Promise: Leveraging Future Gains for Collateral Reduction in Competitive Markets

Abstract.

Comment by Dominik: Proposal: change the paper to restrict the mechanism to protocols in which there is competition, i.e. Bob can exit when Alice is malicious. I think one of the biggest problems is that Promise does not work when you have a monopoly situation.

Collateral employed in cryptoeconomic protocols protects against the misbehaviour of economically rational agents, compensating honest users for damages and punishing misbehaving parties. The introduction of collateral, however, carries two disadvantages: (i) requiring actors to lock up substantial amount of collateral can be an entry barrier, limiting the set of candidates to wealthy agents; and (ii) affected agents incur ongoing opportunity costs as the collateral cannot be utilized elsewhere.
We present Promise, a mechanism to decrease the initial capital requirements of economically rational service providers in cryptoeconomic protocols. Promise leverages future income (e.g. transaction fees) prepaid by users to reduce the collateral actively locked up by service providers, while sustaining secure operation of the protocol. We provide a model for evaluating the effectiveness of Promise and argue its security.

1 Introduction

Since their nascence, arguably the most significant property of blockchains is their facilitation of trustless exchange between entities with *weak identities* [3] [1]. Yet the trustless nature of the systems means not only that parties *may* transact without trusting each other, but also that they *should not trust* each other. This creates a design challenge for interactions which would typically involve such trust. In this paper, we focus on blockchain protocols which, at least in part, express *trust* through *collateral*. Here, collateral is value escrowed by party A to guarantee party B that A will not misbehave given that A's economic gain from cheating is not higher than the collateral value. In particular, payment protocols can be designed such that B is guaranteed to receive at least the same amount of funds from A that are at risk. Protocols involving collateral include cross-chain communication [14], scalable off-chain payments [9], state channels [4], and watchtowers [10].

[1] Nodes, with *weak* identities, are able to suddenly leave a network; agents, with *strong* identities, cannot.

Problem. Collateralization requires the provision of a substantial amount of funds upon protocol initialization, limiting the set of participants to a selected few. Leaving participation to a small set of agents can lead to phenomenons like "rich are getting richer" through wealth compounding [5]. While it is not possible to grant less wealthy agents proportionally higher rewards due to Sybil identities, we can lower the entry barrier for agents to join a protocol. Finally, locked funds result in opportunity costs for the agent who could use their collateral for participating in other protocols [8].

This work. We present Promise, a simple but effective mechanism to lower entry barriers for intermediaries in protocols relying on collateral for secure operation. Instead of locking up a significant amount of funds as collateral, Promise allows intermediaries to stake their future income, "promised" for correctly providing the service (i.e., fees), instead. Similar to online orders in e-commerce, users can choose to pay fees upfront ("forward payments") – for a some pre-agreed service period. However, instead of transferring these payments directly to the intermediary, users lock pre-paid fees in a escrow smart contact, preventing theft by either party. The intermediary needs to provide the service honestly for the entire period set by the user. The benefit of this scheme is twofold: (i) the intermediary is incentivized to act honestly while enjoying a lower initial collateral, and (ii) the user can reduce his transaction cost and only pays if the service was provided honestly over his defined period. As long as the expected future revenue from correct operation exceeds potential gains by the intermediary, users have the option to leave the protocol, and misbehavior can be proved to the smart contract, Promise incentivizes correct behavior.

Outline. We introduce the system model and assumptions in Section 2, followed by a description of Promise in Section 3. Next, we discuss the security of Promise in Section 4. We discuss related work in Section 5 and conclude in Section 6.

2 System Model

In Promise, a user Bob engages an intermediary Alice to fulfill a task valued at V_B on his behalf. Bob pays Alice p each period t for performing the task. Given the absence of *strong identities*, the total value of the task to Bob (V_B) needs to be fully collateralized, via a deposit D, such that $D \geq V_B$. For example, if a particular task involves Alice offering a service and Bob having a $100 exposure—in the form of counter-party risk—to Alice, Alice will need to post at least $100 as collateral to *insure* offset the exposure, such that Bob does not stand to lose funds if Alice behaves maliciously.

Formally, we adopt the definitions of agreements in cryptoeconomic protocols from [8]. The service providing agent Alice A and the receiving agent Bob B participate in an agreement encoded by:

$$\mathcal{A} = \langle \Phi, p, D \rangle \tag{1}$$

In such an agreement, Alice needs to fulfil the specification Φ and provide the collateral D in advance. When Alice fulfills the specification, the payment p held in escrow is released to Alice.

2.1 Specifications

Comment by Dominik: We need to explain what specifications are and give some examples here.

2.2 Roles

Promise adopts the BAR model of rational agents [1] including private preferences of agents as proposed in [8]. We define the following roles.

- **Alice**, the Intermediary: Alice is economically rational and entrusted with executing a task. She provides a deposit D into the escrow before executing the task and receives a payment p upon successful completion.
- **Bob**, the User: Bob represents the user requesting execution of a task by Alice. A user provides payments $\{p_0, ..., p_m\}$ into the escrow.
- **Escrow**: The escrow is a smart contract responsible for holding deposits by Alice and payments by Bob.
- **Verifier**: The verifier detects malicious behavior of Alice. In practice, this role is fulfilled by a smart contract, a dedicated third party, or the user.

2.3 Assumptions

The verifier in the system is able to detect any faults by Alice and is able to prove that Alice was at fault. We further assume that the protocol utilizing Promise implements payments and deposits through a ledger that provides the functionalities as describes in, e.g. [6]. Also, there is a one to one mapping between the collateral and a user, i.e. the collateral of an intermediary is not split between multiple users. Last, agents in the system can be identified with their public/private key pair.

2.4 Utilities

Alice can either behave honestly or cheat, with the following payoffs one period ahead. V_A denotes the numerical value that Alice attaches to the act of cheating. V_B denotes the numerical value that Bob attaches to receiving the service. c denotes the cost of an individual transaction. $\mathrm{E}[r]D$ reflects the expected opportunity cost of locking the capital for one period.

$$\Pi_A = \begin{cases} p - \mathrm{E}[r]D, & \text{if Alice is honest} \\ V_A - \mathrm{E}[r]D - D, & \text{if Alice cheats} \end{cases} \tag{2}$$

$$\Pi_B = \begin{cases} V_B - p - c, & \text{if Alice is honest} \\ D - V_B - c, & \text{if Alice cheats} \end{cases} \tag{3}$$

Each round the game resets. Therefore, Alice will be honest iff $p > V_A - D$. Assuming that $V_B - p > 0$, otherwise Bob would not seek the service from Alice in the first place, he stands to gain utility if Alice behaves honestly.

3 Promise

In Promise we allow Bob to provide multiple payments in advance and delay the receipt of the payments to Alice. In turn, Alice is able to reduce the initially provided collateral from D to D_I such that $D_I < D$. At $t = 0$ Bob is able to lock m future payments $\{p_0, ..., p_m\}$ in escrow and determine a period τ after which Alice can receive the payments. When $t < \tau$, Alice continues to accumulate collateral as time passes by keeping the cumulative total of her payments p_i in escrow. We provide an intuition in Fig. 1. This has the following advantages for Alice and Bob.

Alice: the barrier to entry as a service provider is lowered, as even in the first period Alice only needs to offer D_I as opposed to D.

Bob: the aggregation of multiple payments allows Bob to reduce transaction costs and guarantess Bob that he only pays Alice if she fulfills all tasks for the given period m.

To argue about the security of Promise, we introduce two concepts: (i) temporality and (ii) a likelihood of users exiting the system upon the counterparty cheating.

3.1 Introducing Time

Denoting time by t, we define the period for which Bob locks Alice's payments as τ, such that if Bob makes a payment every period t then $\tau = m$. Let m denote the number of payments that Bob makes into escrow. Discounting is also introduced, where $0 < \delta < 1$ denotes the discount factor of an agent's valuation of future utility, i.e. the notion that the future is worth less than the present. We argue that an agent can spent received payments somewhere else or potentially invest the payment for a profit. Hence, the agent faces an opportunity cost for delayed payments. The payoffs to Alice and Bob are as follows.

$$\Pi_A(t) = \begin{cases} \left(\frac{\delta}{1+r}\right)^m pm - D_I((1 + \mathrm{E}[r])^m - 1), & \text{if Alice is honest} \\ \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t (V_A - \mathrm{E}[r]D_I - D_I), & \text{if Alice cheats} \end{cases} \tag{4}$$

Bob receives the following payoff, depending on Alice's behavior.

$$\Pi_B(t) = \begin{cases} \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t (V_B - p - \mathrm{E}[r]p(m-t)) - c, & \text{if Alice is honest} \\ \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t (D_I - V_B - c), & \text{if Alice cheats} \end{cases} \tag{5}$$

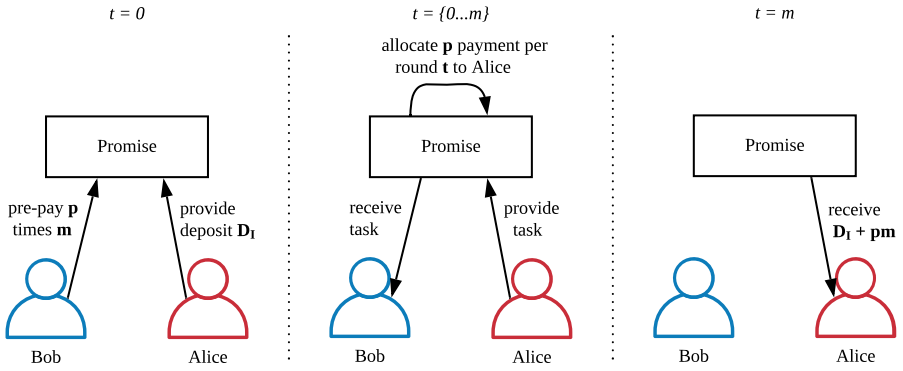

Fig. 1: Promise allows intermediaries (Alice) to lock less initial deposit D_I and use payments p_i provided by users (Bob) as additional deposit. The initial deposit and payments are locked for a period of time m determined by Bob. Only when Alice behaves honestly for the entire period m can Alice withdraw her initial deposit D_I and the payments pm.

3.2 Termination Probability

Lowering Alice's initial collateral to D_I increases the risk of Alice cheating Bob. Specifically, in the first round, Alice's collateral is the lowest since she has not provided the service yet and has not added any payment into her collateral pool.

Eq. (2) and (4) implicitly assume that Alice and Bob are playing a one-shot game ad infinitum. We argue that Bob exits a protocol after being cheated too often, encoded in the function $\beta \rightarrow [0, 1)$ describing the likelihood that Bob remains in the protocol. Each time Bob gets cheated, n_c, the lower the probability that Bob continues to participate. Each user can have its own β function where users might choose to never participate again in a protocol after being cheated once and others might tolerate a higher number of incidents. This changes Alice's payoff for the protocol.

$$\Pi_A(t) = \begin{cases} \left(\frac{\delta}{1+r}\right)^m pm - D_I((1 + \mathrm{E}[r])^m - 1), & \text{if Alice is honest} \\ \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t \beta(V_A - \mathrm{E}[r]D_I - D_I), & \text{if Alice cheats} \end{cases} \tag{6}$$

As β decreases, the payoff to Alice converges to 0 as $n \rightarrow \infty$. For Alice, we increase the motivation to behave honestly by (i) providing a sum of payments mp that Bob locks in the protocol, and (ii) the fear of Bob leaving the protocol altogether if she cheats. As Bob chooses m and τ, he has a direct influence on Alice's expected payoff. By setting large m and τ and being able to quit the protocol upon cheating, he can essentially force rational Alice to behave honestly. Overall, Alice can join protocols with less initial collateral while providing Bob with the option to use ongoing payments as an additional security measurement.

4 Security Analysis

Promise aims to fulfill the following properties:

1. **Security of funds**: Payments p (future and current) provided by a user as well as deposits D by intermediaries cannot be stolen. This property is fulfilled by implementing the escrow as a smart contract on a ledger with a functionality and appropriate security parameters as described in [6,7].
2. **Cost reduction for users**: Bob is able to reduce transaction costs c by paying for multiple rounds m of services for payment p in advance.
3. **Collateral reduction for intermediaries**: Alice is able to provide a lower deposit D_I than the deposit required in a single-shot game setting D while Bob enjoys the same level of security against Alice.
4. **Sybil resistance**: It is not individually rational for Alice to increase her payoff Π by cheating Bob in one round and provide the service honestly in the next rounds.

4.1 Cost Reduction for Users

Assume that Alice behaves honestly. If a user pays every round t for the service provided by Alice, then his payoff per round is $V_B - p - c$. However, locking multiple payments incurs opportunity cost. This cost is lowered at every time step as the payments are assigned to the intermediary. Hence, Bob starts with an opportunity cost of $\mathrm{E}[r]pm$ at $t = 0$ and $\mathrm{E}[r]p(m-1)$ at $t = 1$. Generalizing this for t rounds, leaves us with $\mathrm{E}[r]p(m-t)$. The user locks future payments when the sum of the transaction costs c for m payments is greater than the opportunity cost for locking additional payments plus the single transaction cost for making the prepayments. Hence, the boundary for a user to choose Promise as individually rational choice maximizing his payoff is given by:

$$\sum_{t=1}^{m} \big(\frac{\delta}{1+r}\big)^t c = \sum_{t=0}^{m} \big(\frac{\delta}{1+r}\big)^t \mathrm{E}[r]p(m-t) \tag{7}$$

4.2 Collateral Reduction and Sybil Resistance

We can calculate the decision bound for Alice's individually rational choice to cheat Bob by comparing the two options in Eq. (6). We can re-arrange this equation to determine β:

$$\beta = \frac{\big(\frac{\delta}{1+r}\big)^m pm - D_I((1+\mathrm{E}[r])^m - 1)}{\sum_{t=0}^{m} \big(\frac{\delta}{1+r}\big)^t (V_A - \mathrm{E}[r]D_I - D_I)} \tag{8}$$

Similarly, Bob can determine how high D_I should be under the assumption that he knows V_A and the parameters r and δ.

$$D_I = \frac{\left(\frac{\delta}{1+r}\right)^m pm - \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t \beta V_A}{((1+\mathrm{E}[r])^m - 1) - \sum_{t=0}^{m} \left(\frac{\delta}{1+r}\right)^t \beta(\mathrm{E}[r]-1)} \tag{9}$$

Assuming that β is lowered aggressively, i.e. Bob tolerates only few task violations n_c, Alice chooses to be honest as this maximizes her payoff. Being honest with a reduced collateral D_I is thereby the individually rational choice. Given that Bob is aware for the involved parameters V_A, r, and can set the parameters p and m himself, Bob is able to calculate at which likelihood he should exit a given protocol given by Eq. (8). Similarly, Bob can decide which level of colalteral D_I is sufficent for him using Eq.(9).

Sybil Identities We argue that if β is lowered aggressively, Alice can do no better than being honest. However, if there is a non-negligible probability that Bob remains in the protocol after being cheated once, Alice can try to create a Sybil identity A' to (i) cheat Bob in round $t = 0$ and (ii) behave honestly in round 1 to $m + 1$. Hence, Bob can only be sure to prevent Alice from executing a Sybil attack, if $\beta = 0$.

4.3 Beta Factor Over Time

The β factor set by Bob is the crucial security parameter for Bob given we are in a permissionless system. We argue that setting the β factor depends on Bob's reliance on participating in a protocol. If Bob *requires* the service Alice provides and there is *no alternative* to the protocol in which Alice operates, Bob is faced with a *monopoly*. In a monopoly Bob β likely never approaches 0 as Bob is forced to tolerate Alice's misbehavior. To protect Bob, Promise requires that D_I of Alice is close or equal to D in a monopoly.

The opposite market situation is *perfect competition*. Bob can choose from a wide selection of service providers to interact with[2]. Assuming perfect competition exists, Bob has the choice to leave a protocol any time. As an example, we could imagine Bob choosing between using a payment hub like NOCUST or a payment channel network like Lighting to obtain scalable and cheap payments. For simplicity of argument, we assume that Bob is indifferent if he pays in Bitcoin or Ether. If Bob would be cheated on by an intermediary in NOCUST, he could switch to using the Lightning Network. In turn, Bob can set β to 0. In these cases we can substantially lower the initial collateral D_I using Eq. (9).

5 Related Work

There are two strands of related literature. The first one comes from the financial world covering (advance) payments for financial contracts. The second

[2] However, as we lack strong identities, these service providers could all be Sybil identities of Alice. Recognizing that perfect competition exists is a challenging task and we leave this as future work.

strand comes from the more recent work in decentralized ledgers. In the economics literature, a wide range of work focuses on secured debt, such as [11,12]. However, these concepts rely on trust on third parties to maintain security in the debt and payment positions. Promise replaces this third-party trust by holding advance payments in a smart contract escrow.

On the second strand, Balance is a protocol that allows intermediaries to lower their collateral over time [8]. It operates at the other end of Promise: instead of lowering the initial collateral, the more an agent behaves honestly, the higher the reduction of collateral. Promise and Balance can be combined together to first reduce initial collateral when bootstrapping a new protocol and then lower collateral requirements for established agents over time. Teutsch et al. discuss bootstrapping a token for verifiable computations [13]. Their proposal includes a governance game that allows to exchange special governance tokens into collateral tokens (for intermediaries) and utility tokens (for users). Last, the idea of bundling payments together is also introduced in [2] to create subscriptions for services of agents. Promise extends this idea to allow collateral reduction for intermediaries.

6 Conclusion

We present Promise, a system that allows intermediaries to lower the initially locked collateral. The core assumption for the security of Promise is that a user Bob is able to lock a number of payments up front and exit the protocol when Alice cheats. This allows Alice to reduce her initial collateral requirement, allowing protocols that adopt Promise to lower the burden on intermediaries. Bob is able to reduce his transaction cost as he transfers a sum of payments. Further, Bob is able to receive the service in full — for the entire period he pre-paid — or he is refunded the entire sum of payments. We have introduced a semi-formal model for Promise. We discuss the security and the effect of the β parameter, but leave formal proofs of the security properties as future work.

References

1. Aiyer, A.S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.P., Porth, C.: Bar fault tolerance for cooperative services. In: ACM SIGOPS operating systems review. vol. 39, pp. 45–58. ACM (2005)
2. Berg, P.R.: ERC-1620: Money Streaming (2018), `https://github.com/ethereum/EIPs/issues/1620`
3. Böhme, R.: A primer on economics for cryptocurrencies (2019)
4. Dziembowski, S., Faust, S., Hostáková, K.: General state channel networks. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. pp. 949–966. ACM (2018)
5. Fanti, G., Kogan, L., Oh, S., Ruan, K., Viswanath, P., Wang, G.: Compounding of Wealth in Proof-of-Stake Cryptocurrencies. In: Financial Cryptography and Data Security 2019 (2019)

6. Garay, J.A., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol with chains of variable difficulty. http://eprint.iacr.org/2016/1048.pdf (2016), accessed: 2017-02-06
7. Gervais, A., Karame, G.O., Wüst, K., Glykantzis, V., Ritzdorf, H., Capkun, S.: On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 3–16. ACM (2016)
8. Harz, D., Gudgeon, L., Gervais, A., Knottenbelt, W.J.: Balance: Dynamic Adjustment of Cryptocurrency Deposits. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS '19). ACM, New York, NY, USA (2019), https://eprint.iacr.org/2019/675.pdf
9. Khalil, R., Gervais, A., Felley, G.: NOCUST - A Securely Scalable Commit-Chain (2019), https://eprint.iacr.org/2018/642
10. McCorry, P., Bakshi, S., Bentov, I., Miller, A., Meiklejohn, S.: Pisa: Arbitration outsourcing for state channels. IACR Cryptology ePrint Archive **2018**, 582 (2018)
11. Scott Jr, J.H.: Bankruptcy, secured debt, and optimal capital structure. The Journal of Finance **32**(1), 1–19 (1977)
12. Stulz, R., Johnson, H.: An analysis of secured debt. Journal of Financial Economics **14**(4), 501–521 (1985)
13. Teutsch, J., Mäkelä, S., Bakshi, S.: Bootstrapping a stable computation token (2019), http://arxiv.org/abs/1908.02946
14. Zamyatin, A., Harz, D., Lind, J., Panayiotou, P., Gervais, A., Knottenbelt, W.J.: XCLAIM: Trustless, Interoperable, Cryptocurrency-Backed Assets. In: Proceedings of the IEEE Symposium on Security & Privacy, May 2019. pp. 1254–1271 (2019), https://eprint.iacr.org/2018/643.pdf