SPIN, a stochastic micropayment protocol

Pablo Mansanet
Silkweave
Email: pmansanet@tendril.net

***Abstract*—Scaling blockchain technologies to supply the needs of applications involving frequent, low value transactions remains an open question. We propose a novel stochastic micropayment system, where a high number of small transactions are simplified into a smaller sample of on-chain exchanges, which captures the same economic activity with precision adaptable to each user's preference. We prove that honoring SPIN transactions constitutes a Nash equilibrium, and that under the correct network assumptions it is safe to employ SPIN for immediate, zero-confirmation transfers of value.**

I. Introduction

Distributed consensus, as it is obtained in blockchain platforms such as Bitcoin or Ethereum, requires notifying a multitude of (generally all) network participants of every single change of state. Leaving aside the economic cost of validating those changes through Proof of Work, Proof of Stake or an equivalent mechanism, there will always be a cost associated to the synchronized update of redundant information across thousands of machines.

Currently, the capacity of all major blockchain technologies to process transactions is much lower than that of centralized electronic payment system such as Visa, by orders of magnitude[3]. This inefficiency manifests for the end user in the form of transaction fees, that they must supply for validators (e.g. miners) to prioritize their transaction to be registered. As of the time of writing, Bitcoin transactions incur an average fee of 2 USD[1], and 0.22 USD[2] in the case of Ethereum.

Individual blockchain differences aside, these transaction fees make entire areas of trade infeasible on a blockchain system. Vending machines are a simple example we will explore in this document, for which transaction fees of that magnitude are unjustifiable. For systems where transfers of value are that small, it is also difficult to justify waiting until full economic finality is achieved, which makes zero-confirmation transactions attractive but unsafe.

There are some proposed solutions to the scaling issues, such as off-chain transaction channels[3][4] and on-chain sharding[5]. While they may suffice for some transactions, they fall short of providing the necessary flexibility for the most demanding markets.

A relevant example is the Tendril protocol, developed in tandem with SPIN to serve as a decentralized marketplace for network connectivity. The nature of Tendril demands for participants to interact at extremely high frequency and exchange infinitesimal value, down to the level of one IP packet. When the value exchanged is significantly below 0.01 USD, no degree of sharding justifies transaction fees, yet the economic transfer must be captured. Furthermore, Tendril participants may switch providers often (for example when roaming) which precludes establishing time-locked payment channels.

II. SPIN - Stochastic Payment for Immediate Negotiation

We propose SPIN, a novel micropayment system that allows capturing infinitesimal exchanges of value at a very low cost, by stochastically sampling the set of possible transactions and only relaying a subset to the blockchain, with a dynamic probability agreed upon by participants and a proportionally higher value.

We rely upon solid game theory arguments to prove that SPIN transactions carry real economic value even in the instances where they are not relayed to the blockchain, and we propose safety mechanisms to guarantee strategyproofness, protection against double spending and other economic attacks.

A. Stochastic Payment Basics

The driving force behind SPIN is the concept of stochastic payment. A stochastic payment consists of a cryptographically signed transaction that is exchanged off-chain between payer and payee, to be redeemed by the recipient. Where SPIN first departs from usual blockchain transactions is in this inversion of control; the signed transaction is sent directly to the recipient, who is then in charge of redeeming the transaction by relaying it to the blockchain.

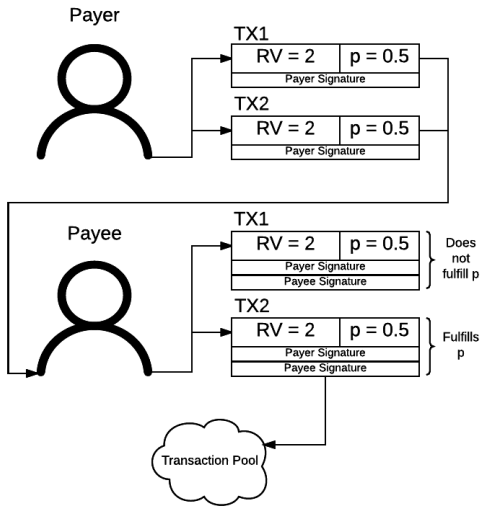

Fig. 1: Two stochastic transactions valued V=1 each

The second and most striking difference is the conditions under which this redeeming takes place. The recipient of a SPIN payment is not always capable of redeeming it. Instead, each transaction has an associated, agreed upon probability p. A transaction may only be redeemable if the SHA-256 hash digest of a transaction signed by both participants, interpreted as a 256 bit unsigned integer is higher than $p * (2^{256} - 1)$.

A transaction of redeemable value RV, with a redeeming probability p has a stochastic value of $V = RV * p$, and may in good faith be exchanged for a good of service of value V. While it may seem unintuitive, the justification why a recipient should accept a stochastic payment in exchange for a concrete good or service follows from strong game theory arguments and will be proven in the following sections.

Stochastic payments allow for a drastic reduction on the number of transactions required to capture economic activity. By automatically negotiating a redeeming probability p that satisfies both parties' tolerance for variance, through an heuristic that will be explored later, a series of N on-chain payments is instead reduced to $p*N$ samples of proportionally higher value. Since transaction fees only apply to redeemable transactions, the aggregated transaction costs are reduced by a factor of p. Furthermore, these reduced costs are shouldered in proportion to each participant's variance tolerance.

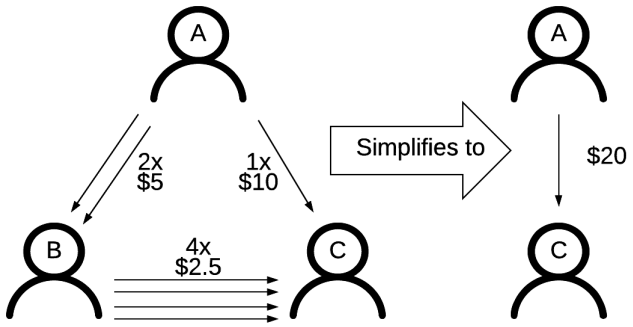

Fig. 2: A possible (ideal) transaction simplification

It can be hard to believe that a participant would accept a 'lottery ticket' as a payment for a concrete good or service, in particular when the recipient can immediately verify whether or not the transaction is redeemable. The idea that a payment is only occasionally valid, yet it always constitutes a transfer of value is difficult to grasp. In the following sections we will show properties that make the system more acceptable at an intuitive level.

B. Blind Redeemability

In order to decide whether a particular payment is redeemable, both parties must sequentially sign it with their respective private keys. As the payer sends the payee the signed transaction, he has irreversibly agreed to pay in case the transaction is redeemable. This transaction is then signed with the payee's private key, and the result is SHA-256 hashed and matched against the probability parameter p. Because of this, the payer is unable to deduce whether any particular transaction will be redeemable, since that would require knowledge of the recipient's private key. Therefore, the payer is incapable of selectively withholding payment. Once a redeemable payment is in the hands of the recipient, it can be sent to the blockchain without the need for further action from the payer, as it already contains their private signature, and so the payer cannot prevent it from being processed.

C. Deviation Tolerance

Variance in the distribution of payments is inversely proportional to the redeeming probability, so the lower p gets, the more imprecise SPIN payments become at reflecting the actual exchanges of value between participants. It is important to understand that SPIN is designed to fulfill the need for frequent, low value payments, around or below the order of magnitude of a blockchain transaction fee. It is therefore a reasonable assumption that the quantities are significantly lower than each participant's purchasing power, such that it would be possible to set a value p that results in meaningful transaction cost savings while remaining within acceptable variation boundaries for both participants.

Participants in a network powered by SPIN payments are likely to display different levels of deviation tolerance. While a user may be comfortable with a standard deviation of $10 a month in exchange for significantly reduced fees, as long as their economic activity averages out to the same result, another user may prefer to keep a tighter control over their finances even if it results on higher transaction fees. Fortunately, SPIN is designed to automatically accomodate any pair of user profiles. The fundamental principle behind this mechanism is that *transaction fees are split in inverse proportion to the participants' variance tolerance*. In fact, SPIN operates normally even in the exceptional situation that one user prefers not to accept any deviation at all, in which case p degenerates to 1 and the variance averse user is forced to cover the transaction costs alone.

D. Variance Negotiation

SPIN participants have a maximum deviation tolerance factor, expressed in terms of maximum redeemable value MRV; that is, the maximum amount of currency they are willing to exchange in a single SPIN transaction. Users set their own MRV on the blockchain, and it will always be bounded from above by their total funds.

For a transaction of stochastic value V, their minimum desired probability p is automatically derived as $p^{min} = V/MRV$. Since each participant's MRV is publicly visible, they are aware of the other party's minimum probability p^{min}.

As participants 1 and 2 establish a transaction, the highest $p \in \{p_1^{min}, p_2^{min}\}$ is automatically chosen as the redeeming probability. Let T be the blockchain transaction fee, which is calculated fairly and publicly from visible blockchain metrics (e.g. fees for last block); the expected cost for a stochastic payment is $T^e = T * p$. This expected cost is split among participants according to the formula:

$$T_1^e = T * p * \frac{p_1^{min}}{p_1^{min} + p_2^{min}}$$

$$T_2^e = T * p * \frac{p_2^{min}}{p_1^{min} + p_2^{min}}$$

$$T^e = T_1^e + T_2^e$$

If T_1^e is the fee fraction corresponding to the payer, he must add the value $\frac{T_1^e}{p}$ to the transaction's redeemable value, to represent their contribution to the shared fees. The remainder of the fee will be covered by the payee when/if the transaction is relayed to the blockchain.

An interesting consequence of this negotiation format is that it *decouples the user's maximum fees from their counterparty's profile*. In other words, it is possible for a user to establish an arbitrarily small upper bound on their transaction fees by offering a sufficiently low p^{min}.

$$\forall p_j^{min}, \quad T_i^e < p_i^{min} * T \tag{1}$$

In a way, users are bidding with their deviation tolerance to obtain a favorable transaction fee split.

III. Game Theory Soundness

As we mentioned briefly during the introduction, accepting a stochastic payment as possessing some value can be very unintuitive, especially after a particular transaction is found to be irredeemable. Because of this conceptual difficulty it is unlikely that applications built upon SPIN will expose the randomness of the protocol at the user level. Instead, outgoing and incoming payments will be aggregated and presented to the user in a weekly or monthly digest, with each concrete exchange being locally shown as "successful" regardless of redeemability.

Assuming that the redeeming procedure is hidden from the user's view, and that the user's awareness of the stochastic nature of the payment is limited to parameters such as variance over weekly/monthly expenditure plotted against savings, the psychological barrier is removed. What remains then is to prove that a malicious user does not stand to gain by abusing the protocol in any way, or by choosing to selectively withhold service after receiving a stochastic transaction, redeemable or otherwise. What follows is a justification for the claim that an irredeemable transaction does in fact possess economic value, and that honoring the transaction constitutes a Nash equilibrium under the correct economic assumptions.

A. One shot SPIN transaction game

Let us model a SPIN transaction as a sequential game of incomplete information with player set $P = \{1, 2\}$ (1-Buyer, 2-Seller). Player 1 moves first and chooses whether to trust player 2 (T) by emitting a stochastic transaction, or not to trust (N) and instead withdraw from the negotiation. Nature then chooses the redeemability status of the transaction with probability p, which was agreed upon by participants. Player 2 then chooses whether to honor the transaction and cooperate (C), providing the requested service or good, or to defect (D), withholding the service or good.

Let v be the subjective value of the good or service from the buyer's perspective. Let c be the cost of producing that good or service for the seller. In both cases, let us assume for simplicity that the transaction fees are already factored in those variables. Let f be the agreed upon price of the good or service. For the exchange to be economically sound, we assume that $v > f > c$. Let p be the probability of the transaction to be redeemable.

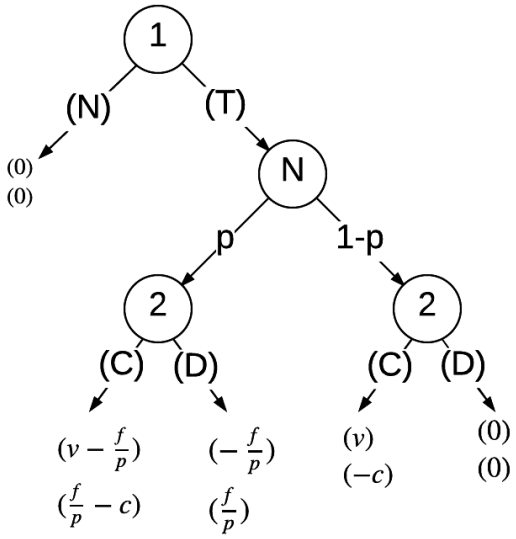

Fig. 3: One shot SPIN transaction game in extensive form

From the extensive form diagram it is easy to see that player two's best response is to defect in every scenario. Player two's dominant strategy is to unconditionally defect, obtaining an average payoff of $\frac{f}{p} * p + 0 * (1 - p) = f$. Player one's best response to this strategy is to withdraw, imposing a payoff of 0 on both players and avoiding an average payoff of $-f$.

A single SPIN transaction's unique Nash equilibrium is for the buyer to withdraw trust and for the seller to inconditionally defect. This apparently bleak conclusion is not that surprising if we compare it to other common transactions modeled in a similar way. Any sequential exchange of value where the second player is able to get away with the money at no consequence will lead to the same result.

Day to day transactions depend upon social norms and external enforcement to force participants to abide to the rules, often through the threat of punishment. While this scenario can be replicated to some degree at the protocol level through a reputation system, we can obtain stronger guarantees simply by making some assumptions about the market.

B. Infinitely repeated SPIN transaction game

Let us assume that there is some probability that a successful exchange will be followed by another exchange between the same participants in the future. Let $\delta < 1$ be the economic discount factor that takes into account this probability as well as the time value of money. Let us model the SPIN transaction as an infinitely repeated game, which results in the total expected payoff $E_i = \sum_{t=1}^{\infty} \delta^t * v_i^t$, where v_i^t is the expected payoff for player i in stage t.

Let h_t be the history of moves by both players up to stage t, such that $h_t = ((s_1^1, s_2^1)(s_1^2, s_2^2), ..., (s_1^{t-1}, s_2^{t-1}))$. We propose the following grim trigger strategy, where both

participants collaborate as long as neither of them deviates from collaborative behaviour:

$$s_1^1 = T$$

$$\forall t > 1, \quad s_1^t(h_t) = \begin{cases} T & h_t = ((T,C)...(T,C)) \\ N & h_t \neq ((T,C)...(T,C)) \end{cases}$$

$$s_2^t(h_t, s_1^t) = \begin{cases} C & s_1^t = T \wedge h_t = ((T,C)...(T,C)) \\ D & s_1^t = N \vee h_t \neq ((T,C)...(T,C)) \end{cases}$$

Abiding to this collaborative strategy results on the following average payoff at each stage:

$$\forall t, \quad v_1^t = (v - \frac{f}{p}) * p + v * (1-p) = v - f \tag{2}$$

$$\forall t, \quad v_2^t = (\frac{f}{p} - c) * p - c * (1-p) = f - c \tag{3}$$

Therefore, the total expected payoff becomes:

$$E_1 = \sum_{t=1}^{\infty} \delta^t * (v - f) = \frac{v - f}{1 - \delta} \tag{4}$$

$$E_2 = \sum_{t=1}^{\infty} \delta^t * (f - c) = \frac{f - c}{1 - \delta} \tag{5}$$

If the seller does not deviate, the buyer is playing a best response since the alternative (withdrawing) achieves an average payoff of 0, and we have established that $v > f$ so $\frac{v-f}{1-\delta} > 0$. On the other hand, the seller is playing a best response as long as her average payoff exceeds the immediate benefit of defecting:

Remark. *Honoring the trade is a pareto-optimal, subgame perfect Nash equilibrium if* $\frac{f-c}{1-\delta} > f$.

Note that the conditions for cooperation to be viable are independent of the transaction's redeeming probability p. Intuitively, we can see that the seller's choice to cooperate is independent of the redeemability state of the last transaction received. After the seller has received a redeemable transaction, she already has access to the funds regardless of whether she cooperates; the motivation for her choice is the prospect of future profit from dealing with the same buyer again.

This last observation allows us to prove that the Nash equilibrium is subgame-perfect. If we consider the subgame involving the seller's decision after the transaction has been confirmed redeemable, we have the following total expected payoffs:

$$E_2^{cooperate} = \frac{f}{p} - c + \frac{f - c}{1 - \delta}$$

$$E_2^{defect} = \frac{f}{p}$$

Cooperating is a best response when $\frac{f}{p} - c + \frac{f-c}{1-\delta} > \frac{f}{p}$, that is, $\frac{f-c}{1-\delta} > c$, and from the above remark and the definitions of f and c we know that $\frac{f-c}{1-\delta} > f > c$. Similarly, for the seller subgame where the transaction is not redeemable, the total expected payoffs are:

$$E_2^{cooperate} = \frac{f - c}{1 - \delta} - c$$

$$E_2^{defect} = 0$$

Again, cooperating is a best response when $\frac{f-c}{1-\delta} > c$, so we prove by exhaustion that the Nash equilibrium is subgame-perfect. Here is where we substantiate the claim that an irredeemable transaction carries value; even if a particular transaction did not result in an actual transfer of wealth, the seller can verify that it was was valid and have an incentive to obtain more.

C. Modeling implicit seller reputation

While the previous result offers a powerful guarantee in case of repeated transactions between the same payer and payee, we may strengthen the result even more if we assume that the particular SPIN powered market can enforce cooperation through some external mechanism.

This difference is best explained through an example. Imagine the case of a network of vending machines that use SPIN to process payments. A user wants to purchase chewing gum for the price of $f = \$0.05$. The total cost of delivering the product is $c = \$0.02$. If we substitute in expression 5 we find that for the vending machine to prefer honoring the trade, the probability to encounter the same buyer again must be higher than 40%. Depending on where this vending machine is located this may not be an easy ask. For example, a vending machine stationed at an airport is unlikely to deal with the same customer multiple times, so non-cooperative behaviour becomes the only Nash equilibrium.

Fortunately for us we can tap into reputation and accountability to shift the balance back into cooperative behaviour. A vending machine that consistently scams its users will in one way or another be removed from the market. Either a user will make it known to the machine administrators until they hang an "out of order" sign, or in case the administrators refuse to comply, users will communicate and inform each other that they should avoid that particular machine.

Knowing that, we can modify our analysis slightly: let δ be the economic discount value, which factors in the time value of money and the probability of dealing with **any** other customer in the future, in case the seller honors the trade. Let ϵ be the similarly formulated economic discount value, in case the seller does **not** honor the trade. We should assume that a user that was just scammed will not continue to interact with the seller, so this value reflects the probability of that user affecting the seller's future profits through reputation impact (e.g. complaining to the staff). Going back to our strategy analysis, if we consider the strategy where the seller consistently defects, he obtains the following expected payoff:

$$E_2 = \sum_{t=1}^{\infty} \epsilon^t * f = \frac{f}{1 - \epsilon} \tag{6}$$

Comparing that payoff with the collaborative payoff (5), we find that honoring the trade is a pareto-optimal, subgame-perfect Nash equilibrium when $\frac{f-c}{1-d} > \frac{f}{1-e}$. We omit the proof for subgame perfection, since it can be constructed identically to the single customer case.

Going back to the airport vending machine example, we can estimate $\delta \approx 0.99$, since it is very likely that a working vending machine will interact with further customers, and $\epsilon = 0.95$, if we generously assume a 5% chance that a dissatisfied customer will force the machine to be labeled out of order. With these values we can calculate the expected payoffs from the two pure strategies:

$$E_2^{cooperate} = \frac{0.05 - 0.02}{1 - 0.99} = 3$$

$$E_2^{defect} = \frac{0.05}{1 - 0.95} = 1$$

$E_2^{cooperate} > E_2^{defect}$, so the machine vendor is incentivized to cooperate and honor the trade, even when dealing with the same customer repeteadly is unlikely.

Every particular SPIN application will have to approach its marketplace differently; possibly by combining both models. For example, in the case of Tendril, SPIN's sister protocol, there is no need for external enforcement, since by nature of the economic activity itself (paying at a packet level for network connectivity) buyers and sellers are bound to interact repeteadly hundreds or thousands of times, so defecting for immediate gain is never justifiable.

D. Modeling failed payment and loss of communication

When modeling the one-shot SPIN transaction game, we defined the buyer's move (N) as an immediate withdrawal from negotiation. In case a seller refuses to produce a good or service, it is reasonable to expect the buyer to withdraw permanently. However, it is possible that the seller didn't receive the transaction correctly, or that she failed to produce the good for extraneous reasons. An example of this situation is packet loss in the Tendril network; a genuine relay could fail to deliver a packet because the destinatary disconnected, or because the stochastic transaction itself was corrupted in transit to the relay.

In case of an unfulfilled service, the buyer may choose to establish a new trade by emitting the same transaction again. Since it is identical to the last one, no actual transfer of value takes place. If the previously sent transaction was not redeemable, the current one would not be redeemable either, and if it was, they would both be redundant. In case the buyer wants to request a different service priced higher than the failed one, he only needs to produce two transactions; the redundant one and a second transaction making up for the difference. In either case, the seller will be aware it failed to produce the good or service and will accept the redundant payment.

Knowing this, we can model player one's choice (N) not as an immediate withdrawal from negotiation, but as a stream of attempts to pay with the same redundant transaction. This also results on an average payoff of 0 for both participants but reflects a more realistic scenario in low value transactions with user retries (e.g. a customer repeteadly inserting a rejected coin in a vending machine).

IV. SPIN and settlement finality

We define *settlement finality*[6] as the degree of confidence that a transfer of value can be finalized, even in the face of insolvency problems by any participant. Settlement finality means that once a transaction is completed, the recipient can be sure the funds are "theirs" and that the operation can not be taken back.

Centralized institutions like banks guarantee settlement finality through legal mechanisms, which ultimately reduce to the credible threat of punishment. Since blockchain technology is by nature pseudonymous, it is not capable of enforcing good behaviour in that way. Instead, the blockchain offers a form of *probabilistic* settlement finality backed by cryptography and game theory, where confidence in a transaction's finality grows with the length of the chain containing it.

Leaving aside the general risks of such a system (e.g. 51% attacks), blockchain technologies offer a form of settlement finality that is safe enough for most commerce. The most obvious attack a malicious participant could attempt is a *double spend*[7][8] attack, where the attacker would emit conflicting transactions, moving the same funds to two or more recipients, one of them potentially their own alias.

To protect herself from this attack, a seller only needs to wait for a sufficient amount of blocks to be confident that her legitimate transaction has been prioritized, and then produce the good or service for the payer. This guarantee is explored in detail in the original Bitcoin paper[8], since it is the main problem that Bitcoin was designed to address.

For markets where the time between exchanging currency and goods must be shorter than the confirmation time, the problem becomes significantly more complicated. The safety of zero-confirmation transactions is still subject of research[7] beyond the scope of this text.

SPIN's relation to settlement finality is more nuanced. If we take it to the degenerate extreme and assume $p = 1$, every transaction is on-chain and finality is guaranteed in the same way, by waiting for sufficient confirmation. However, any $p < 1$ causes finality issues; every time a seller is accepting a non-redeemable transaction as having real value, he is implicitly trusting the payer as not having the means and intention to revert a redeemable transaction in the future.

A. Listener double spend attack

Imagine Alice is a malicious participant in a SPIN powered network that intends to steal from Bob, and that she possesses the technical means to mount a double-spend attack. Alice and Bob agree on a redeeming probability $p = 0.1$ and anticipate to perform 10 trades. The first five transactions are not redeemable, but Bob produces the requested good as per the protocol. However, when the sixth transaction turns out to be redeemable, she preempts it by posting another which is prioritized by miners/stakers and moves the entirety of her funds to an alias account. Bob's legitimate transaction then gets rejected due to lack of funds in Alice's account.

This attack is harder to orchestrate than a traditional double-spend, since it requires Alice to notice Bob's transaction being uploaded, but still have time to preempt it. It

is nevertheless possible if the seller is badly connected or Alice can somehow intercept unencrypted information packets emitted by Bob.

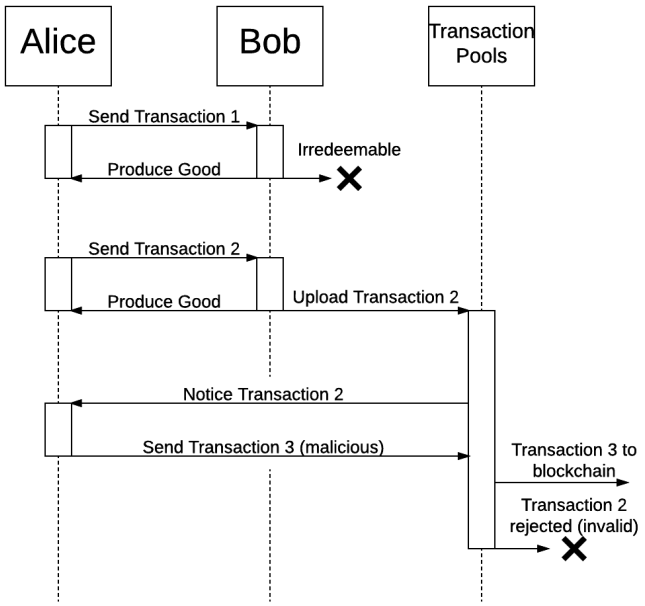

Fig. 4: Listener double spend attack sequence

We do not need to invoke game theory to see that this scenario is very profitable for Alice. While Bob may avoid the attack for that precise transaction, he would have been already scammed for every unredeemed transaction before it. We call this a *listener double spend attack*, because it requires the attacker to monitor the pools to detect the redeemable transaction. An aspect of this attack is fundamental to the payment mechanism and cannot be avoided; an indefinite (inversely proportional to p) number of transactions occur before settlement finality. However, while we cannot completely prevent it, we will see that we can effectively neutralize the attack by removing all economic incentive to perform it.

In the context of SPIN, we define **loss finality** as the degree of confidence that a transference results in the payer losing control over the transferred funds. This definition must be Sybil-proof; preempting a transaction by sending the funds to a third account instead does not constitute loss finality since the recipient may be another alias of the payer. If we are able to guarantee loss finality, we make sure an attacker does not stand to gain from removing funds before a legitimate transaction is finalized.

B. ADSC - Automatic Double Spend Correction

We propose a mechanism to automatically enforce loss finality, curiously by relaxing the double spend protection present in most mining pools. Conflicting SPIN transactions (that is, transactions which would result in transferring the same currency to multiple recipients) are valid as long as they are not further than some number N_{adsc} of blocks apart, for example $N_{adsc} = 2$. When two or more conflicting transactions are registered in the blockchain within said block distance N_{adsc}, the contested funds are removed from **every** beneficiary and redistributed as miner/staker fees.

Example: Alice produces a redeemable transaction to pay Bob \$5. Simultaneously and before Bob's payment is processed, Alice transfers \$10 to Carol. Since Alice's account only contains \$12, there are \$3 in contested funds. As both payments are added to the blockchain, the contested funds are "burned" from both recipient accounts and transferred to the miners, which results in a balance of \$7 for Carol, \$2 for Bob and \$0 for Alice. The block miners receive \$3, which means the correction process is a zero-sum game.

ADSC is comprehensive and retroactive. The moment an offending transaction is first added to the blockchain, **every** participant who received funds from that payer recently has their balance reduced by the contested value, which may trigger ADSC on transactions emitted by themselves, but never affect history earlier than N_{adsc} blocks in the past.

ADSC effectively counteracts the listener double spend attack, by guaranteeing that a malicious payer will not be able to retain control of the funds by preempting a legitimate transfer. The fraudulent participant can't stand to gain from this practice and will in fact lose money to the transaction fees required to perform the attack.

This mechanism does not make the attack impossible; instead it removes all economic incentive to perform it. Seeing as the attack has other nontrivial costs, (mounting a double spend attack is difficult, and it becomes significantly harder when the payer is not immediately aware of whether a transaction is redeemable) it is extremely unlikely it will take place for the kind of market SPIN is designed for.

It is nonetheless possible for a determined attacker to cause economic damage to a seller by "burning" a particular payment at their own expense. However, the average extent of this damage is the seller's MRV, since any transactions with value higher than that would have been performed with $p = 1$, and performing this attack immediately and publicly flags the buyer as untrusted, unless they repay the debt by injecting more funds in the account, which would be immediately relayed to creditors.

While ADSC causes some short term uncertainty over the stability of received funds (since they may be burned), that uncertainty is short lived; after N_{adsc} blocks have elapsed, no further transactions may trigger a double spend correction. Conversely, the recipient is responsible for providing sufficient mining fees to stop the redeemable transaction from being "stuck" unconfirmed and thus extending that uncertainty. In particular, all transactions are timestamped and stop being valid after a comparatively longer number of blocks M_{adsc}.

Since the funds are distributed as miner fees in the case of conflicting transactions, miners have an incentive to spot and match these pairs, so it is not possible to perform a double spend attack by expecting a malicious transaction to be processed earlier than N_{adsc} blocks before the legitimate one. Even if the original transaction is stuck for hours, a conflict in the pool would force it to be registered on the blockchain alongside the conflicting transaction, due to the implicit higher reward for the miners. While convoluted, it is also worth pointing out there is no risk of collusion between miner and

buyer here, since the seller is responsible for choosing the transaction pool.

An exception to the ADSC rule is identical transactions. Two transactions with the exact same nonce, sender and recipient are by definition the same one, and are only valid the first time they are added to the chain. As a consequence of this, a transaction stuck in the pool may be republished with a higher fee to speed up its inclusion to a block without fear of triggering ADSC.

C. Real double spend attack

So far we have only explored one of the two main double spend attack "modes", in which one of the recipient accounts is an alias of the fraudster. When mounting a double spend in this way, the attacking transaction is not part of an actual exchange of value. ADSC eliminates the economic incentive for this type of double spend attack because it guarantees that none of the participants (seller, fraudulent buyer and their alias) end with control over the funds.

However, there is another form of double spend attack, where the attacker participates in two actual exchanges of value in conflict with each other. Instead of preempting a legitimate transaction with a transfer of funds to an alias account, in this scenario the attacker would purchase another good or service from a second seller.

For example Alice, a fraudulent buyer, would simultaneously request a good from Bob and Carol. If Bob and Carol both produce the good immediately after receiving their respective transactions, there is a chance (which could degenerate to 1 for higher value transactions) that both will turn out to be redeemable and trigger ADSC. This requires that neither Bob or Carol know of each other's transaction at the moment of producing the good.

ADSC does not eliminate the incentive for this type of double spend attack. However, we will see that it is extremely difficult to mount in practice, and impossible under very simple restrictions. What follows are two conditions that make the attack infeasible, as long as any of them hold.

Condition 1: The sellers involved are *patient*. Our definition of "patient" is any seller that is willing to wait until the blockchain confirmation time T regardless of whether the received transaction is redeemable. If all sellers are willing to wait until T, mounting the attack becomes impossible. In this case, we achieve safety at the cost of zero-confirmation transactions. This is trivial to explain when $p = 1$ (common blockchain case), but when either of the redeeming probabilities is $p < 1$ the analysis is slightly more complicated.

Let us first consider the case where Alice indiscriminately sends many transactions to multiple buyers, such that a substantial amount of them become redeemable and cause her to overspend. Let $R \times V = \{(r_1, v_1), (r_2, v_2), ..., (r_n, v_n)\}$ be the set of seller/value pairs where the seller transaction is redeemable, and $I = \{i_1, i_2, ..., i_m\}$ the set of sellers with irredeemable transactions. Let V_a be the total amount of funds in Alice's account before the attack.

Assuming appropriate miner fees and since T is an upper bound on the time necessary to guarantee finality (in particular, $T > N_{adsc}$), after some time $t < T$, transactions from $R \times V$ will start appearing on the chain. Let us assume without loss of generality than they appear in the order we defined them. By construction of the attack model, we know there will be an index K such that:

$$v_1 + v_2 + ... + v_{K-1} <= V_a$$

$$v_1 + v_2 + ... + v_K > V_a$$

Index K corresponds to the first offending transaction, added to the blockchain at time t_K, and the contested funds are $V_c = \sum_{i=1}^{K} v_i - V_a$. The moment it is registered in the blockchain, every seller r_i such that $i <= K$ has their funds reduced by V_c, as long as their respective transactions happened within a N_{adsc} block distance to the current one.

Since $t_K < T$, this process is visible to all sellers in R and I before they produce any good or service, which causes them to reject Alice's request. It is important for the members of I to wait as well, because not waiting would provide an economic incentive for Alice to perform the attack; obtaining the goods or services provided by I.

Let us now assume a closely related but slightly different scenario. In this new case, Alice engages in a constant stream of low p transactions with Bob, in exchange for goods of value $v_1, v_2...$, while monitoring the unconfirmed transactions pool. When she realizes that one of them is redeemable, she immediately purchases an expensive good of value $v_e \approx V_a$ from Carol, with $p = 1$.

Under this assumption all participants are patient, so Carol would easily perceive the attack and refuse to serve Alice. From an economic perspective, this would be then reduced to a listener double spend under ADSC, where Bob suffers some minor economic damage at Alice's own cost, but would result in no financial gain for Alice.

Condition 2: The sellers involved are sufficiently well connected. Generally, mounting a double-spend attack in ecosystems such as Bitcoin is based on precise timing and visibility manipulation[7], so that a race condition is created where none of the victims are aware of it. A SPIN powered market makes this kind of manipulation more difficult due to the inversion of control involved in making a stochastic payment; since the seller is responsible for adding the transaction to the pool, they can afford to be as thorough and selective as necessary. In fact, they are *incentivized* to be redundant and provide visibility over the transaction.

A seller that decides to relay the transaction to an isolated mining pool or in a deliberately obfuscated way is acting against his own interests, since it increases the chance that another seller will unwillingly trigger ADSC. Unless of course, the seller is colluding with the buyer, in which case no real economic transaction is taking place and we are back to the listener double spend attack model. Similarly, a seller that does not check the pools before uploading their own transaction is not being rational, since he knows there is a risk ADSC will trigger, costing him the funds and the transaction fee.

This condition is the strongest guarantee against real double spend attacks, and we estimate it will suffice for most markets.

Sellers only need to verify that the redeemable transaction is visible enough (i.e. present in major mining pools that he can reasonably expect other sellers in the same space to monitor) before producing the good or service. For particular high value transactions with $p \approx 1$, the seller may selectively choose to wait until T, but for low value transactions, this implicit cooperation between legitimate sellers is sufficient.

V. Conclusion

The SPIN protocol represents a new paradigm in low value blockchain payments. It introduces the concept of stochastic transactions, thanks to which it is possible to arbitrate commerce for arbitrarily small quantities, by aggregating them into a smaller set of randomly sampled transactions, which capture the same economic activity over a period of time.

We believe SPIN is a worthwhile tool towards worldwide adoption of cryptocurrencies as a means of exchange. To provide a first working example, we will integrate SPIN with Tendril, our upcoming platform for community network governance and arbitration of networking infrastructure. As we develop it, we will release an open source reference SPIN implementation for the community to adapt for their own blockchain projects.

References

[1] https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

[2] https://bitinfocharts.com/comparison/ethereum-transactionfees.html

[3] Poon, J., Dryja, T., *The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments* (2016)

[4] http://raiden.network

[5] https://github.com/ethereum/wiki/wiki/Sharding-FAQ

[6] http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=LEGISSUM:l24039&from=EN&isLegissum=true

[7] O. Karame, G., Androulaki, E., Capkun, S., *Two Bitcoins at the Price of One? Double Spending attacks on Fast Payments in Bitcoin* (2012)

[8] Nakamoto, S., *Bitcoin: A Peer-to-Peer Electronic Cash System* (2008)