanni, S. Rational vs byzantine players in consensus-based blockchains. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020* (2020), A. E. F. Seghrouchni, G. Sukthankar, B. An, and N. Yorke-Smith, Eds., International Foundation for Autonomous Agents and Multiagent Systems, pp. 43–51.

[6] Amoussou-Guenou, Y., Del Pozzo, A., Potop-Butucaru, M., and Tucci-Piergiovanni, S. Dissecting tendermint. In *Networked Systems* (Cham, 2019), M. F. Atig and A. A. Schwarzmann, Eds., Springer International Publishing, pp. 166–182.

[7] Andoulaki, E., Jarkeand, M., and Quisquater, J.-J. Introduction to the special theme: Blockchain engineering. *ERCIM NEWS*, 110 (2017), 6,7.

[8] Avarikioti, G., Kogias, E. K., and Wattenhofer, R. Brick: Asynchronous state channels. *arXiv preprint arXiv:1905.11360* (2019).

[9] Avarikioti, G., Laufenberg, F., Sliwinski, J., Wang, Y., and Wattenhofer, R. Towards secure and efficient payment channels. *arXiv preprint arXiv:1811.12740* (2018).

[10] Badertscher, C., Garay, J., Maurer, U., Tschudi, D., and Zikas, V. But why does it work? a rational protocol design treatment of bitcoin. In *Annual international conference on the theory and applications of cryptographic techniques* (2018), Springer, pp. 34–65.

[11] Bellman, R. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America 42*, 10 (1956), 767.

[12] Belotti, M., Božić, N., Pujolle, G., and Secci, S. A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys & Tutorials 21*, 4 (2019), 3796–3838.

[13] Belotti, M., Moretti, S., Potop-Butucaru, M., and Secci, S. Game theoretical analysis of Atomic Cross-Chain Swaps. In *40th IEEE International Conference on Distributed Computing Systems (ICDCS)* (Singapore, Singapore, Dec. 2020).

[14] Belotti, M., Moretti, S., and Zappalà, P. Rewarding miners: bankruptcy situations and pooling strategies. In *17th European Conference on Multi-Agent Systems (EUMAS)* (Tessaloniki, Greece, July 2020).

[15] Bentov, I., Hubácek, P., Moran, T., and Nadler, A. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. *IACR Cryptology ePrint Archive 2017* (2017), 300.

[16] Bernheim, B., Peleg, B., and Whinston, M. D. Coalition-proof nash equilibria i. concepts. *Journal of Economic Theory 42*, 1 (1987), 1 – 12.

[17] Borkowski, M., McDonald, D., Ritzer, C., and Schulte, S. Towards atomic cross-chain token transfers: State of the art and open questions within tast. *Distributed Systems Group TU Wien (Technische Universit at Wien), Report* (2018).

[18] Brenguier, R. Robust equilibria in mean-payoff games. In *Foundations of Software Science and Computation Structures* (Berlin, Heidelberg, 2016), B. Jacobs and C. Löding, Eds., Springer Berlin Heidelberg, pp. 217–233.

[19] Cachin, C., and Vukolić, M. Blockchain consensus protocols in the wild. *arXiv preprint arXiv:1707.01873* (2017).

[20] Chen, J., and Micali, S. Algorand: A secure and efficient distributed ledger. *Theor. Comput. Sci. 777* (2019), 155–183.

[21] Chen, Z., Li, B., Shan, X., Sun, X., and Zhang, J. Discouraging pool block withholding attacks in bitcoins, 2020.

[22] Chinchuluun, A., Pardalos, P., Migdalas, A., and Pitsoulis, L. *Pareto Optimality, Game Theory And Equilibria*, vol. 17. 01 2008.

[23] Ewerhart, C. Finite blockchain games. *University of Zurich, Department of Economics, Working Paper*, 355 (2020).

[24] Eyal, I. The miner's dilemma. pp. 89–103.

[25] Eyal, I., and Sirer, E. G. Majority is not enough: Bitcoin mining is vulnerable. In *International conference on financial cryptography and data security* (2014), Springer, pp. 436–454.

[26] Garay, J., Katz, J., Maurer, U., Tackmann, B., and Zikas, V. Rational protocol design: Cryptography against incentive-driven adversaries. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (2013), IEEE, pp. 648–657.

[27] Garay, J., and Kiayias, A. Sok: A consensus taxonomy in the blockchain era. In *Cryptographers' Track at the RSA Conference* (2020), Springer, pp. 284–318.

[28] Garay, J., Kiayias, A., and Leonardos, N. The bitcoin backbone protocol: Analysis and applications. In *Advances in Cryptology - EUROCRYPT 2015* (Berlin, Heidelberg, 2015), E. Oswald and M. Fischlin, Eds., Springer Berlin Heidelberg, pp. 281–310.

[29] Gramoli, V. From blockchain consensus back to byzantine consensus. *Future Generation Computer Systems* (2017).

[30] Gudgeon, L., Moreno-Sanchez, P., Roos, S., McCorry, P., and Gervais, A. Sok: Off the chain transactions. *IACR Cryptology ePrint Archive 2019* (2019), 360.

[31] Hammond, P. *Utility Invariance in Non-Cooperative Games*, vol. 38. 06 2006, pp. 31–50.

[32] Herlihy, M. Atomic cross-chain swaps. In *Proceedings of the 2018 ACM symposium on principles of distributed computing* (2018), pp. 245–254.

[33] Hillas, J. On the definition of the strategic stability of equilibria. *Econometrica 58*, 6 (1990), 1365–1390.

[34] Inc., O. F. The on-line encyclopedia of integer sequences, year = 2021, url = https://oeis.org/A178682, urldate = 2021-01-29.

[35] Kiayias, A., Koutsoupias, E., Kyropoulou, M., and Tselekounis, Y. Blockchain mining games. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (2016), pp. 365–382.

[36] Kiayias, A., and Stouka, A.-P. Coalition-safe equilibria with virtual payoffs. *arXiv preprint arXiv:2001.00047* (2019).

[37] Kohlberg, E., and Mertens, J.-F. On the strategic stability of equilibria. *Econometrica: Journal of the Econometric Society* (1986), 1003–1037.

[38] Koutsoupias, E., and Papadimitriou, C. Worst-case equilibria. In *STACS 99* (Berlin, Heidelberg, 1999), C. Meinel and S. Tison, Eds., Springer Berlin Heidelberg, pp. 404–413.

[39] Kuhn, H. W., and Tucker, A. W. *Contributions to the Theory of Games*, vol. 2. Princeton University Press, 1953.

[40] Kwon, J. Tendermint: Consensus without mining. *Draft v. 0.6, fall 1*, 11 (2014).

[41] Liu, Z., Luong, N. C., Wang, W., Niyato, D., Wang, P., Liang, Y., and Kim, D. I. A survey on blockchain: A game theoretical perspective. *IEEE Access 7* (2019), 47615–47643.

[42] Mailath, G. J. Do people play nash equilibrium? lessons from evolutionary game theory. *Journal of Economic Literature 36*, 3 (1998), 1347–1374.

[43] Moscibroda, T., Schmid, S., and Wattenhofer, R. When selfish meets evil: Byzantine players in a virus inoculation game. vol. 2006, pp. 35–44.

[44] Nakamoto, S. A peer-to-peer electronic cash system.

[45] Nash, J. F. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences 36*, 1 (1950), 48–49.

[46] Natoli, C., Yu, J., Gramoli, V., and Esteves-Verissimo, P. Deconstructing blockchains: A comprehensive survey on consensus, membership and structure. *arXiv preprint arXiv:1908.08316* (2019).

[47] Nolan, T. Re: Alt chains and atomic transfers. accessed on January 10, 2020. `https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949`.

[48] Osborne, M. J., and Rubinstein, A. *A course in game theory*. MIT press, 1994.

[49] Pass, R., and Shi, E. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (2017), pp. 315–324.

[50] Pedrosa, A. R., and Gramoli, V. Platypus: Offchain protocol without synchrony. In *18th IEEE International Symposium on Network Computing and Applications, NCA 2019, Cambridge, MA, USA, September 26-28, 2019* (2019), A. Gkoulalas-Divanis, M. Marchetti, and D. R. Avresky, Eds., IEEE, pp. 1–8.

[51] Poon, J., and Dryja, T. The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[52] Popov, S., Saa, O., and Finardi, P. Equilibria in the tangle. *Comput. Ind. Eng. 136* (2019), 160–172.

[53] Roughgarden, T. *Selfish Routing and The Price of Anarchy*, vol. 74. 01 2005.

[54] Roughgarden, T., and Tardos, E. Introduction to the inefficiency of equilibria. *Algorithmic Game Theory 17* (2007), 443–459.

[55] Schrijvers, O., et al. Incentive compatibility of bitcoin mining pool reward functions. In *International Conference on Financial Cryptography and Data Security* (2016), Springer, pp. 477–498.

[56] Tsabary, I., and Eyal, I. The gap game. In *Proceedings of the 2018 ACM SIGSAC conference on Computer and Communications Security* (2018), pp. 713–728.

[57] Tschorsch, F., and Scheuermann, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials 18*, 3 (2016), 2084–2123.

[58] Von Neumann, J., Morgenstern, O., and Kuhn, H. W. *Theory of games and economic behavior (commemorative edition).* Princeton university press, 2007.

[59] Wang, W., Hoang, D. T., Xiong, Z., Niyato, D., Wang, P., Hu, P., and Wen, Y. A survey on consensus mechanisms and mining management in blockchain networks. *arXiv preprint arXiv:1805.02707* (2018), 1–33.

[60] Zheng, Z., Xie, S., Dai, H.-N., Chen, X., and Wang, H. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services 14*, 4 (2018), 352–375.