A Game-Theoretic Analysis of Cross-Ledger Swaps with Packetized Payments

Alevtina Dubovitskaya[1,2][0000−0002−9669−1250], Damien
Ackerer[3][0000−0002−3408−2185], and Jiahua Xu[4,5][0000−0002−3993−5263]

[1] Lucerne University of Applied Sciences and Arts
`alevtina.dubovitskaya@hslu.ch`
[2] Swisscom
[3] Covario
`damien.ackerer@covar.io`
[4] UCL Centre for Blockchain Technologies
`jiahua.xu@ucl.ac.uk`
[5] École Polytechnique Fédérale de Lausanne (EPFL)

Abstract. We propose a game-theoretic framework to study the outcomes of packetized payments, a cross-ledger transaction protocol, with strategic and possibly malicious agents. We derive the transaction failure rate and demonstrate that without disciplinary mechanisms, packetized payments are likely to be incomplete. Our analysis suggests that collateral deposits can prevent malicious agents from taking advantage of the protocol. We further infer that the deposit amount should depend on the underlying asset price volatility or that it should be dynamically adjusted as the price changes.

Keywords: Blockchain · Packetized payments · Atomic swaps

1 Introduction

1.1 Background

HTLCs Hashed Time Lock Contracts (HTLCs) have been recently proposed [3] to achieve atomicity of a cross-ledger transaction without any connections between the ledgers, and are often employed in decentralized exchanges (DEX)[6] to complete peer-to-peer exchange [6,1].

An atomic swap with HTLCs starts with one transactional agent, say Alice, randomly generating a secret key. Alice then locks her asset in an HTLC that will transfer the asset to her counterparty, say Bob, upon verification of the secret key. Bob subsequently locks his asset in an HTLC that will transfer the asset to Alice upon verification of the same secret key. The swap completes when Alice unlocks Bob's asset with the secret key generated by herself, which simultaneously exposes the secret key, allowing Bob to also unlock Alice's asset.

---

[6] Cross-ledger DEX protocols are not to be confused with DEX protocols operated within one chain, such as automated market makers (AMM) on Ethereum [16].

Should Alice fail to unlock Bob's asset with her secret key, the two HTLCs will respectively send the locked asset to their original owners when the time locks expire.

One problem of HTLCs is that they create a free option for Alice, who can ultimately choose when and whether or not to expose the secret key, thus delaying the completion of the swap or even causing it to fail. Bob also has the option not to lock his asset, which leads to blocking of Alice's funds for nothing [15].

Packetized payments (PP) Robinson [12] underlines the aforementioned problems associated with HTLCs, and proposes an alternative approach for cross-ledger atomic swaps: packetized payments (PP). Originally developed as part of the Interledger Protocol [5] named *Hyperledger Quilt*,[7] a cross-ledger swap with packetized payments is conducted with a series of alternating transactions.

First, the total asset amounts to be traded are split into N "economically-insignificant" amounts (Figure 1a Step A). Next, these small portions of assets will be sent on one and then on another blockchain sequentially: Steps B and C are to be repeated N times in order to complete the transaction.

Note that, at each iteration, the protocol may require the agent to match and extend the previous transfer such that the agents are alternately exposed to counterparty risk (Figure 1b). Otherwise, the payment initiator would have to agree to always bear the risk of abandonment from the other agent. If one agent behaves maliciously and does not execute the transfer when it is his or her turn, the counterparty loses *only* a fraction of the asset they would be willing to trade. Therefore, PP caps the amount of assets that can be lost at a fraction of the asset determined at Step A and prevents the whole amount of assets being blocked for a long period of time. It also prevents a potential loss of the whole amount of assets, while requiring only simple transfer transactions.

1.2   Contribution

Our framework builds on finite extensive-form games with imperfect information [9], where the only known unknown information is the counterparty's type, which can be either honest or malicious. We study agents' strategies and derive preference parameter conditions consistent with their actions. We also derive the transaction failure rate as a function of the percentages of honest and malicious agents.

We show that in a swap game with packetized payments, it is impossible to enforce malicious agents to complete the transaction without an additional disciplinary mechanism. We illustrate that the "biased" preferences of agents for completed transactions have to be economically large, which motivates the necessity of alternative contracting mechanisms such as collateral deposits. Still, we infer that the initial collateral amount should depend on the asset price volatility, or that it should be dynamically adjusted as the asset price fluctuates.

---

[7] `https://github.com/hyperledger/quilt`

(a) Generic PP   (b) Match-and-extend PP with odd (left) and even (right) N
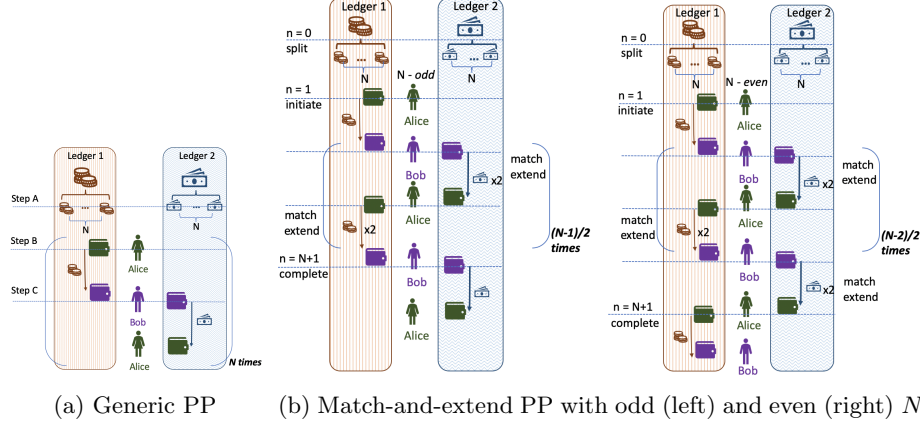
Fig. 1: Packetized payment (PP) schemas.

As the first cross-chain packetized payment protocol *Hyperledger Quilt* is yet to be launched and empirical evidence is absent, our work provides the first simulation result that can facilitate further development of the protocol.

We focus on packetized payments, yet our approach can be extended to other cross-ledger transaction protocols.

2 A game-theoretic analysis

2.1 Framework

Two agents, Alice and Bob, or a and b, want to exchange one unit of asset 1, say one Altcoin, from a for some units of asset 2, say Tether (USDT), from b. We assume that asset 2 is the reference asset in which the agents value their goods. We denote P_t the time-t price of asset 1 expressed in units of asset 2, for example the price of one Altcoin in USDT. We assume for simplicity that there is no interest rate or coin staking, meaning that the asset quantities do not increase by themselves whenever locked in a special wallet or account. Therefore, only the price of asset 1 is stochastic in our framework.

There are three possible times t at which the agents may take actions: 0, 1, and 2. The price dynamics of asset 1 is given by

$$P_t = P_{t-1} \pm \delta \tag{1}$$

for $t = 1, 2$ with equal probability of up and down moves, for some initial price $P_0 > 0$ and some constant $\delta > 0$ such that $\delta \leq \frac{P_0}{2}$ so that the price remains non-negative during the game. Note that the asset price is a martingale, that is the expected value of next period's price is equal to the current price, $\mathbb{E}\left[P_t \mid P_{t-1}\right] = P_{t-1}$ for $t = 1, 2$.

There are three types of actions that the agents may take: continue c, wait w, and stop s. If an agent plays s then the game is over and the transaction fails. If an agent plays w then one time period passes and the price changes. If an agent plays c then either it is the other agent's turn, or the transaction is completed. The agents take actions sequentially and the set of possible actions at a particular instant depends on the history of previous actions.

We assume that the agents are strategic and aim to maximize their interests which is a function of two terms: the financial profit resulting from the asset price change, and the transaction success. Indeed, transaction failures typically have a negative economic impact on agents by delaying further trade actions, and increasing the exposure to price risk. We assume that there are two types of agents: the honest or high type h, and the malicious or low type l. We formalize the two types in the following definition.

Definition 1 (Agent types). *An agent of type h, namely* honest*, always chooses to play continue c. An agent of type l, namely* malicious*, satisfies the parameter condition $\alpha_{i,l} = 0$ for $i = a, b$.*

We model the agent incentives using a utility function as follows:

$$\mathcal{U}(i, j) = \alpha_{i,j} X + \beta_i XY \tag{2}$$

for any agent $i \in \{a, b\}$ of type $j \in \{h, l\}$, and where $X = 1$ indicates transaction success and $X = -1$ transaction failure, and Y is the profit and loss resulting from the asset price change and transfer. The constant $\alpha_{i,j} \geq 0$ measures the extent to which an agent is willing to complete the transaction. For example, if $\alpha_{i,j}$ is large then the agent will most likely prefer to complete the transaction despite an adverse price change. We set $\beta_b = 1$ and $\beta_a = -1$ modeling the agent's opposite exposures to price changes. Note that if the transaction fails, that is $X = -1$, then Alice is positively exposed to Y because asset 1 was not transferred to Bob as $\beta_a X = 1$ in this case.

In Section 2.2, we derive the optimal strategy of the malicious agent, and the conditions on $\alpha_{i,h}$ such that an agent is *willingly honest*.

We denote μ_i the fraction of honest agents i and, thus, $1 - \mu_i$ the fraction of malicious agents i for $i \in \{a, b\}$. The agents meet at random, and each does not know whether the other agent is malicious or not. Furthermore, the agents have full information about their environments. We write $\mathbb{E}\left[\mathcal{X} \mid \mathcal{Y}\right]$ the expected value of the variable $\mathcal{X}$ given the history of actions and other possible refinements $\mathcal{Y}$. We write $\mathcal{T}(i)$ the type of agent i, for example $\mathcal{T}_a = h$ means that Alice is honest. We denote $\mathcal{A}(j, \mathcal{H})$ the best response, or action taken, by an agent of type j following the history $\mathcal{H}$, which is defined as the action maximizing their expected utility.

Note that Alice and Bob must take into account the likelihood that they can be trading with either a malicious or an honest agent. For example, the expected utility of a type j Bob conditioned on the history of actions $\mathcal{H}$ is given by

$$\begin{aligned}
\mathbb{E}\left[\mathcal{U}(b, j) \mid \mathcal{H}\right] = {} & \mu_a \mathbb{E}\left[\mathcal{U}(b, j) \mid \mathcal{H}, \mathcal{T}_a = h\right] \\
& + (1 - \mu_a)\mathbb{E}\left[\mathcal{U}(b, j) \mid \mathcal{H}, \mathcal{T}_a = l\right]
\end{aligned}$$

Table 1: Summary of notations

Notation	Description
a and b	Alice and Bob
h and l	honest and malicious
c, w, and s	actions: continue, wait, and stop
$\mathcal{T}_i$	agent i type
μ_i	honest agent i percentage, $\mathbb{P}[\mathcal{T}_i = h]$
$\mathcal{A}(j, \mathcal{H})$	agent type j action after $\mathcal{H}$
X	swap success (1) or failure (-1)
Y	financial profit and loss
$\alpha_{i,j}$	agent preference parameter for swap success
β_i	$\beta_a = -1$ and $\beta_b = 1$ indicate the asset price exposure direction
P_t	time-t price of asset 1 denominated in asset 2
δ	one-period price change of asset 1 denominated in asset 2

where $\mathbb{E}\left[\mathcal{U}(b,j) \mid \mathcal{H}, \mathcal{T}_a = l\right]$ denotes the expected utility of type j Bob under the assumption that Alice is malicious, and so on.

We use brackets to denote the history of actions, for examples $\{\emptyset\}$ for no action taken and $\{c, w, c\}$ for continue–wait–continue actions. Which agent played a particular action and whose turn it is to play next will be clear from the game descriptions. Notations are summarized in Table 1.

2.2 A short packetized payment game

Packetized payments split the transaction into small transfers where each agent exposed herself or himself to a one-way transfer alternately. At any point in time, one agent may decide not to transfer furthermore and stop the transaction. As a consequence, the variable Y depends on the exit time and is given by

$$
Y_n = \begin{cases}
0 & n = 0, \\
\frac{n}{N} P_{t_n} - \frac{n-1}{N} P_0 & n \text{ is odd and } 0 < n \le N \\
\frac{n-1}{N} P_{t_n} - \frac{n}{N} P_0 & n \text{ is even and } 0 < n \le N \\
P_{t_n} - P_0 & n = N + 1
\end{cases}
\tag{3}
$$

where the subscript n indicates the current step of the transaction, $\frac{1}{N}$ is the granularity amount of the PP, and t_n indicates the time at which the n-th step takes place.

For clarity of exposition, we study a swap performed in 3 payment transactions in total so that $t_n = n$ for $n = 0, 1, 2$ (see Figure 2a). Still, this setup is sufficient to illustrate the functioning of packetized payments.

The sequence of actions for the packetized payment game are described in Figure 2b. In summary, Alice transfers half of the asset to Bob, then Bob makes the whole P_0 payment, and finally Alice transfers the remaining half of the asset. The agents can only decide to continue c or stop s. However, when an

(a) 3-step PP schema

(b) Sequence of actions (∘: root node)

Fig. 2: Match-and-extend PP in 3 steps

agent continues, i.e. plays c, we assume that the transaction also waits, i.e. plays w, for a short time period over which the asset price changes. Note that the game resembles the *centipede game* from [13], however there are important differences: the agents have partial information, and the payoffs are stochastic. Indeed, each agent does not know the other agent type, honest or malicious, and the payoff they get depends on the asset price which is stochastic.

Remark 1. We assumed that Bob matches Alice's payment of $\frac{1}{N}$ and sends an additional $\frac{1}{N}$ payment at the same time (match and extend). This is a fairer mechanism as the agents alternately expose themselves to a loss of $\frac{1}{N}$. Indeed, the alternative would be to let Bob only match while Alice initiates all the payments. This would however result in Alice being the only one exposed to counterparty risk and would require $2N$ transfers instead of $N + 1$.

The first and striking result is that malicious agents, either Bob or Alice, will never complete the transaction. Indeed, there is no incentive for an agent who only cares about its financial profit to complete the transaction, as shown in Proposition 1. All proofs can be found in Appendix.

Proposition 1 (Malicious Alice and Bob). *We have that $\mathcal{A}(l, \{c, w\}) = s$ and $\mathcal{A}(l, \{c, w, c, w\}) = s$.*

From this result we can also infer the percentage of failed transactions.

Corollary 1 (Transaction failure probability). *Assuming that both malicious and honest agents participate in the transaction, then the percentage of incomplete transactions is $1 - \mu_b \mu_a$.*

As it is always best for the malicious type to stop, the transaction will only be completed if the two agents are honest. We derive a necessary condition so that Bob is honest and continues the transaction.

Proposition 2 (Honest Bob). *Assume that Alice of either type plays c at the initial step. Then Bob is honest if and only if*

$$\mu_a > \frac{2P_0}{4\alpha_{b,h} + P_0 - \delta}$$

Note that this result holds only if Bob did not update his prior that Alice is honest with probability μ_a, which is the case when both Alice types play c at step 0. We observe that if the price is more volatile, meaning larger δ, then a larger fraction of honest a agents is required. Interestingly, even with no price movement $\delta = 0$ and only honest a agents, we see that Bob's preference parameter for a completed transaction must be fairly large in value, $\alpha_{b,h} > \frac{P_0}{4}$.

We now derive conditions necessary for Alice to be honest.

Proposition 3 (Honest Alice). *Alice is honest if and only if*

$$\alpha_{a,h} > \frac{P_0 + 2\delta}{4} \quad and \quad \mu_b > \frac{P_0}{4\alpha_{a,h} + P_0}. \tag{4}$$

We see that the conditions for Alice to be honest are more stringent than for Bob even in a setup without any malicious agents, that is when $\mu_a = \mu_b = 1$. Notably if the percentage of honest Bob becomes very small, $\mu_b \to 0$, then there cannot exist any honest Alice unless $\alpha_{a,h} \to +\infty$.

3   Discussion

Our game-theoretic analysis shows that there is no incentive for malicious agents to complete a PP transaction. As a consequence, the transaction failure rate should be large and the economic incentive for agents to behave honestly would need to be enormous. Importantly, a malicious agent can enter multiple PP transactions in parallel with different counterparties for larger profits.

From a practical perspective, PPs are relatively simple, but require many transfers, whose total cost is therefore uncertain. Lightning networks [11] can be employed for micropayments needed for packetized payments without incurring high transaction fees. However, this reintroduces the problem of the assets being locked and, in this case, in the form of collateral deposit on the escrow accounts of each agent on each blockchain: Alice and Bob will need to create two micropayment channels, one on top of each blockchain, and lock the collateral on each channel. In addition, if an honest agent does not receive a payment from a counterparty, and is willing to close the micropayment channel, the funds on the escrow account will be blocked for a certain blockchain-specific period of time [7]. It is also worth noticing that there may be small delays between transfers for network validation, which in turn lead to price fluctuations, as described in the PP game. Importantly, PPs cannot be used to exchange non-fungible assets such as CryptoKitties[8] or "digital twins" of physical goods [12].

---

[8] https://www.cryptokitties.co/

3.1   Collateral deposit

Using collateral deposits to reduce the risk of agents exposed to adverse behavior of other agents is not new. For example, applying collateral to disincentivize aborting a fair exchange has been discussed in works on rational fair exchange [14]. Zamyatin et al. [17] suggest using collateral at least equal to the assets locked on the blockchain for a trade. They also propose overcollateralization and a liquidation mechanism to mitigate extreme price fluctuations for short- and long-term cross-ledger transactions. While this ensures that economically rational agents have no incentive to misbehave, a disadvantage of this solution is that if an agent would like to transfer all their assets of one kind, they will be obliged to execute multiple transactions, each with an amount (approximately) equal to a half of the amount of the asset they currently possess.

Based on the proposed game-theoretic model, it can be shown that a marginal amount of collateral is sufficient to prevent agents from behaving maliciously. We modify the frameworks of Section 2 so as to require agents to place collateral which will be lost if they exit the transaction without completing it. The following Proposition shows that this extinguishes malicious behaviors in our framework.

Proposition 4 (PP with collateral).   *Assume that Bob places a collateral larger than $\frac{P_0+\delta}{2}$, and Alice places a collateral larger than $\frac{P_0+2\delta}{2}$, then it is optimal for malicious agents to continue the transaction in the packetized payment game described in Section 2.2.*

Two relevant observations can be made for real-world applications. First, both expressions for the minimum collateral requirement involve the term δ which suggests that collateral demand should be a function of the asset price volatility, which is known to be time varying. Second, for the packetized payment, the initial collateral involves the fractional transfer value $\frac{P_0}{2}$, or $\frac{P_{t_n}}{N}$ in general, which suggests that the collateral requirement can be small, with N large, but should be adjusted dynamically as the asset price changes. Indeed, the price can vary up or down to $P_0 \pm N\delta$ in extreme scenarios, but will in general fluctuate significantly less.

3.2   Reputation mechanism

We have always assumed that an agent cannot predict the strategy of their counterparty ex-ante as the agent types, malicious or honest, are not observable. However, in reality, if an agent trades regularly with another agent that it can identify, or if an agent has some information on the previous behavior of another agent, then a self-selected agent matching can occur instead of a random one.

In principle, as all the transactions executed on a ledger can generally be seen, the transaction history of an agent can be analyzed to build his or her reputation. However, computation of such reputation value is problematic in a case of permissionless blockchains for several reasons. First, an agent can create multiple accounts and attempt to preserve their anonymity. Even though de-anonymization is possible [2], one cannot guarantee a perfect mapping between

one user and all his or her transactions, in the case of multiple accounts. Second, it may not always be possible to distinguish a cross-ledger transaction from a single-chain transaction. However, if these two challenges are addressed, thanks to the book-keeping property [4] and immutability of a ledger, using a reputation mechanism can complement existing protocols.

4   Conclusion and future work

We introduce a game-theoretic approach to model agent behaviors in cross-ledger transactions with packetized payments. We derive conditions for agents to behave honestly or maliciously, as well as different measures of economic and transaction success. We propose to dynamically compute and adjust the collateral amounts in order to enforce honest behaviors among agents, and we discussed the implementation challenges of reputation systems as a disciplinary mechanism.

An important observation is that trustless cross-ledger swap protocols should use disciplinary mechanisms such as collateral deposit. The implementation, cost, performance, and complexity of various protocols on permissionless blockchains supporting smart contracts – e.g. Ethereum, EOS, Tezos [10], and Neo [8] – thus merit future research.

Appendix

Most of the arguments in the proofs below follow the hypothesis that an agent always takes the actions which maximize their expected utility, taking into account future and possibly adversarial actions from the other agent. We always describe the key conditions (inequalities) to be verified but provide limited details on the derivations as they can be long and tedious.

Proof of Proposition 1 At time 2 if $\mathcal{T}_a = l$, then Alice loses $\frac{P_2}{2}$ in utility by playing c instead of s. Similarly, at time 1 if $\mathcal{T}_b = l$, then Bob gets $\frac{P_1}{2}$ in utility by playing s whereas he expects to receive $\mathbb{E}\left[\mathcal{U}(b,l) \mid \{c,c\}\right] = \mu_a(P_1 - P_0) + (1-\mu_a)(\frac{P_1}{2} - P_0)$ if he plays c. We have $\mathbb{E}\left[\mathcal{U}(b,l) \mid \{c,c\}\right] < \frac{P_1}{2}$ since $\delta < \frac{P_0}{2}$ and $\mu_a \le 1$, hence a malicious Bob plays s.

Proof of Corollary 1 The transaction succeeds only if Alice and Bob are honest which happens with probability $\mathbb{P}[\mathcal{T}_a = \mathcal{T}_b = h] = \mu_a \mu_b$.

Proof of Proposition 2 We have $\mathbb{E}\left[\mathcal{U}(b,h) \mid \{c,w,c\}\right] = \mu_a(P_1 - P_0 + \alpha_{b,h}) + (1 - \mu_a)(-\alpha_{b,h} + \frac{P_1}{2} - P_0)$ and $\mathbb{E}\left[\mathcal{U}(b,h) \mid \{c,w,s\}\right] = -\alpha_{b,h} + \frac{P_1}{2}$. We obtain that $\mathcal{A}(b, \{c,w\}) = c$ by taking $P_1 = P_0 - \delta$.

Proof of Proposition 3 We have $\mathcal{A}(h, \{c,w,c,w\}) = c$ if and only if $\alpha_{a,h} + P_0 - P_2 > -\alpha_{a,h} + P_0 - \frac{P_2}{2}$ which is equivalent to $\alpha_{a,h} > \frac{P_0 + 2\delta}{4}$. Then, with $\mathcal{A}(h, \{c,w,c,w\}) = c$, we have that $\mathbb{E}\left[\mathcal{U}(a,h) \mid \{c\}\right] = \mu_b \alpha_{a,h} + (1 - \mu_b)(-\alpha_{a,h} - \frac{P_0}{2})$ and $\mathbb{E}\left[\mathcal{U}(a,h) \mid \{s\}\right] = -\alpha_{a,h}$. Therefore, for agent a to be honest it must also be that $\mu_b > \frac{P_0}{4\alpha_{a,h} + P_0}$

Proof of Proposition 4 This is immediate as malicious agents would never be able to make any profit by exiting prematurely the transaction.

References

1. Decred-compatible cross-chain atomic swapping (2018), `https://github.com/decred/atomicswap/`
2. Biryukov, A., Khovratovich, D., Pustogarov, I.: Deanonymisation of clients in bitcoin P2P network. In: The ACM Conference on Computer and Communications Security. pp. 15–29. ACM (2014). https://doi.org/10.1145/2660267.2660379
3. Herlihy, M.: Atomic cross-chain swaps. In: Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing. pp. 245–254. ACM, Association for Computing Machinery (7 2018). https://doi.org/10.1145/3212734.3212736
4. Ibañez, J.I., Bayer, C.N., Tasca, P., Xu, J.: REA, Triple-Entry Accounting and Blockchain: Converging Paths to Shared Ledger Systems. SSRN Electronic Journal (2021). https://doi.org/10.2139/ssrn.3602207
5. Interledger: Interledger Protocol (2020), `https://interledger.org/rfcs/0027-interledger-protocol-4/`
6. Komodo: Komodo's Atomic-Swap Powered, Decentralized Exchange: Barterdex (2021), `https://docs.komodoplatform.com/whitepaper/chapter6.html`
7. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 17–30. ACM (2016)
8. Neo: Neo White Paper (2020), `https://docs.neo.org/docs/en-us/basic/whitepaper.html`
9. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT press (1994)
10. Perez, D., Xu, J., Livshits, B.: Revisiting Transactional Statistics of High-scalability Blockchains. In: The ACM Internet Measurement Conference. pp. 535–550 (10 2020), `https://dl.acm.org/doi/10.1145/3419394.3423628`
11. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)
12. Robinson, D.: HTLCs Considered Harmful. In: Stanford Blockchain Conference (2019), `http://diyhpl.us/wiki/transcripts/stanford-blockchain-conference/2019/htlcs-considered-harmful/`
13. Rosenthal, R.W.: Games of perfect information, predatory pricing and the chain-store paradox. Journal of Economic Theory **25**(1), 92–100 (8 1981), `https://linkinghub.elsevier.com/retrieve/pii/0022053181900181`
14. Syverson, P.: Weakly secret bit commitment: applications to lotteries and fair exchange. In: The 11th IEEE Computer Security Foundations Workshop. pp. 2–13. IEEE Comput. Soc (1998), `http://ieeexplore.ieee.org/document/683149/`
15. Xu, J., Ackerer, D., Dubovitskaya, A.: A Game-Theoretic Analysis of Cross-Chain Atomic Swaps with HTLCs. In: IEEE 41st International Conference on Distributed Computing Systems (ICDCS) (2021)
16. Xu, J., Vavryk, N., Paruch, K., Cousaert, S.: SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols (3 2021), `http://arxiv.org/abs/2103.12732`
17. Zamyatin, A., Harz, D., Lind, J., Panayiotou, P., Gervais, A., Knottenbelt, W.: XCLAIM: Trustless, interoperable, cryptocurrency-backed assets. In: IEEE Symposium on Security and Privacy. pp. 193–210. IEEE (5 2019), `https://ieeexplore.ieee.org/document/8835387/`