Game Theoretical Analysis of Atomic Cross-Chain Swaps

Marianna Belotti, Stefano Moretti, Maria Potop-Butucaru, Stefano Secci

Game Theoretical Analysis of Cross-Chain Swaps

Marianna Belotti
Groupe Caisse des Dépôts, Cnam
Paris, France
marianna.belotti@caissedesdepots.fr

Stefano Moretti
Univ. Paris Dauphine, PSL
CNRS LAMSADE, Paris, France
stefano.moretti@dauphine.fr

Maria Potop-Butucaru
Sorbonne Université
CNRS LIP6, Paris, France
maria.potop-butucaru@lip6.fr

Stefano Secci
Cnam
Paris, France
stefano.secci@cnam.fr

***Abstract*—In this paper we address the distributed cross-chain swap problem in the blockchain context where multiple agents exchange assets across multiple blockchain systems (e.g. trading Bitcoins for Litecoins or Ethers). We present a mathematical framework allowing to characterize blockchain swap protocols as the combination of a publishing and a commitment phase, where contracts are respectively published and then committed. We characterize the equilibria of existing cross-chain swap protocols (i.e., blockchain swap protocols exchanging assets among different blockchains). More precisely, we prove that following a swap protocol characterized by concurrent publishing of exchange contracts and snap (immediate) assets transfers is a Nash equilibrium. Furthermore, we prove that for protocols with a sequential publishing and commitment of the assets transfers, following the prescribed protocol is a sub-game perfect equilibrium.**

I. Introduction

The modern economy is moving to a new era where economical transactions use cryptocurrencies instead of fiat money. Crypto-currencies (i.e., Bitcoin, Ethereum, etc.) are based on the use of blockchains, which are basically transactional systems governed by decentralized protocols. Blockchains pave the way for a new approach to organize and sustainably maintain long-term transactions as well as high-level services. It is interesting to note that the number of blockchains that currently hold the head of newspapers has gone from one in 2008 (the famous Bitcoin blockchain [1]) to a few tens in 2018 such as Hyperledger [2], Ethereum [3], Zcash [4], Corda [5], Ripple [6], Tendermint [7] etc. Each of these systems has its own modus operandi, its own governance and even its own way of agreeing on a common history. Each system has its own advantages that make it attractive for various applications and geopolitical contexts. We are witnessing the creation of several ecosystems, each with its own currency and governance.

Similar to modern international economical exchanges which are based on different government-issued currencies, inter-blockchain exchanges must be based on common rules resilient to attacks, failures or malicious behaviors affecting the network. There are currently several operational systems for achieving interoperability between different blockchains such as Kybernetwork [8], Aion [9], Cosmos [10] or Polkadot [11]. These systems can be classified into two categories according to their decentralization level: systems that use a trusted third-party to validate transactions or systems that realize it directly between blockchains without the need of a trusted third-party. In order to execute an exchange or a *swap* (i.e., a set of transactions between parties), transacting agents (i.e., blockchain users) are provided with a protocol to stick to. A protocol in this case consists of a specific sequence of instructions agents should perform to preserve the ACID properties [12] of the individual transactions or exchanges; that is, *Atomicity*: the all-or-nothing occurs and each participant must know which state he or she is in; *Consistency*: each successful transaction by definition commits only valid results; *Isolation*: transactions run independently; *Durability*: transactions cannot be abrogated after commitment.

Atomic cross-chain swaps fall into the class of the so called TAST (i.e., Token Atomic Swap Technology) [13] research ideas aiming at making blockchains interoperable. Differently from decentralized exchanges – initiatives recently emerged to remove the need of trust on traditional exchange services (e.g., the format proposed by the Ethereum DEX protocols [14; 15] atomically swapping ERC-20 tokens on the Ethereum blockchain) – atomic swaps have not limits in operating cross-chain. The very first atomic swap solution has been proposed for Bitcoin by *Nolan* [16] making use of hash-time locked contracts enabling conditional assets transfers. Nowadays few platforms actually support cross-chain exchanges that at this stage are still slow and inefficient. *Decred* [17] implements Nolan's logic on UTXO-based premissionless blockchains such as Bitcoin Cash [18; 19; 20], Litecoin [21], Qtum [22], etc. *BartherDEX* [23], part of the Komodo project [24], represents a cross-chain solution that matches orders and defines the swap protocol. *Blockchain.io* [25] implements atomic cross-chain swaps by combining centralized components (order matching) with decentralized ones (trade settlement and execution). Therefore, research now focuses on *hybrid* swap protocols, replacing decentralized commitment/locking schemes (hash-locks) with centralized ones, resulting more attractive and efficient. *AC3TW* and *AC3WN* [26] protocols propose atomic cross-chain swaps respectively with centralized and "decentralized" trusted authorities (i.e., an external agent and an external blockchain) acting as witnesses. According to *Arwen* protocol [27] crypto-assets are swapped through centralized exchange services that in no way acquire the assets custody. *XClaim* [28] overcomes blockchain data-structures incompatibility by swapping cryptocurrency-backed assets.

It should be noted that different swap protocols differ essentially in the involved parties. The set of swap participants can be composed only of the asset owners (e.g., as in [29])

or by owners accompanied by a trusted third party (e.g., as in the AC3TW protocol [26]).

To the best of our knowledge, there is (i) no formal analysis on the structure and the properties blockchain swap protocols satisfy and, (ii) no game theoretical modeling of participants strategic interactions (i.e., to follow or not to follow the prescribed protocol). More precisely, *Nolan* [16] presents the Bitcoin swap protocol in a functional manner. In [29] the author provides a partial game theoretical analysis (specific to the protocol) presenting no structural characterization of the possible equilibria of the system (see Section III-A). Authors in [26] analyze the atomicity violations characterizing the protocol in [16]. However, no game theoretical result is provided.

According to recent studies [30; 31] proposed swap protocols are not properly analyzed neither from the structural point of view, responsible for their atomicity, nor from the strategical point of view making them satisfying *liveness* and *safety* properties (i.e., the protocol terminates in a valid state). More precisely, it is important to analyze the behavior of the swap participants that are rational agents actively participating in the exchange by following or not the prescribed protocol according to their own objective function.

A. Our contribution

We propose a generic game theoretical framework that formalizes the swap problem and characterizes blockchain swap protocols by clearly separating the contracts publishing phase and their commitment phase. Furthermore, we prove that (i) following a swap protocol characterized by an *effective* decision function (when players have the power to accept or decline the desired assets) is a subgame perfect equilibrium in dominant strategies and, that (ii) following a swap protocol characterized by concurrent publishing and *snap* (immediate) commitment is a Nash equilibrium. Our generic framework allows us to characterize equilibria of two representative recent protocols presented in [16] and [26] respectively. In the case of the protocol proposed in [16] and generalised in [29], following the protocol is the unique subgame perfect equilibrium (in dominant strategies), while in the case of the protocol proposed in [26], following the protocol is a Nash equilibrium.

II. Swap Games

In this section we propose a formal definition of the *swap* problem and a formalization of the corresponding *blockchain swap protocol* that can be atomic or not. The latter consists of two different phases (i.e., publishing and commitment of transfers) and represents a particular strategy of a *swap game* that players can choose to adopt or not.

A. Swap problem and swap protocol

In a general *swap problem*, swapping parties aim at exchanging assets among themselves. A *swap protocol* defines the set of asset transfers and the order in which they should be executed. Given a set of assets and the corresponding owners, a *swap* consists of an asset ownership exchange within the set of owners.

Definition 1 (swap problem)**.** *A swap problem is defined as a tuple* $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}} \rangle$ *where:*

- $\mathcal{A} = \{1, \dots, m\}$ *is the set of assets to be swapped;*
- $\mathcal{O} = \{1, \dots, n\}$ *is the set of owners or agents participating in the exchange. We consider in the following that* $n \leq m$ *since each owner owns at least one asset;*
- $b_0, b_* : \mathcal{A} \to \mathcal{O}$ *are the original and the desired ownership maps – both surjective – such that* $\forall a \in \mathcal{A},\ b_0(a) \neq b_*(a)$ *and;*
- u_i *is the payoff function for owner* $i \in \mathcal{O}$ *over bundles of assets in* $2^{\mathcal{A}}$ *such that* $u_i(b_0^{-1}(i)) < u_i(b_*^{-1}(i))$ *(i.e., each owner* i *strictly prefers the desired bundle of assets to her original bundle) and* $\forall S, T \in 2^{\mathcal{A}} : S \subseteq T$, $u_i(T) \geq u_i(S)$ *(i.e., a larger bundle is strictly preferred to a smaller one), for each* $i \in \mathcal{O}$*.*

The representation with two different surjective functions b_0, b_* describes swaps as ownership exchanges problems. Agents participate in a swap with the asset(s) they aim to exchange for others, preferring the desired new asset(s) to finding themselves as in the initial configuration. Moreover, we assume owners' payoff function increases monotonically with the size of the asset bundle.

The transition from an initial configuration to a post-swap configuration is defined by the corresponding *swap protocol* consisting in a sequence of operations to be executed in a certain order. In centralized swap protocols a central role in the swap is played by a *trusted third party*. That is, asset ownership is transferred first to the trusted third party that in turn transfers assets back to the new owners. On the other hand, decentralized swap protocols contemplate ownership transfers within asset owners only; the latter agrees on a particular swap configuration (i.e., the assets to exchange) without trusting each other.

In order to formally define a swap protocol we need to introduce first, the structure of a *decentralized exchange protocol*, consisting of a sequence of asset(s) transfers to be committed.

Definition 2 (decentralized exchange protocol)**.** *Let* $\sigma = \{(A^k, O^k, X^k) : |A^k| \geq |O^k|\}_k,\ k \in \{1, \dots, t\}, t \in \mathbb{N} : t \leq m$ *be a sequence of exchanges where,*

- $A^k \subseteq \mathcal{A}$ *specifies the subset of assets involved in the exchange at step* k*;*
- $O^k \subseteq \mathcal{O}$ *specifies the subset of owners involved in the exchange at step* k*;*
- $X^k : A^k \to O^k$ *(surjective) specifies the owner* $X^k(a) \in O^k$ *of any asset* $a \in A^K$ *at step* k*;*

A sequence σ *defines a decentralized exchange protocol that engenders a sequence of maps* $b_1^\sigma, b_2^\sigma, \dots, b_t^\sigma : \mathcal{A} \to \mathcal{O}$ *such that for all* $k \in \{1, 2, \dots, t\}$*:*

- $b_k^\sigma(z) = b_{k-1}^\sigma(z),\ \forall z \in \mathcal{A} \setminus A^k$*;*
- $b_k^\sigma(z) = X^k(z),\ \forall z \in A^k$*,*

where we set $b_0^\sigma = b_0$*.*

So, the triple (A^k, O^k, X^k) specifies that the asset set $A^k \subseteq \mathcal{A}$ is transferred at step k to owners $O^k \subseteq \mathcal{O}$ according to X^k. Note that, b^σ_k and b^σ_{k-1} differ only for the ownership of assets A^k belonging to $b^\sigma_{k-1}(A^k)$ at step $k-1$ and to O^k at step k.

Example 1. *Let us consider the swap problem* $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}}\rangle$ *such that the set of assets is* $\mathcal{A} = \{a, b, c, d, e\}$*, the set of owners is* $\mathcal{O} = \{1, 2, 3\}$*, the original ownership map is* $b_0 = (1, 1, 2, 3, 3)$ *(meaning that the asset ownership is:* 1 *for assets* a *and* b*,* 2 *for* c *and* 3 *for* d *and* e*) and the desired ownership map is* $b_* = (2, 3, 1, 2, 1)$.

One may consider a sequence of exchanges involving first owners 1 *and* 2*, then* 1 *and* 3 *and finally* 2 *and* 3*. Precisely,*

$$\sigma = (\{a, c\}, \{1, 2\}, \{X^1(a) = 2, X^1(c) = 1\}),$$

$$(\{b, e\}, \{1, 3\}, \{X^2(b) = 3, X^2(e) = 1\}),$$

$$(\{d\}, \{2\}, \{X^3(d) = 2\}).$$

Sequence σ *engenders a sequence of ownership maps such that* $b_0 = (1, 1, 2, 3, 3)$*,* $b_1 = (2, 1, 1, 3, 3)$*,* $b_2 = (2, 3, 1, 3, 1)$*,* $b_3 = (2, 3, 1, 2, 1) = b_*$.

The sequence σ defines, for each protocol step k, (i) the assets A^k whose property is to be transferred, (ii) the new assets' owners O^k and, (iii) the function X^k assigning the precise owner to each asset. The protocol is decentralized as ownership transfers take place among the owners themselves. At each step k an asset can change owner, the final configuration at time t provides the final asset owner.

In the simple case where agents agree on exchanging assets through a single ownership transfer at each step k, the corresponding *single-swap protocol* is represented by the sequence $\sigma = \{(a^k, o^k) : \ o^k \in \mathcal{O}, a^k \in \mathcal{A}\}_{k\in\{1,\dots,t\}}, t \in \mathbb{N} : t \leq m$.

Protocol σ describes a general asset exchange agreement, and not necessarily a swap where the transfer of an asset ownership exists only if associated with the transfer of another one. More precisely, swap participants are interested in the final (at time t) and not temporary (intermediate at step $k \neq t$) acquisition of one or more assets owned by other agents. In a swap protocol, each asset changes owner only once.

Definition 3 (decentralized swap protocol)**.** *A decentralized swap protocol is defined as a decentralized exchange protocol where the set* $\{A^k : k = 1, \dots, t, t \in \mathbb{N} : t \leq m\}$ *is a partition of the asset set* $\mathcal{A}$.

Example 1 provides a decentralized swap protocol σ since $\{\{a, c\}, \{b, e\}, \{d\}\}$ is a partition of $\mathcal{A}$.

From the formalization introduced in Definition 2 it is possible to derive a graphic representation, by means of a digraph $\mathcal{D} = (V, E)$, as the one proposed by *Herlihy* in [29]. More precisely, vertexes are asset owners $V = \mathcal{O}$ and edges $E \ni e = (e_i, e_o) : (e_i, e_o) \in V^2 \wedge e_i \neq e_o$ can be derived by the original ownership map b_0 (providing e_i) and the desired one b_* (providing e_o). Fig. 1 presents the digraph of Example 1.

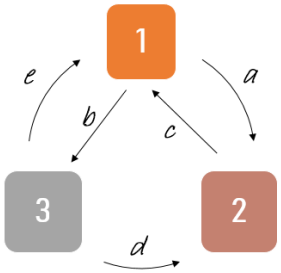

Fig. 1. Digraph swap protocol representation of the sequence $\sigma = (\{a, c\}, \{1, 2\}, \{X^1(a) = 2, X^1(c) = 1\}), (\{b, e\}, \{1, 3\}, \{X^2(b) = 3, X^2(e) = 1\}), (\{d\}, \{2\}, \{X^3(d) = 2\})$.

B. Atomic swap protocol

Whenever swapping parties do not trust each other it is in their interest to ensure that no participant can take advantage from the swap agreed on. The protocol must be constructed in such a way that the swap is performed in its entirety or no asset transfer is committed (i.e., all-or-nothing). In the case of failures during the protocol execution, every swap participant must be able to regain possession of the original owned assets.

Definition 4 (efficient decentralized swap protocol)**.** *A decentralized swap protocol* σ *is said to be efficient if the engendered sequence* $b^\sigma_1, b^\sigma_2, \dots, b^\sigma_t$ *is such that* $b^\sigma_t = b_*$.

Definition 5 (atomic decentralized swap protocol)**.** *A decentralized swap protocol* σ *is atomic whenever it is efficient (in the sense of Definition 4) or* $b^\sigma_t = b_0$.

In order to prevent participants to externally exchange the assets involved in the swap, the protocol requires assets to be *locked* in specific transactions (i.e., they cannot be the object of other transfers). Once locked, the transfer commitment allows every participant to redeem the new swapped asset(s). Moreover, due to the atomicity requirement:

(i) any commitment should be conditioned on the correct asset locking i.e., transfers are committed only when all assets of the swap are correctly locked;
(ii) consequently to failures in the assets locking, the initial situation must be restored;
(iii) once an asset transfer is committed all the other transfers have to be committed, too.

When considering blockchains – decentralized trustless environments – swap protocols need to be *atomic* according to Definition 5. Conditional transfers in blockchain systems are implemented with *distributed contracts* i.e., scripts executed on blockchain nodes that can enforce and regulate relationships among network actors. A widely adopted commitment scheme is the crypto-primitive *hash-lock* [32] i.e., asset ownership is locked with a hash value h that is the outcome of a one-way function H with a secret s as input (i.e., $h = H(s)$) and can be unlocked only when s is revealed. Transfers are conditioned on the hash constraint guaranteeing that assets to be swapped have been properly locked. Whenever transfers are not committed, due to atomicity, the initial ownership configuration needs to be restored; this can

be implemented with *time-locks* [33] i.e., a contract primitive restricting the asset transfer until a specified future time. Hash-locks and time-locks properly combined enable conditional asset transfers necessary to satisfy requirements (i) and (ii) of a swap protocol. Hence, when considering a blockchain swap, an asset transfer consists of an atomic distributed contract that, according to certain protocols [16; 29], coincides with a *distributed hash-locked contract* (HTLC).

The latter needs to be correctly implemented, published and validated on the network (i.e., part of the valid transaction history) before being committed. Concerning requirement (iii), a mechanism that forces committing all transfers is needed in order to avoid possible atomicity violations (see Section III-A3). However, having a forcing scheme comes at a price, i.e., loss of decentralization (see Section III-B).

C. Atomic blockchain swap protocol

In this work we consider blockchain swap protocols characterized by two distinct phases: a first phase in which transfers (i.e., hash-locked contracts) are published and a second one where they are committed. To achieve atomicity, the order in which operations in the two phases are executed matters (see Section III). The commitment phase has to be conditioned by the execution of the publishing phase; all the asset transfers have to be published before being committed. Therefore, a blockchain swap protocol is characterized by a publishing protocol σ_P followed by a commitment protocol σ_T. While the first protocol (σ_P) is a simple sequence of transfers, the second one (σ_T) is a decentralized swap protocol according to Definition 3.

In order to analyze the participants' strategic behaviors the protocol needs to specify the schedules in which agents perform operations. Therefore, given a sequence of operation σ, a set of asset owners $\mathcal{O}$ and a centralized trusted authority τ, we define a *decision function* $F : \{1, \dots, t\} \to \mathcal{O} \cup \{\tau\}$ as a map that specifies the agent(s) $F(k)$ responsible for the publication (or commitment) of the transfer (A^k, O^k). Note that the agent(s) called to decide on the transfer can be either asset owners (i.e., $F(k) \in \mathcal{O}$) or an external trusted actor (i.e., $F(k) = \tau$).

Definition 6 (decentralized blockchain swap protocol)**.** *A decentralized blockchain swap protocol or simply a blockchain swap protocol is defined by the pair (σ_P, σ_T) where*

- $\sigma_P = \{(A^j, O^j)\}_{j\in\{1,\dots,t_P\}}$, $t_P \in \mathbb{N} : t_P \leq m$, $A^j \subseteq \mathcal{A}$ *and* $O^j \subseteq \mathcal{O}$ *is a sequence such that* $\forall j \in \{1, \dots, t_P\}$, $O^j = \{o \in \mathcal{O} : o \in b_*(A^j) \vee o \in b_0(A^j)\}$ *and,*
- $\sigma_T = \{(A^k, O^k, X^k)\}_{k\in\{1,\dots,t_T\}}$ *is a swap protocol engendering the sequence of maps $b_1^{\sigma_T}, \dots, b_{t_T}^{\sigma_T} : \mathcal{A} \to \mathcal{O}$ according to Definition 3.*

We associate to each sequence σ_P, σ_T the corresponding decision function F_P, F_T defined above.

Let us note that the publishing sequence σ_P is constructed in such a way that transfers that can be published are of type $(A^j, b_*(A^j))$ or $(A^j, b_0(A^j))$. That is, blockchain transactions can transfer the asset ownership to desired owners and original owners only.

Definition 7 (atomic blockchain swap protocol)**.** *An atomic blockchain swap protocol consists of a pair (σ_P, σ_T) where the engendered ownership map at time t of the commitment protocol coincides with the desired one; $b_{t_T}^{\sigma_T} = b_*$ or with initial one; $b_{t_T}^{\sigma_T} = b_0$.*

1) Phases separation: We have defined a blockchain swap protocol as a publishing sequence followed by a commitment emphasizing the precedence of the first phase over the second. However, it is necessary to condition the execution of the commitment protocol to the publication of all the contracts in order to have an atomic blockchain swap protocol. The following definition formalizes such a *commitment requirement*.

Definition 8 (commitment requirement)**.** *Given a blockchain swap protocol (σ_P, σ_T) whenever there exists an asset transfer (using the swap structure previously given) that is not correctly published, then no asset transfer is committed. Formally, if, in σ_P, $\exists \bar{j} \in \{1, \dots, t_P\} : O^{\bar{j}} \cap b_0(A^{\bar{j}}) \neq \emptyset$ then, in σ_T, $b_k^{\sigma_T} = b_0$ $\forall k \in \{1, \dots, t_T\}$.*

Every blockchain swap protocol has to meet the commitment requirement defined above in order to be atomic. Hence, Definition 8 is a necessary condition (but not sufficient, see Section III-A3) for atomicity. Indeed, whenever an asset transfer is not correctly published the time-lock acts by not modifying the original asset ownership, i.e., transferring the asset back to the original owner (see [34; 35] for more details).

Focusing separately on the commitment protocol, it is possible to derive initial properties on blockchain swaps.

2) Commitment protocol: As stated in the following proposition, whenever an asset transfer is not committed, the swap participant supposed to acquire the asset(s) ends up with less assets than expected while the original asset owner finds himself with an extra asset. Next proposition does not depend on the exact specification maps X^k, so we omit them in the sequence σ_T for the sake of simplicity.

Proposition 1. *Given a commitment sequence $\sigma_T = \{(A^k, O^k, X^k)\}_{k\in\{1,\dots,t_T\}}, t_T \in \mathbb{N} : t_T \leq m$ then, replacing O^k by $b_{k-1}^{\sigma_T}(A^k)$ in σ_T i.e., considering a new sequence $\sigma_T^k = (A^1, O^1), \dots, (A^{k-1}, O^{k-1}), (A^k, b_{k-1}^{\sigma_T}(A^k)), (A^{k+1}, O^{k+1}), \dots, (A^{t_T}, O^{t_T})$, for some $k \in \{1, \dots, t_T\}$, implies that:*

(i) $\left(b_{t_T}^{\sigma_T^k}\right)^{-1}(O^k) \subseteq \left(b_{t_T}^{\sigma_T}\right)^{-1}(O^k)$ *and,*

(ii) $\left(b_{t_T}^{\sigma_T^k}\right)^{-1}(b_{k-1}^{\sigma_T}(A^k)) \supseteq \left(b_{t_T}^{\sigma_T}\right)^{-1}(b_{k-1}^{\sigma_T}(A^k))$.

Proof. The claims follow from the fact that A^k is not given to O^k in σ_T^k but it is given back to the original owners. □

At time t, assets A^k do not necessarily belong to O^k thus, O^k could find themselves with no asset i.e., $\left(b_t^{\sigma_T^k}\right)^{-1}(O^k) = \emptyset$. Moreover, the original asset owners find themselves with both the original assets and the swapped ones.

Let us denote by σ_T^K, with $K \subseteq \{1, \ldots, t_T\}$ the sequence of pairs obtained by replacing (A^k, O^k) in σ_T by $(A^k, b_{k-1}^{\sigma_T}(A^k))$ for any $k \in K$ (i.e. by the owner(s) of A^k at step $k-1$).

3) Sequential phases and beyond: The formalization provided by Definition 6, enables capturing:

1- *Single-asset and multi-assets swaps* depending on the cardinality of the asset set A^k at step k in the commitment protocol.
2- Swap protocols with *sequential publishing and commitment*, both single-asset and multiple-assets, where ownership transfers are published and committed according to a precise temporal order. Every crypto-asset transfer has to be executed before or after another one. In Section III-A we show that the sequentiality of the publishing phase combined with the leader role of a swap participant guarantees that the commitment ownership is verified.
3- Swap protocols with *concurrent publishing* where transfers are no longer published according to a given time order, but may be concurrently created and propagated to the blockchain network. In fact, it is not essential that the various transfers are published one at a time but only that there is a clear distinction between the publication and the commitment phase. Therefore, the transfers publication can take place in a concurrential way and this is captured by the sequence σ_P where at step $j : j \in \{1, \ldots, t_P\}$ multiple assets transfers (A^j, O^j) are published.
4- Swap protocols with *snap commitment*, where assets transfers are all committed in the same time.

Proposition 2. *A blockchain swap protocol with a snap commitment scheme satisfying the commitment requirement as in Definition 8 is atomic.*

Proof. By contradiction, if $b_{t_T}^{\sigma_T} \neq b_0 \ \wedge \ b_{t_T}^{\sigma_T} \neq b_*$ then, since no problem in the publishing occurs, there should be a situation where some assets transfers are triggered and some others are not. However, this contradicts the snap commitment. □

4) Decision function: Let us focus on the decision function F previously defined. A blockchain swap can be completely driven by the asset owners only, in that case the function outcomes are all elements of $\mathcal{O}$. On the other hand, we can have the intervention of an external actor τ entrusted for committing assets transfers. More precisely, the external trusted actor cannot publish blockchain contracts in behalf of asset owners since blockchain swap protocols contemplate ownership transfers among owners only (as in Definiton 6). Considering the publishing phase we can notice that every asset owner is in charge of publishing the signed contract transferring the asset ownership. In case of multi-assets transfers a *multisig* scheme [36] can be adopted.

Definition 9 (ownership requirement)**.** *In a blockchain swap protocol, the decision function F_P corresponding to the publishing sequence σ_P is such that $F_P(j) = b_0(A^j) \subseteq \mathcal{O}$ for any $j \in \{1, \ldots, t_P\}, t_P \in \mathbb{N} : t_P \leq m$.*

Focusing on the commitment phase we can derive the following definition.

Definition 10. *A decision function F_T is effective on σ_T if and only if $F_T(k) = O^k$ for any $k \in \{1, \ldots, t_T\}, t_T \in \mathbb{N} : t_T \leq m$.*

Whenever the decision function is effective, agents in O^k have the power to accept or decline (i.e., redeem or not) the acquisition of the asset(s) in A^k.

Now, we are ready to analyze the strategic behaviors of the swap participants in a blockchain swap protocol. In this paragraph we associate a specific blockchain swap protocol (σ_P, σ_T) (e.g., sequential, concurrent publishing and snap commitment) with the corresponding game (in strategic or extensive form).

D. Swap problem as a game

In the economic sphere, the strategies adopted by the agents mostly concern the maximization (or minimization) of their outcomes (e.g., prices or costs). Differently, in decentralized distributed systems (e.g., blockchains) strategies are related more on following or deviating from a prescribed protocol. In the blockchain environment several network and consensus algorithm attacks have been analyzed through game theoretical models [37; 38; 39] in order to qualify systems' strategic robustness. Considering decentralized swap protocols, the participants, intervening in both commitment and publishing phase, are more or less incentivized to stick or not to the protocol regulating the swap. The outcomes of a single agent, varying according to personal strategies, result depending from other agents strategic behaviors, too. What is common knowledge among the swap participants is the swap protocol structure that is represented by the pair (σ_P, σ_T).

1) Preliminary notions on games: A *strategic form game* is a tuple $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ where $N = \{1, \ldots, n\}$ is the set of players, S_i is a finite set of *pure strategies* or *actions* for player i and $u_i : \prod_{j \in N} S_i \to \mathbb{R}$ is a *payoff function* specifying for each *strategy profile* or *state* $s = (s_i)_{i \in N} \in \prod_{j \in N} S_i$ player i' s payoff $u_i(s) \in \mathbb{R}$, for each $i \in N$.

Given a strategy profile $s = (s_i)_{i \in N} \in \prod_{j \in N} S_i$, in the following, s_{-i} will denote s from which the strategy of player i is removed, i.e. $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, s_n)$ and (x, s_{-i}) will denote the strategy profile s from which s_i is replaced by $x \in S_i$, i.e. $(x, s_{-i}) = (s_1, \ldots, s_{i-1}, x, s_{i+1}, s_n)$. We say that x is a *best response* to s_{-i} when $u_i(x, s_{-i}) = \max_{y \in S_i} u_i(y, s_{-i})$. A state $s = (s_i)_{i \in N} \in \prod_{j \in N} S_i$ is a (*pure*) *Nash equilibrium* of the strategic game $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$, if for every player $i \in N$, it holds that s_i is a *best response* to s_{-i} for each $i \in N$.

An *extensive form game (with perfect information)* on N is defined as a tuple $\langle N, T, P, (A_h)_{h \in V}, (u_i)_{i \in N} \rangle$ where $T = (V, E)$ is a tree with root v_0 and the set of end nodes denoted by Z; $P : V \setminus Z \to N$ is a function assigning to each non-end node (also called *decision node*) a player in N; $A_h = \{(x_h, x_i) \in E\}$ for each node $h \in V \setminus Z$ is the set of

edges going from node h to some other node and represents the set of *actions* at node h; $u_i : Z \to \mathbb{R}$ is a *payoff function* for player $i \in N$. A pure strategy s_i for player $i \in N$ is a map assigning an action (edge) $a \in A_h$ to every node in $P^{-1}(i)$ and we denote by S_i the set of all pure strategy. Any *strategy profile* $(s_1, \ldots, s_n)$ results in an end node in Z. So, to any extensive form game $\langle N, T, P, (A_h)_{h\in V}, (u_i)_{i\in N}\rangle$ we can associate a strategic form game $\langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle$. A Nash equilibrium of an extensive form game is a Nash equilibrium of the associated strategic form game. A *subgame* of an extensive form game (with perfect information) on node $v \in V \setminus Z$ is another extensive form game (with perfect information) obtained as a part of the directed tree starting at the decision node v. A *subgame perfect equilibrium* is a strategy profile that induces a Nash equilibrium on any subgame. For more details see, for instance, the book [40].

2) Blockchain swaps as a game: As discussed in Section II-C, blockchain swap protocols in both single and multi-assets case, can be characterized by sequential publishing and commitment, concurrent publishing and snap commitment. For those protocols characterized by concurrent moves (i.e., concurrent publishing) we can adopt a representation with general strategic games.

Definition 11. *Let $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}}\rangle$ be a swap problem, let (σ_P, σ_T) be a blockchain swap protocol and let $F_P : \{1, \ldots, t_P\} \to \mathcal{O}$ and $F_T : \{1, \ldots, t_T\} \to \mathcal{O} \cup \{\tau\}$ be decision functions. We associate the strategic game form $\langle N, (S_i)_{i\in N}, (u_i)_{i\in N}\rangle$ such that:*

- *$N = \mathcal{O}$ is the set of players;*
- *$(S_i)_{i\in N} = \{action\ 1, action\ 0\}$ is the set of pure strategies for player i consisting in the pair (*follow *the protocol,* not follow *the protocol) labelled respectively as action 1 and action 0;*
- *$(u_i)_{i\in N} : u_i : \prod_{j\in N} S_i \to \mathbb{R}$ is the payoff function for asset owners $N = \mathcal{O}$ evaluating the outcomes of type $b_{t_T}^{\sigma_T}$ that is, for every player i we have $u_i\big((b_{t_T}^{\sigma_T})^{-1}(i)\big)$.*

We model protocols with sequential phases with extensive form games in order to represent the sequencing of swap participants' moves and the fact that at each decision point asset owners know the moves history so far. Let us recall that blockchains involved in a swap are of public nature with open read access.

Definition 12. *Let $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}}\rangle$ be a swap problem, let (σ_P, σ_T) be a blockchain swap protocol and a let $F_P : \{1, \ldots, t_P\} \to \mathcal{O}$, $F_T : \{1, \ldots, t_T\} \to \mathcal{O} \cup \{\tau\}$ be decision functions. We associate the extensive game form $\Gamma^\sigma = \langle N, T, P, (A_h)_{h\in V}, (u_i)_{i\in N}\rangle$ such that:*

- *$N = \mathcal{O}$ is the set of players;*
- *T is a (binary) directed tree such that each directed path from the root $v_0 \in V$ to an end node $v \in Z$ is formed by precisely $t + 1$ nodes and t arcs (V is the set of the nodes of the tree T and $Z \subset V$ is the set of leaf nodes);*
- *$P(v) = F_{P,T}(l(v))$, for each $v \in V \setminus Z$, is the publisher/activator of the asset transfer(s) at step $l(v)$ in the protocol (σ_P, σ_T), where $l(v)$ is the number of arcs between v_0 and v on the unique path from v_0 to v (we assume that $l(v_0) = 0$);*
- *A_h for all $h \in V$ is formed by two outgoing arcs in h; one arc in A_h is labeled with* action 1 *(i.e.,* follow *the protocol) and the other one,* action 0 *(i.e.,* not follow *the protocol) for any $h \in V \setminus Z$. So, a unique path from v_0 to an end node $z \in Z$ identifies a binary vector $p^z \in \{0, 1\}^t$ such that p_k^z is the label of the arc starting from node v with $l(v) = k$ on the path from v_0 to z.*
- *Any end node $z \in Z$ is associated with a unique outcome corresponding to the map $b_{t_T}^{\sigma_T^K}$ where $K \subseteq \{1, \ldots, t_T\}$ is such that $p_k^z = 0$ for any $k \in K$, and $p_k^z = 1$ for any $\{1, \ldots, t_T\} \setminus K$. So, for any $i \in \mathcal{O}$, the outcome $b_t^{\sigma_T^K}$ is evaluated by i with the payoff function $u_i\big((b_t^{\sigma_T^K})^{-1}(i)\big)$.*

Proposition 3. *Let $\Gamma^\sigma = \langle N, T, P, (A_h)_{h\in V}, (u_i)_{i\in N}\rangle$ be the extensive form game associated with the swap problem $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}}\rangle$ and the blockchain swap protocol with sequential phases (σ_P, σ_T) and let $F_T : \{1, \ldots, t_T\} \to \mathcal{O} \cup \{\tau\}$ be a decision function. If F_T is effective on σ_T, then the strategy profile $(\hat{s}_1, \ldots, \hat{s}_n)$ that specifies action* 1 *(follow the protocol) at any node is the unique subgame perfect equilibrium (in dominant strategies).*

Proof. For each node v, by the fact that F_T is effective, we have that $P(v) = F_T(l(v)) = O^{l(v)}$. So, at each decision node $v \in V \setminus Z$, if player $P(v)$ specifies action 0 (not follow the protocol) at node v, then by the first claim of Proposition 1, player $P(v)$ ends up with a set of assets that is contained in the one that player $P(v)$ would obtain if she/he specifies action 1 at node v. So, the utility of player $P(v)$ is larger if it chooses, at each decision node, action 1 than action 0. It follows that at each node v action 1 strictly dominates action 0 for player $P(v)$. □

Notice that if (σ_P, σ_T) is efficient, than in Proposition 3 the unique subgame perfect equilibrium (in dominant strategies) corresponds to the desired outcome b_*. Proposition 3 shows that, whenever players in the game have to decide whether to accept or decline an asset acquisition, they are incentivized to follow the protocol by accepting the desired asset. However, a stronger result can be proved.

Proposition 4. *Let $\Gamma^\sigma = \langle N, T, P, (A_h)_{h\in V}, (u_i)_{i\in N}\rangle$ be the extensive form game associated with the swap problem $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i\in\mathcal{O}}\rangle$ and the blockchain swap protocol with sequential phases (σ_P, σ_T) and let $F_T : \{1, \ldots, t_T\} \to \mathcal{O} \cup \{\tau\}$ be a decision function. The decision function F_T is effective on the blockchain swap protocol (σ_P, σ_T), if and only if the strategy profile $(\hat{s}_1, \ldots, \hat{s}_n)$ that specifies action* 1 *(follow the protocol) at any node is the unique subgame perfect equilibrium (in dominant strategies).*

Proof. We have to prove the "*if*" since the "*only if*" directly follows from Proposition 3. By contradiction, if F_T is not effective we have the following cases: (*i*) $P(v) = F_T(l(v)) =$

$b_0^{\sigma_T}(A^{l(v)})$, the original owner of the assets decides whether or not to follow the protocol or, (*ii*) $P(v) = F_T(l(v)) = O^k \neq b_0^{\sigma_T}(A^{l(v)})$ such that $k \in \{1, \ldots, t_T\} : k \neq l(v)$, the activators are any player in the game but the original asset owners and the asset receivers.

(*i*) Whenever the owners of the assets in the initial configuration have to decide between action 0 (not follow the protocol) and action 1 (follow the protocol), by the second claim of Proposition 1 they have no incentive to follow the protocol σ_T. That is, not following the protocol the asset set obtained at time t_T is greater then the one obtained by following the protocol.

(*ii*) Whenever the activators of transfers (A^k, O^k) are neither O^k nor $b_0^{\sigma_T}(A^k)$, the situation is more complex. Assuming that players activate only exchanges of other players, it always exist an activator $o^l(v) : l(v) \in \{1, \ldots, t\}$ that has to decide whether to follow the protocol σ_T or to deviate by originating the sequence σ_T^K (defined in Section II-C). However, the activator would be indifferent to the two strategies since $u_{o^{l(v)}}\big((b_t^{\sigma_T})^{-1}(o^{l(v)})\big) = u_{o^{l(v)}}\big((b_t^{\sigma_T^K})^{-1}(o^{l(v)})\big)$ due to the fact that the cardinality of the asset set remains unchanged.

Therefore, the strategy profile specifying action 1 (follow the protocol) at each stage cannot be a perfect equilibrium in dominant strategies if F_T is not effective. □

This result seems to rule out any swap protocol with a non-effective decision function $F_T : \{1, \ldots, t_T\} \to \mathcal{O}$. However, we can imagine protocols with transaction triggering mechanism not involving receiving agents (see Section III-B).

It is possible to have a subgame perfect equilibrium by adopting a penalty mechanism for protocol deviations. Protocols allowing the owner of the asset in the initial configuration to trigger the exchange have to contemplate a high penalty overcoming the value of the extra-asset gained in case of deviation. On the other hand, protocols with players activating exchanges not involving them may work with small penalties assigned in case of deviation. More precisely, by adopting a proper penalty mechanism, "following the protocol" still represents a subgame perfect equilibrium.

Corollary 1. *If the decision function $F : \{1, \ldots, t_T\} \to \mathcal{O}$ is not effective on a blockhain swap protocol (σ_P, σ_T), then the strategy profile $(\hat{s}_1, \ldots, \hat{s}_n)$ that specifies action 1 at any node is the unique subgame perfect equilibrium (in dominant strategies) only when combined with a penalty function $p : A_h \to \mathbb{R}_+$ such that:*

- $p(0; k) = \epsilon \in \mathbb{R}_+$ *for* $k \neq b_0^{\sigma}(A^{l(v)})$ *and,*
- $p\big(0; b_0^{\sigma_T}(A^{l(v)})\big) = \delta + u_{O^{l(v)}}\big((b_{t_T}^{\sigma_T^K})^{-1}(O^{l(v)})\big) - u_{O^{l(v)}}\big((b_{t_T}^{\sigma_T})^{-1}(O^{l(v)})\big)$ *where* $\delta \in \mathbb{R}_+$.

In the first case the activators are no more indifferent to the two strategies σ_T and σ_T^K while in the second case the penalty matches, for a small δ, the advantage in deviating with respect to following the protocol σ_T.

III. Atomic Cross-Chain Swap protocols

This section is devoted to the analysis of existing cross-blockchains swap protocols aiming to move forward the custodial trading performed by centralized exchanged services.

A. Sequential publishing and commitment

Here we present the swap solution proposed by *Nolan* [16] for permissionless UTXO-based blockchains. Nolan's protocol make use of contracts [41], hash-locks as commitment/locking scheme and time-locks to restore the initial situation consequently to failures in the publishing phase.

Given two asset owners (e.g., Alice and Bob) aiming at cross-swapping two crypto-assets (e.g., x Bitcoins and y Litecoins), the protocol (represented in Fig. 2) works as follows:

1) The agent Alice creates a secret s such that $h = H(s)$, and publishes a contract transferring the ownership of her x Bitcoins to Bob on the Bitcoin blockchain. The contract is locked with the hashlock h and a timelock Δ_{Bob} ensuring that: "Bob can claim the asset property providing s before time Δ_{Bob}".
2) When Bob confirms that Alice's contract has been correctly published on the Bitcoin blockchain, he publishes a contract on the Litecoin blockchain with the same hashlock h but with timelock Δ_{Alice} stipulating that: "before time Δ_{Alice}, Alice can claim the asset property with secret s". Note that $\Delta_{Alice} < \Delta_{Bob}$
3) When Alice confirms that Bob's contract has been correctly published on the Litecoin blockchain, she sends s to Bob's contract (before time Δ_{Alice}), acquiring the y Litecoins and revealing s to Bob.
4) Bob then sends s (in the time interval $[\Delta_{Alice}, \Delta_{Bob}]$) to Alice's contract, acquiring the x Bitcoins and completing the swap.

According to our formalization the two party cross-chain swap is represented as follows:

$$\sigma_P = \{(x, B), (y, A)\}, \quad F_P(j) = \{A, B\}, \quad j = \{1, 2\};$$

$$\sigma_T = \{(y, A), (x, B)\} \quad F_T(k) = \{A, B\}, \quad k = \{1, 2\}.$$

The protocol assumes that swap participants (i) actively monitor the involved blockchain in order to confirm contracts publication and, (ii) adopt a common hashing method. *Herlihy* [29] extends the protocol for multi-assets and multi-agents swaps (with the secret creators forming a *feedback vertex set* L i.e., a subset of V whose deletion leaves $\mathcal{D} = (V, E)$ acyclic) and analyzes under which conditions it is 'atomic'. *Atomicity*, in this case, is defined in a game theoretical fashion: in [29] a swap protocol is defined as atomic if:

- "following the protocol" is a *nash equilibrium strategy* and,
- no conforming party is affected (in terms of payoff) by a protocol deviation.

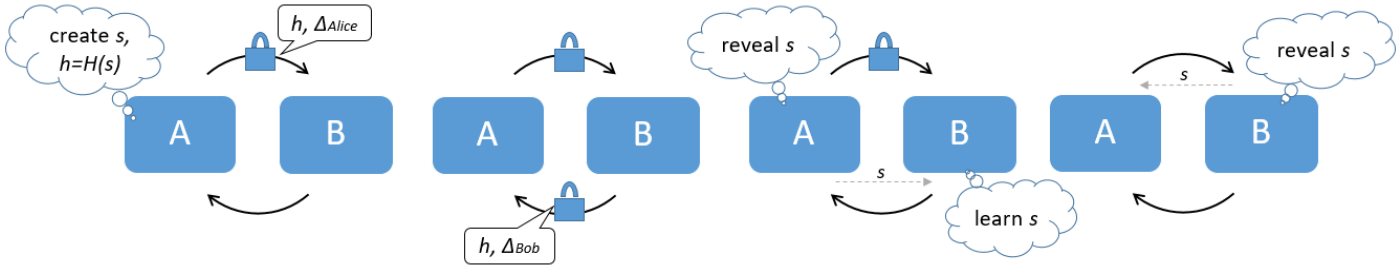

Fig. 2. Two party 'atomic' cross-chain swap protocol proposed in [16; 29] characterized by sequential publishing and commitment: $\sigma_P = \{(x, B), (y, A)\}$ and $\sigma_T = \{(y, A), (x, B)\}$.

Moreover, multi-asset swaps that are atomic according to [29] have a *strongly connected* corresponding digraph $\mathcal{D}$ (i.e., for every pair of distinct owners there is always a path from one to the other and viceversa).

1) The separating agent: In multi-players protocol implementations, the role of the secret creator or *leader*, denoted as l, becomes crucial. Let us consider only single leader swap protocols. The latter initiates the publishing and most importantly the commitment phase of the protocol by disseminating the secret s. A swap leader reveals the secret whenever all the contracts transferring her the ownership of the desired crypto-assets are correctly published. The sequentiality of the publishing together with the leader role ensures the required separation between the two phases of the blockchain swap. Let us formalize this concept in the following proposition:

Proposition 5. *A blockchain swap protocol verifying the atomicity definition proposed in [29] and characterized by:*

(i) *a leader participant initiating the publication phase $F_P(1) = \{l\}$ together with the commitment one $F_T(1) = \{l\}$ when all the contracts where she is directly involved are published,*

(ii) *a sequential publishing phase where asset owners are called to publish as soon as all the contracts transferring them the desired assets' property are published,*

satisfies the commitment requirement of Definition 8.

Proof. We can state that the last contract(s) to be published are the ones involving the leader acting as a receiver; $O^{t_P} = \{l\}$. If a contract is not correctly published ($\exists \bar{j} \in \{1, \ldots, t_P\} : O^{\bar{j}} \cap b_0(A^{\bar{j}}) \neq \emptyset$), either the receiver is the leader, hence the commitment cannot start for (i), or the receiver is a different participant which for (ii) cannot publish her contracts. Indeed, since the graph is strongly connected, there is a sequence of participants from the receiver to the leader that will not activate their contracts. □

2) Protocol strategic behaviour: The blockchain swap protocol presented above works with a sequential commitment phase where swap participants trigger contracts transferring them the ownership of the desired assets. Therefore, the decision function characterizing the protocol is effective. It should be noted that thanks to Proposition 4 we can derive a stronger result than the one proposed by the original paper concerning the players strategic behaviour.

Corollary 2. *In the blockchain swap protocol (σ_P, σ_T) presented in [16] (and generalised in [29]), the strategy profile $(\hat{s}_1, \ldots, \hat{s}_n)$ specifying action 1 (follow the protocol) at any node is the unique subgame perfect equilibrium (in dominant strategies).*

Let us provide (Fig. 3) the graphical representation of the extensive form game associated to the blockchain swap protocol (σ_P, σ_T) presented in [16]. We do consider two owners $\mathcal{O} = \{A, B\}$ swapping two assets $\mathcal{A} = \{x, y\}$ where $t_P = t_T = 2$. Alice, as leader of the swap protocol, is the first

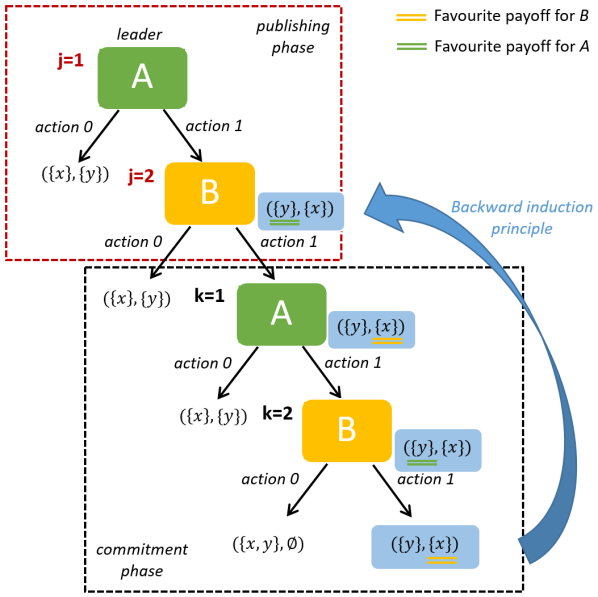

Fig. 3. Extensive form game associated to the blockchains swap protocol, presented in [16]. Publishing and commitment phases are sequentially executed one after the other. Outcomes represent the payoffs (assets) owned by Alice and Bob respectively. The figure represents the backward induction process for each decision step.

player to decide between *action 0* and *action 1*. Whenever one of the two parties opts for action 0 during the publishing phase, due to the commitment requirement (i.e., Definition 8), the games terminates with the original ownership configuration as the outcome: $b_{1,2}^{\sigma_P} = b_0 = (A, B)$ for $\mathcal{A} = \{x, y\}$. If

Alice does not start the commitment phase the outcome $b_1^{\sigma_T}$ coincides with $b_0 = (A, B)$. When Bob is called to decide, Alice's previous moves are known; if Bob decide to follow the protocol the swap takes place, $b_2^{\sigma_T} = b_* = (B, A)$, otherwise Alice acquires both the desired asset and the originally owned one leaving Bob empty-handed, $b_2^{\sigma_T} = (A, A)$. Outcomes in Fig. 3 represent the payoffs of Alice and Bob respectively.

Subgame perfect equilibria are computed by applying the *backward induction* process. By reasoning from the end to the beginning of the game, at each decision step the strategy providing a better payoff (i.e., the one providing greater utility) is selected. Then, considering the game associated to the swap protocol presented in [16], the sequence of optimal actions is the one specifying action 1 at each decision step (Fig. 3).

3) Atomicity violations: In [26] the authors observe that the protocol presented by *Nolan* [16] is not immune to violation of the all-or-nothing atomicity as in Definition 7. More precisely, a time-lock expiration before commitment can lead an honest swap participant to deviate from the protocol. As in Fig. 3 if Bob does not commit the transfer at the last decision step (i.e., if the time-lock expires before) the outcome is $b_2^{\sigma_T} = (A, A)$ which differs from both b_0 and b_*. Crash failures together with network delays are some of the possible causes of time-lock expiration before commitment making the deviating party ending up with an asset loss. A snap commitment may be a solution to atomicity violations in asynchronous environments.

B. Concurrent publishing and snap commitment protocols

In [26] authors propose an atomic swap protocol characterized by a concurrent publishing phase and a snap commitment. The protocol AC3TW works with a centralized trusted authority τ called Trent, acting as a separating agent that (i) verifies the correctness of the publishing phase and, (ii) witness the redemption contracts. Thanks to the trusted witness Trent the protocol benefits from all-or-nothing atomicity and faster publishing phase with respect to the sequential one (i.e., increasing overhead proportional to the number of contracts involved in the swap).

The protocol constructs for every possible swap configuration a directed graph $\mathcal{D} = (V, E)$ similar to the one of [29] (see Section II-A for more details). $\mathcal{D}$ is multisigned by all swap participants in the set V generating a graph multisignature $m_s(\mathcal{D})$. The signatures order is irrelevant, the multisignature represents the participants' agreement on $\mathcal{D}$.

Given two asset owners (e.g., Alice and Bob) aiming at swapping x Bitcoins for y Ethereum here below the steps characterizing the two party AC3TW protocol (represented in Fig. 4):

(1) Alice and Bob create the digraph $\mathcal{D}$ and multisign it generating $m_s(\mathcal{D})$.
(2) The multisignature is registered and stored by the centralized trusted authority, Trent, only if not registered before and it is set to a null value $\rceil$.
(3) Alice publishes the contract C_1 on the Bitcoin blockchain stating that:
   - if Bob provides Trent's signature to the redemption instance, i.e., if he provides $T(m_s(\mathcal{D}); RD)$ then, x Bitcoins' ownership is transferred from Alice to Bob.
   - if Alice provides $T(m_s(\mathcal{D}); RF)$ then, the x Bitcoins are transferred back to Alice.
(4) Concurrently, Bob publishes a contract C_2 on the Ethereum network stating the following:
   - if Alice provides $T(m_s(\mathcal{D}); RD)$ then, y Ethereums' ownership is transferred from Bob to Alice.
   - if Bob provides $T(m_s(\mathcal{D}); RF)$ then, the y Ethereums are transferred back to Bob.
(5) After the publication of $C_i : i = 1, 2$ either Alice or Bob requests Trent to trigger a redemption commitment scheme to redeem the assets. Trent issues $T(m_s(\mathcal{D}); RD)$ only if both C_1 and C_2 are correctly published in their corresponding blockchains and the value of $m_s(\mathcal{D})$ stored by Trent is $\rceil$.
(6) Whenever a contract is not correctly published any participant can request Trent to trigger a refund commitment scheme. $T(m_s(\mathcal{D}); RF)$ is issued only if $m_s(\mathcal{D})$ has value $\rceil$.
(7) Depending on the case, Trent sets the value of $m_s(\mathcal{D})$ to $T(m_s(\mathcal{D}); RD)$ or $T(m_s(\mathcal{D}); RF)$ accordingly.

According to our formalization the two party cross-chain AC3TW protocol is represented as follows:

$$\sigma_P = \{(\{x, y\}, \{A, B\})\}, \quad F_P(j) = \{A, B\}, \quad j = \{1\};$$

$$\sigma_T = \{(\{x, y\}, \{A, B\}, \{X^1(x) = B, X^1(y) = A\})\}$$
$$F_T(k) = \tau, k = \{1\}.$$

All-or-nothing atomicity is achieved by the fact that the redemption $T(m_s(\mathcal{D}); RD)$ and the refund $T(m_s(\mathcal{D}); RF)$ events are mutually exclusive. The protocol meets the commitment requirement due to Trent's witnessing activity. The latter reduces as well the *interactivity* (i.e., the active participation of the swap participants) of the swap protocol [28] with respect to Nolan's swap implementation where asset owners have to be constantly on-line monitoring the involved blockchains. The two contracts of the AC3TW protocol are respectively a Bitcoin contract C_1 and an Ethereum smart contract C_2 (see [41; 42; 43] for more details) therefore, the protocol works in both UTXO-based and account-based blockchains [44]. Moreover, concerning the protocol strategic behaviour we have that "following the protocol" is a Nash equilibrium.

Proposition 6. *Let $\langle N, (S_i)_{i \in N}, (u_i)_{i \in N} \rangle$ be the strategic form game associated with the swap problem $\langle \mathcal{A}, \mathcal{O}, b_0, b_*, (u_i)_{i \in \mathcal{O}} \rangle$ and the blockchain swap protocol (σ_P, σ_T) characterized by a concurrent publishing and a snap commitment where the decision function F_T is such that $F_T(k) = \tau \ \forall k \in \{1, \ldots, t_T\}$. Then, the strategy profile $(\hat{s}_1, \ldots, \hat{s}_n)$ that specifies action 1 (follow the protocol) for every player i is a Nash equilibrium.*

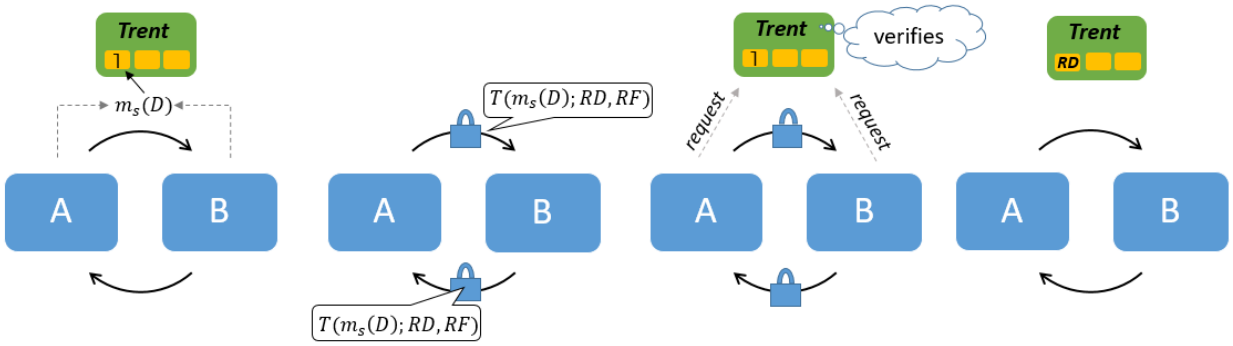

Fig. 4. Two party AC3TW cross-chain swap protocol proposed in [26] characterized by concurrent publishing and snap commitment: $\sigma_P = \{(\{x, y\}, \{A, B\})\}$ and $\sigma_T = \{(\{x, y\}, \{A, B\}, \{X^1(x) = B, X^1(y) = A\})\}$.

Proof. Given a strategy profile $s = (s_i)_{i \in N} \in \prod_{j \in N} S_i$ whenever $\exists i \in N$ such that s_i is action 0 then, due to Trent witnessing, the protocol ends up in the original configuration b_0. The outcome corresponding to the desired configuration b_* is reached whenever action 1 is chosen by all the players. Hence, since $u_i(b_0^{-1}(i)) < u_i(b_*^{-1}(i)) \ \forall i \in N$, action 1 is the *dominant best response strategy* to all $s_{-i} \ \forall i \in N$. □

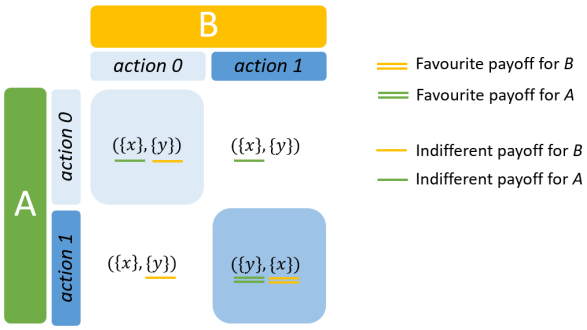

Fig. 5. Strategic form game associated to the blockchains swap protocol, presented in [26]. Alice and Bob decide whether to publish or not the contract on the corresponding blockchain. Outcomes represent the payoffs of Alice and Bob respectively. By eliminating dominated strategies the emphasized outcomes are the equilibria of the game. The dominant strategy, the one providing strictly greater payoffs, is a Nash equilibrium.

Let us analyze the strategic form game associated to the blockchain swap protocol presented in [26] in the case of two owners $\mathcal{O} = \{A, B\}$ swapping two assets $\mathcal{A} = \{x, y\}$ where $t_P = t_T = 2$. Strategic form games represent situations where players make decision simultaneously. In this cases, a matrix representation (Fig. 5) allows to quickly analyze each possible outcome. In this case, Alice and Bob have to decide between *action 0* and *action 1* during the publishing phase. Once the transfers are correctly published (i.e., action 1 is chosen by both players) Trent commit the swap, $b_1^{\sigma_T} = b_* = (B, A)$ for $\mathcal{A} = \{x, y\}$. On the other hand, if one of the two parties chooses action 0 the outcome is $b_1^{\sigma_P} = b_0 = (A, B)$.

In order to identify the game's equilibria, *dominated strategies* (i.e., strategies providing a lower utility than others) have to be eliminated. Two different equilibria are computed: "following the protocol" is a dominant strategy always providing a greater utility for all the other players strategies, "do nothing" (i.e., choosing action 0) is a weakly dominant strategy that provides the same payoffs for all the other players strategies. Since dominant strategies are always Nash equilibria [40], the startegy profile specifying action 1 for every player of the game is a Nash equilibrium.

IV. Conclusions

This paper formalizes the distributed cross-chain swap problem in the blockchain context where parties exchange assets across multiple blockchains. To the best of our knowledge this work is the first to propose a complete framework allowing to analyze existing cross-chain swap protocols as strategic games. We prove that (i) following a swap protocol characterized by an *effective* decision function (e.g. the protocol proposed in [16] and generalised in [29]) is a subgame perfect equilibrium in dominant strategies while (ii) following a swap protocol characterized by concurrent publishing and snap commitment (e.g. the protocol proposed in [26]) is a Nash equilibrium. The presented framework can be further improved by considering a formal adversary model and *concurrent commitment* schemes (e.g., AC3WN [26]). Our work opens several research directions. We are currently considering cooperative scenario where agents may collaborate to form coalitions. Analyzing the resilience of the cross-chain swap protocols to deviating coalitions is a challenging research direction. Another interesting open question is generalizing our framework in order to characterize protocols such as Lightning Networks.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, accessed on January 10, 2020. https://bitcoin.org/bitcoin.pdf.

[2] "Hyperledger Architecture Vol.1, Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus," accessed on January 10, 2020. https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG_Paper_1_Consensus.pdf.

[3] V. Buterin, "Ethereum White-paper," 2016, accessed on January 10, 2020. https:github.com/ethereum/wiki/wiki/White-Paper.

[4] M. Green and I. Miers, "Bolt: Anonymous payment channels for decentralized currencies," in *ACM CCS*, 2017.

[5] M. Hearn, "Corda: A distributed ledger," accessed on January 10, 2020. https://docs.corda.net/head/_static/corda-technical-whitepaper.pdf.

[6] D. Schwartz *et al.*, "The Ripple protocol consensus algorithm – White Paper," 2014, accessed on January 10, 2020. https://ripple.com/files/ripple_consensus_whitepaper.pdf.

[7] "Tendermint Byzantine-fault tolerant state machine replication," accessed on January 10, 2020. http://tendermint.com.

[8] "Kyber: An On-Chain Liquidity Protocol," accessed on January 10, 2020. https://files.kyber.network/Kyber_Protocol_22_April_v0.1.pdf.

[9] "Open Application Network," accessed on January 10, 2020. https://github.com/aionnetwork.

[10] "Cosmos: A Network of Distributed Ledgers," accessed on January 10, 2020. https://cosmos.network/cosmos-whitepaper.pdf.

[11] "Polkadot: Vision for a Heterogeneous Multi-Chain Framework," accessed on January 10, 2020. https://polkadot.network/PolkaDotPaper.pdf.

[12] B. A. Lewis, P.M. and M. Kifer, *Databases and transaction processing: an application-oriented approach*. Addison-wesley Reading, 2002.

[13] M. Borkowski *et al.*, "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.

[14] "Etherdelta," accessed on January 10, 2020. https://etherdelta.com/.

[15] W. Warren, "Front-running, griefing and the perils of virtual settlement (part 1)," accessed on January 10, 2020. https://blog.0xproject.com/front-running-griefing-and-theperils-of-virtual-settlement-part-1-8554ab283e97.

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

[17] "Decred cross-chain atomic swapping," accessed on January 10, 2020. https://github.com/decred/atomicswap.

[18] "Bitcoin abc," accessed on January 10, 2020. https://github.com/Bitcoin-ABC/bitcoin-abc.

[19] "Bitcoin unlimited," accessed on January 10, 2020. https://github.com/BitcoinUnlimited/BitcoinUnlimited.

[20] "Bitcoin xt," accessed on January 10, 2020. https://github.com/bitcoinxt/bitcoinxt.

[21] "Litecoin core integration/staging tree," accessed on January 10, 2020. https://github.com/litecoin-project/litecoin.

[22] "Qtum project," accessed on January 10, 2020. https://github.com/qtumproject/qtum.

[23] "Komodo barterdex," accessed on January 10, 2020. https://github.com/KomodoPlatform/BarterDEX.

[24] "Komodo (advanced blockchain technology, focused on freedom)," accessed on January 10, 2020. https://docs.komodoplatform.com/whitepaper/introduction.html.

[25] "Blockchain.io (your gateway to the internet of value)," accessed on January 10, 2020. https://blockchain.io/.

[26] V. Zakhary, D. Agrawal, and A. Abbadi, "Atomic commitment across blockchains," *Proceedings of the VLDB Endowment*, 2020.

[27] E. Heilman, S. Lipmann, and S. Goldberg, "The arwen trading protocols," 2019.

[28] A. Zamyatin *et al.*, "Xclaim: Trustless, interoperable, cryptocurrency-backed assets," *IEEE Security and Privacy. IEEE*, 2019.

[29] M. Herlihy, "Atomic cross-chain swaps," in *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 2018, pp. 245–254.

[30] M. Miraz and D. Donald, "Atomic cross-chain swaps: Development, trajectory and potential of non-monetary digital token swap facilities," *AETiC*, vol. 3, 2019.

[31] J. Zie *et al.*, "Extending atomic cross-chain swaps," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2019, pp. 219–229.

[32] B. Wiki, "Hashlock," accessed on January 10, 2020. https://en.bitcoin.it/wiki/Hashlock.

[33] ——, "Timelock," accessed on January 10, 2020. https://en.bitcoin.it/wiki/Timelock.

[34] Coincer, "Atomic protocol n.1," accessed on January 10, 2020. https://www.coincer.org/2015/01/27/atomic-protocol-1/.

[35] ——, "Atomic protocol n.2," 2015, accessed on January 10, 2020. https://www.coincer.org/2015/02/03/atomic-protocol-2/.

[36] L. Harn, "Digital multisignature with distinguished signing authorities," *Electronics Letters*, vol. 35, no. 4, pp. 294–295, 1999.

[37] T. Eyal and E. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," *Communications of the ACM*, vol. 61, no. 7, pp. 95–102, 2018.

[38] E. Pagnotta and A. Buraschi, "An equilibrium valuation of bitcoin and decentralized network assets," *Available at SSRN 3142022*, 2018.

[39] W. Wang *et al.*, "A survey on consensus mechanisms and mining management in blockchain networks," *arXiv preprint arXiv:1805.02707*, pp. 1–33, 2018.

[40] H. Peters, *Game theory: A Multi-leveled approach*. Springer, 2015.

[41] B. Wiki, "Distributed contract," accessed on January 10, 2020. https://en.bitcoin.it/wiki/Contract.

[42] T. Dickerson *et al.*, "Adding concurrency to smart contracts," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*. ACM, 2017, pp. 303–312.

[43] L. Cong and Z. He, "Blockchain disruption and smart contracts," *The Review of Financial Studies*, vol. 32, no. 5, pp. 1754–1797, 2019.

[44] M. Belotti *et al.*, "A vademecum on blockchain technologies: When, which and how," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3796–3838, 2019.