

Trust is Risk: A Decentralized Financial Trust Platform

Orfeas Stefanos Thyfronitis Litos
School of Informatics
University of Edinburgh
Edinburgh, United Kingdom
o.thyfronitis@ed.ac.uk

Dionysis Zindros
Department of Informatics
National and Kapodistrian University of Athens
Athens, Greece
dionyziz@di.uoa.gr

Abstract—Centralized reputation systems use stars and reviews and thus require algorithm secrecy to avoid manipulation. In autonomous open source decentralized systems this luxury is not available. We create a reputation network for decentralized marketplaces where the trust each user gives to the other users is quantifiable and expressed in monetary terms. We introduce a new model for bitcoin wallets in which user coins are split among trusted associates. Direct trust is defined using shared bitcoin accounts via bitcoin’s 1-of-2 multisig. Indirect trust is subsequently defined transitively. This enables formal game theoretic arguments pertaining to risk analysis. We prove that risk and maximum flows are equivalent in our model and that our system is Sybil-resilient. Our system allows for concrete financial decisions on the subjective monetary amount a pseudonymous party can be trusted with. Risk remains invariant under a direct trust redistribution operation followed by a purchase.

I. INTRODUCTION

Online marketplaces can be categorized as centralized and decentralized. Two examples of each category are [ebay](#) and [OpenBazaar](#). The common denominator of online marketplaces is that the reputation of each vendor and client is typically expressed in the form of stars and user-generated reviews that are viewable by the whole network.

The goal of “Trust is Risk” is to offer a reputation system for decentralized marketplaces where the trust each user gives to the others is quantifiable in monetary terms. The central assumption used throughout this paper is that trust is equivalent to risk, or the proposition that *Alice*’s trust in another user *Charlie* is defined as the *maximum sum of money Alice* can lose when *Charlie* is free to choose any strategy. To flesh out this concept, we will use *lines of credit* as proposed by Sanchez [1]. *Alice* joins the network by explicitly entrusting some money to another user, say her friend, *Bob* (see Fig. 1 and 2). If *Bob* has already entrusted some money to a third user, *Charlie*, then *Alice* indirectly trusts *Charlie* since if the latter wished to play unfairly, he could have already stolen the money entrusted to him by *Bob*. We will later see that *Alice* can now engage in economic interaction with *Charlie*.

Research supported by ERC project CODAMODA, project #259152

To implement lines-of-credit, we use Bitcoin [2], a decentralized cryptocurrency that differs from conventional currencies in that it does not depend on trusted third parties. All transactions are public as they are recorded on a decentralized ledger, the blockchain. Each transaction takes some coins as input and produces some coins as output. If the output of a transaction is not connected to the input of another one, then this output belongs to the UTXO, the set of unspent transaction outputs. Intuitively, the UTXO contains all coins not yet spent.

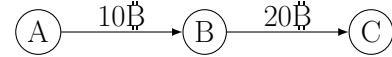


Fig. 1: A indirectly trusts C 10฿

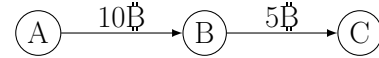


Fig. 2: A indirectly trusts C 5฿

We propose a new kind of wallet where coins are not exclusively owned, but are placed in shared accounts materialized through 1-of-2 multisigs, a bitcoin construct that permits any one of two predesignated users to spend the coins within the shared account [3]. We use the notation $1/\{Alice, Bob\}$ to represent a 1-of-2 multisig that can be spent by either *Alice* or *Bob*. In this notation, the order of names is irrelevant, as either user can spend. However, the user who deposits the money initially into the shared account is relevant – she is the one risking her money.

Our approach changes the user experience in a subtle but drastic way. A user no more has to base her trust towards a store on stars or ratings which are not expressed in financial units. She can simply consult her wallet to decide whether the store is trustworthy and, if so, up to what value, denominated in bitcoin. This system works as follows: Initially *Alice* migrates her funds from her private bitcoin wallet to 1-of-2 multisig addresses shared with friends she trusts. We call this direct trust. Our system is agnostic to the means players use to determine who is trustworthy for these direct 1-of-2 deposits. Nevertheless, these deposits contain an objective value visible to the

network that can be used to deterministically evaluate subjective indirect trust towards other users.

Suppose *Alice* is viewing the listings of vendor *Charlie*. Instead of his stars, *Alice* sees a positive value calculated by her wallet representing the maximum value she can safely pay to purchase from *Charlie*. This value, known as indirect trust, is calculated in Theorem 2 – Trust Flow. Indirect trust towards a user is not global but subjective; each user views a personalized indirect trust based on the network topology. The indirect trust reported by our system maintains the following security property: If *Alice* makes a purchase from *Charlie*, then she is exposed to no more risk than she was already taking willingly. The existing voluntary risk is exactly that which *Alice* was taking by sharing her coins with her trusted friends. We prove this in Theorem 3 – Risk Invariance. Obviously it is not safe for *Alice* to buy anything from any vendor if she has not directly entrusted any value to other users.

In Trust is Risk, money is not invested at the time of purchase directly to the vendor, but at an earlier point in time and only to parties that are trustworthy for out-of-band reasons. We will clarify the fact that this system can function in a completely decentralized fashion in Theorem 5 – Sybil Resilience.

We make the design choice that an entity can express her trust maximally in terms of her available capital. Thus, an impoverished player cannot allocate much direct trust to her friends, no matter how trustworthy they are. On the other hand, a rich player may entrust a small fraction of her funds to a player that she does not extensively trust and still exhibit more direct trust than the impoverished player. There is no upper limit to trust; each player is only limited by her funds. We thus take advantage of the following remarkable property of money: To normalise subjective human preferences into objective value.

A user has many incentives to join. First, she has access to otherwise inaccessible stores. Moreover, two friends can formalize their trust by directly entrusting the same amount to each other. A company that casually subcontracts others can express its trust towards them. Governments can choose to directly entrust citizens with money and confront them with a suitable legal arsenal if they make irresponsible use of this trust. Banks can provide loans as outgoing and manage savings as incoming direct trust. Last, the network is an investment and speculation field since it constitutes a new area for financial activity.

Observe that the same physical person can maintain multiple pseudonymous identities in the same trust network and that multiple independent trust networks for different purposes can coexist.

Trust is Risk is not just a theoretical conception, but can be deployed and applied in existing decentralized markets such as OpenBazaar. All the necessary bitcoin constructs such as multisigs are readily available. Our only concern pertains to the scalability of such an implementation, but we are confident that such difficulties can be overcome.

II. MECHANICS

We now trace *Alice*'s steps from joining the network to successfully completing a purchase. Suppose initially all her coins, say 10 B , are under her exclusive control.

Two trustworthy friends, *Bob* and *Charlie*, persuade her to try out Trust is Risk. She installs the Trust is Risk wallet and migrates the 10 B from her regular wallet, entrusting 2 B to *Bob* and 5 B to *Charlie*. She now exclusively controls 3 B . She is risking 7 B to which she has full but not exclusive access in exchange for being part of the network.

A few days later, she discovers a shoes shop owned by *Dean*, also a member of Trust is Risk. She finds a pair of shoes that costs 1 B and checks *Dean*'s trustworthiness through her new wallet. Suppose *Dean* is deemed trustworthy up to 5 B . Since 1 B < 5 B , she confidently proceeds to purchase the shoes with her new wallet.

She can then see in her wallet that her exclusive coins have remained 3 B , the coins entrusted to *Charlie* have been reduced to 4 B and *Dean* is entrusted 1 B , equal to the value of the shoes. Also, her purchase is marked as pending. If she checks her trust towards *Dean*, it still is 5 B . Under the hood, her wallet redistributed her entrusted coins in a way that ensures *Dean* is directly entrusted with coins equal to the value of the purchased item and that her reported trust towards him has remained invariant.

Eventually the shoes reach *Alice*. *Dean* chooses to redeem *Alice*'s entrusted coins, so her wallet does not show any coins entrusted to him. Through her wallet, she marks the purchase as successful. This lets the system replenish the reduced trust to *Bob* and *Charlie*, setting the entrusted coins to 2 B and 5 B respectively once again. *Alice* now exclusively owns 2 B . Thus, she can now use a total of 9 B as expected, since the shoes cost 1 B .

III. THE TRUST GRAPH

We now formally describe the proposed system.

Definition 1 (Graph). *Trust is Risk is represented by a sequence of directed weighted graphs (\mathcal{G}_j) where $\mathcal{G}_j = (\mathcal{V}_j, \mathcal{E}_j)$, $j \in \mathbb{N}$. Also, there exists a corresponding sequence of weight functions (c_j) with $c_j : \mathcal{E}_j \rightarrow \mathbb{R}^+$.*

The nodes represent the players, the edges represent the existing direct trusts and the weights represent the amount of value attached to the corresponding direct trust. As we will see, the game evolves in turns. The subscript of the graph represents the corresponding turn.

Definition 2 (Players). *The set $\mathcal{V}_j = \mathcal{V}(\mathcal{G}_j)$ is the set of all players in the network, otherwise understood as the set of all pseudonymous identities.*

Each node has a corresponding non-negative number that represents its capital. A node's capital is the value that the node possesses exclusively; nobody else can spend it.

Definition 3 (Capital). *The capital of A in turn j , $\text{Cap}_{A,j}$, is defined as the number of coins that belong exclusively to A at the beginning of turn j .*

The capital of A can be reallocated only during her turns, according to her actions. We model the system in a way that no capital can be added in the course of the game through external means. The use of capital will become clear once turns are formally defined.

The formal definition of direct trust follows:

Definition 4 (Direct Trust). *Direct trust from A to B at the end of turn j , $DTr_{A \rightarrow B,j}$, is defined as the total finite amount that exists in $1/\{A,B\}$ multisigs in the UTXO in the end of turn j , where the money is deposited by A .*

$$DTr_{A \rightarrow B,j} = \begin{cases} c_j(A,B), & \text{if } (A,B) \in \mathcal{E}_j \\ 0, & \text{else} \end{cases} \quad (1)$$

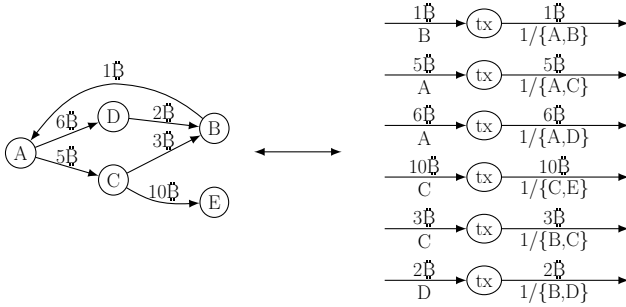


Fig. 3: Trust is Risk Game Graph and Equivalent Bitcoin UTXO

The definition of direct trust agrees with the title of this paper and coincides with the intuition and sociological experimental results of Karlan et al. [4] that the trust *Alice* shows to *Bob* in real-world social networks corresponds to the extent of danger in which *Alice* is putting herself into in order to help *Bob*. An example graph with its corresponding transactions in the UTXO is found in Fig. 3.

Any algorithm that has access to the graph \mathcal{G}_j has implicitly access to all direct trusts of this graph.

Definition 5 (Neighbourhood). *The nodes directly trusted by A at the end of turn j are $N^+(A)_j$. The nodes that directly trust A at the end of turn j are $N^-(A)_j$.*

$$\begin{aligned} N^+(A)_j &= \{B \in \mathcal{V}_j : DTr_{A \rightarrow B,j} > 0\}, \\ N^-(A)_j &= \{B \in \mathcal{V}_j : DTr_{B \rightarrow A,j} > 0\}. \end{aligned} \quad (2)$$

Definition 6 (Total In/Out Direct Trust). *At the end of turn j , the total incoming and outgoing direct trust to and from A is $in_{A,j}$ and $out_{A,j}$ respectively.*

$$in_{A,j} = \sum_{v \in N^-(A)_j} DTr_{v \rightarrow A,j}, \quad out_{A,j} = \sum_{v \in N^+(A)_j} DTr_{A \rightarrow v,j} \quad (3)$$

IV. EVOLUTION OF TRUST

Trust is Risk is a game that runs indefinitely. In each turn, a player is chosen, decides what to play and, if valid, the chosen turn is executed.

Definition 7 (Turns). *In each turn j a player $A \in \mathcal{V}$, $A = \text{Player}(j)$, chooses one or more actions from the following two kinds:*

Steal(y_B, B): *Steal value y_B from $B \in N^-(A)_{j-1}$, where $0 \leq y_B \leq DTr_{B \rightarrow A,j-1}$. Then set $DTr_{B \rightarrow A,j} = DTr_{B \rightarrow A,j-1} - y_B$.*

Add(y_B, B): *Add value y_B to $B \in \mathcal{V}$, where $-DTr_{A \rightarrow B,j-1} \leq y_B$. Then set $DTr_{A \rightarrow B,j} = DTr_{A \rightarrow B,j-1} + y_B$.*

$y_B < 0$ amounts to direct trust reduction, while $y_B > 0$ to direct trust increase.

Let Y_{st}, Y_{add} be the total value to be stolen and added respectively by A . The capital is updated in every turn: $Cap_{A,j} = Cap_{A,j-1} + Y_{st} - Y_{add}$. For a turn to be valid we require $Cap_{A,j} \geq 0$ and $DTr_{A \rightarrow B,j} \geq 0$ and $DTr_{B \rightarrow A,j} \geq 0$. A player cannot choose two actions of the same kind against the same player in one turn. Turn_j denotes the set of actions in turn j . The graph that emerges by applying the actions on \mathcal{G}_{j-1} is \mathcal{G}_j .

We use $\text{prev}(j)$ to denote the previous turn played by $\text{Player}(j)$. If she has not played before, $\text{prev}(j) = 0$.

Definition 8 (Damage). *Let $A = \text{Player}(j)$.*

$$Dmg_{A,j} = out_{A,\text{prev}(j)} - out_{A,j-1} \quad (4)$$

We say that A has been stolen value $Dmg_{A,j}$ between $\text{prev}(j)$ and j . We omit turn subscripts if implied.

Definition 9 (History). *We define History, $\mathcal{H} = (\mathcal{H}_j)$, as the sequence of all tuples containing the sets of actions and the corresponding player.*

$$\mathcal{H}_j = (\text{Player}(j), \text{Turn}_j) \quad (5)$$

Knowledge of the \mathcal{G}_0 graph, players' initial capital and the history equal full knowledge of the evolution of the game. Building on Fig. 3, Fig. 4 is the result when D plays

$$\text{Turn}_1 = \{\text{Steal}(1, A), \text{Add}(4, C), \text{Add}(-1, B)\} \quad (6)$$

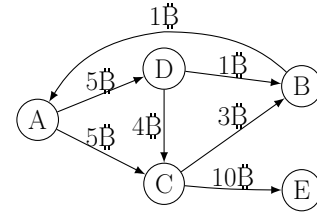


Fig. 4: Game Graph after Turn_1 (6) on the Graph of Fig. 3

The pseudocode defining the Trust is Risk game and other functionalities can be found in [5].

V. TRUST TRANSITIVITY

In this section we define some strategies for the players and the Transitive Game, the worst-case scenario for an honest player when another player plays maliciously.

Definition 10 (Idle Strategy). *An idle player passes her turn.*

Definition 11 (Evil Strategy). *An evil player steals all incoming direct trust and nullifies her outgoing direct trust.*

Definition 12 (Conservative Strategy). *A conservative player replenishes the value she lost since the previous turn by stealing from others who directly trust her as much as she can up to Dmg_A .*

The rationale behind this strategy arises from a real-world common situation. Suppose there are a client, an intermediary and a producer. The client entrusts some value to the intermediary so that the latter can buy the desired product from the producer and deliver it to the client. The intermediary in turn entrusts an equal value to the producer, who needs the value upfront to be able to complete the production process. However the producer eventually breaks the trust by not giving the product neither reimbursing the value. The intermediary can choose either to reimburse the client and suffer the loss, or refuse to return the money and lose the client's trust. The latter choice for the intermediary is exactly the conservative strategy. It is used throughout this work as a strategy for all the intermediary players because it models effectively the worst-case scenario that a client can face after an evil player decides to steal everything she can and the rest of the players do not engage in evil activity.

The Transitive Game is the special game of the Trust is Risk Game in which one player is Idle, one player is Evil and the rest are Conservative. An example follows.

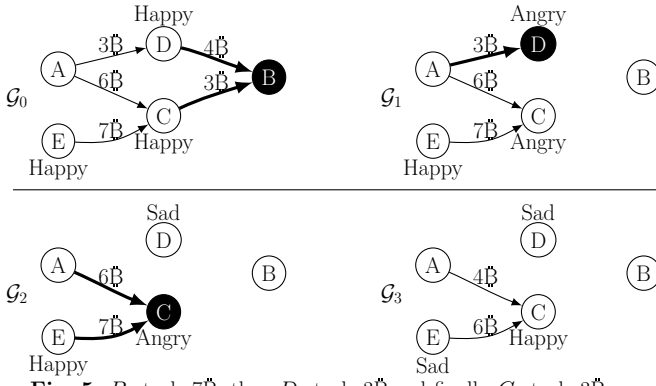


Fig. 5: B steals 7€ , then D steals 3€ and finally C steals 3€

In turn 0, there is already a network in place. All players apart from A and B follow the conservative strategy. Each conservative player can be in one of three states: Happy, Angry or Sad. Happy players have 0 loss, Angry players have positive loss and positive incoming direct trust, thus are able to replenish their loss at least in part; Sad players have positive loss, but 0 incoming direct trust, thus they cannot replenish the loss.

Let j_0 be the first turn on which B is chosen to play. Until then, all players will pass their turn since nothing has been stolen yet [5]. Moreover, let $v = \text{Player}(j)$. The

Transitive Game generates turns:

$$\text{Turn}_j = \bigcup_{w \in N^-(v)_{j-1}} \{\text{Steal}(y_w, w)\} \text{ , where} \quad (7)$$

$$\sum_{w \in N^-(v)_{j-1}} y_w = \min(in_{v,j-1}, Dmg_{v,j}) \text{ .} \quad (8)$$

We see that if $Dmg_{v,j} = 0$, then $\text{Turn}_j = \emptyset$. From the definition of $Dmg_{v,j}$ and knowing that no strategy in this case can increase any direct trust, we see that $Dmg_{v,j} \geq 0$. Also $Loss_{v,j} \geq 0$ because if $Loss_{v,j} < 0$, then v has stolen more value than she has been stolen, thus she would not be following the conservative strategy.

VI. TRUST FLOW

Definition 13 (Indirect Trust). *Indirect trust from A to B after turn j is defined as the maximum possible value that can be stolen from A after turn j in the setting of $\text{TransitiveGame}(\mathcal{G}_j, A, B)$.*

Note that $Tr_{A \rightarrow B} \geq DTr_{A \rightarrow B}$. The next result shows $Tr_{A \rightarrow B}$ is finite.

Theorem 1 (Trust Convergence Theorem).

Consider a Transitive Game. There exists a turn such that all subsequent turns are empty.

Proofs of all theorems and lemmas can be found in [5].

In the setting of $\text{TransitiveGame}(\mathcal{G}, A, B)$ and j being a turn in which the game has converged, we use the notation $Loss_A = Loss_{A,j}$. $Loss_A$ is not the same for repeated executions of this kind of game, since the order in which players are chosen may differ between executions and conservative players can change which incoming direct trusts they will steal and how much from each.

Let G be a weighted directed graph. We investigate the maximum flow on it. For an introduction to maximum flows see Introduction to Algorithms, p. 708 [6]. Considering each edge's capacity as its weight, a flow assignment $X = [x_{vw}]_{\mathcal{V} \times \mathcal{V}}$ with source A and sink B is valid when:

$$\forall (v, w) \in \mathcal{E}, x_{vw} \leq c_{vw} \text{ and} \quad (9)$$

$$\forall v \in \mathcal{V} \setminus \{A, B\}, \sum_{w \in N^+(v)} x_{vw} = \sum_{w \in N^-(v)} x_{vw} \text{ .} \quad (10)$$

The flow value is $\sum_{v \in N^+(A)} x_{Av} = \sum_{v \in N^-(B)} x_{vB}$. We do not suppose skew symmetry in X . There exists an algorithm $\text{MaxFlow}(A, B)$ that returns the maximum possible flow from A to B . This algorithm needs full knowledge of the graph and runs in $O(|\mathcal{V}||\mathcal{E}|)$ time [7]. We refer to the flow value of $\text{MaxFlow}(A, B)$ as $\text{maxFlow}(A, B)$.

We will now introduce two lemmas that will be used to prove one of our central results, the Trust Flow theorem.

Lemma 1 (MaxFlows Are Transitive Games).

Let \mathcal{G} be a game graph, let $A, B \in \mathcal{V}$ and $\text{MaxFlow}(A, B)$ the maximum flow from A to B executed on \mathcal{G} . There

exists an execution of $\text{TransitiveGame}(\mathcal{G}, A, B)$ such that $\text{maxFlow}(A, B) \leq \text{Loss}_A$.

Lemma 2 (Transitive Games Are Flows).

Let $\mathcal{H} = \text{TransitiveGame}(\mathcal{G}, A, B)$ for some game graph \mathcal{G} and $A, B \in \mathcal{V}$. There exists a valid flow $X = \{x_{uv}\}_{\mathcal{V} \times \mathcal{V}}$ on \mathcal{G}_0 such that $\sum_{v \in \mathcal{V}} x_{Av} = \text{Loss}_A$.

Theorem 2 (Trust Flow Theorem).

Let \mathcal{G} be a game graph and $A, B \in \mathcal{V}$. It holds that

$$\text{Tr}_{A \rightarrow B} = \text{maxFlow}(A, B) .$$

Note that the maxFlow is the same in the following two cases: If a player chooses the evil strategy and if that player chooses a variation of the evil strategy where she does not nullify her outgoing direct trust.

Further justification of trust transitivity through the use of MaxFlow can be found in the sociological work by Karlan et al. [4] where a direct correspondence of maximum flows and empirical trust is experimentally validated.

Here we see another important theorem that gives the basis for risk-invariant transactions between different, possibly unknown, parties.

Theorem 3 (Risk Invariance Theorem). Let \mathcal{G} be a game graph, $A, B \in \mathcal{V}$ and l the desired value to be transferred from A to B , with $l \leq \text{Tr}_{A \rightarrow B}$. Let also \mathcal{G}' with the same nodes as \mathcal{G} such that

$$\forall v \in \mathcal{V}' \setminus \{A\}, \forall w \in \mathcal{V}', \text{DTr}'_{v \rightarrow w} = \text{DTr}_{v \rightarrow w} .$$

Furthermore, suppose that there exists an assignment for the outgoing direct trust of A , $\text{DTr}'_{A \rightarrow v}$, such that

$$\text{Tr}'_{A \rightarrow B} = \text{Tr}_{A \rightarrow B} - l . \quad (11)$$

Let another game graph, \mathcal{G}'' , be identical to \mathcal{G}' except for the following change: $\text{DTr}''_{A \rightarrow B} = \text{DTr}'_{A \rightarrow B} + l$. It then holds that

$$\text{Tr}''_{A \rightarrow B} = \text{Tr}_{A \rightarrow B} .$$

A can reduce her outgoing direct trust in a manner that achieves (11), since $\text{maxFlow}(A, B)$ is continuous with respect to A 's outgoing direct trusts.

VII. SYBIL RESILIENCE

One of our aims is to mitigate Sybil attacks [8] whilst maintaining decentralized autonomy [9]. We begin by extending the definition of indirect trust.

Definition 14 (Indirect Trust to Multiple Players). Indirect trust from player A to a set of players, $S \subset \mathcal{V}$ is defined as the maximum possible value that can be stolen from A if all players in S are evil, A is idle and everyone else ($\mathcal{V} \setminus (S \cup \{A\})$) is conservative. Formally, let choices be the different actions between which the conservative players choose, then

$$\text{Tr}_{A \rightarrow S, j} = \max_{j': j' > j, \text{choices}} [\text{out}_{A, j} - \text{out}_{A, j'}] . \quad (12)$$

We now extend the Trust Flow theorem to many players.

Theorem 4 (Multi-Player Trust Flow).

Let $S \subset \mathcal{V}$ and T be an auxiliary player such that, for the sake of argument, $\forall B \in S, \text{DTr}_{B \rightarrow T} = \infty$. It holds that

$$\forall A \in \mathcal{V} \setminus S, \text{Tr}_{A \rightarrow S} = \text{maxFlow}(A, T) .$$

We now define several useful notions to tackle the problem of Sybil attacks. Let Eve be a possible attacker.

Definition 15 (Corrupted Set). Let \mathcal{G} be a game graph and let Eve have players $\mathcal{B} \subset \mathcal{V}$ corrupted, so that she fully controls their actions. We call this the corrupted set. The players \mathcal{B} are considered legitimate before the corruption, thus they may be directly trusted by any player in \mathcal{V} .

Definition 16 (Sybil Set). Let \mathcal{G} a game graph. Joining is free, so Eve can create unlimited players. These players are the Sybil set, \mathcal{C} . Moreover, Eve fully controls their actions. However, players \mathcal{C} can be directly trusted only by players $\mathcal{B} \cup \mathcal{C}$, not by players $\mathcal{V} \setminus (\mathcal{B} \cup \mathcal{C})$, with \mathcal{B} the corrupted set.

Definition 17 (Collusion). Let \mathcal{G} be a game graph. Let $\mathcal{B} \subset \mathcal{V}$ be a corrupted set and $\mathcal{C} \subset \mathcal{V}$ be a Sybil set. The tuple $(\mathcal{B}, \mathcal{C})$ is called collusion and is controlled by Eve.

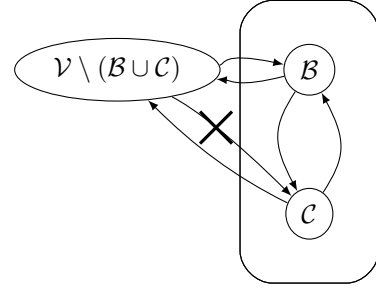


Fig. 6: Collusion

Players $\mathcal{V} \setminus (\mathcal{B} \cup \mathcal{C})$ perceive the collusion as independent players with a distinct strategy each, whereas in reality they are all subject to a single strategy dictated by Eve.

Theorem 5 (Sybil Resilience).

Let $(\mathcal{B}, \mathcal{C})$ be a collusion of players on \mathcal{G} . It is

$$\text{Tr}_{A \rightarrow \mathcal{B} \cup \mathcal{C}} = \text{Tr}_{A \rightarrow \mathcal{B}} .$$

We saw that controlling $|\mathcal{C}|$ is irrelevant for Eve, thus Sybil attacks are pointless. The theorem does not guard against deception attacks though. A malicious player can make many accounts, use them legitimately to inspire the confidence of others and then switch to the evil strategy, defrauding everyone that trusted the fabricated identities. These identities correspond to the corrupted set of players and not to the Sybil set because they have direct incoming trust from outside the collusion.

In conclusion, we have delivered on our promise of a Sybil-resilient decentralized financial trust system with invariant risk for purchases.

VIII. RELATED WORK

Webs-of-trust can be used as a basis for trust [10]. PGP [11] implements one and Freenet [12] another, built against spam. Ways of calculating trust towards distant nodes are proposed in [13], [14]. In [15], trust is calculated hierarchically. CA- and Byzantine-based [16] PKIs [17] require central trusted third parties or at least authenticated membership. [18], [19], [20], [21] do not prove any Sybil resilience. All these systems do not define trust financially.

To discourage Sybil attacks, Beaver [22] relies on fees, something we avoid. Our motivating application is OpenBazaar, where transitive financial trust has previously been explored in [9]. We are strongly inspired by [4] who give a sociological justification for the central design choice of identifying trust with risk. We appreciate [23], which proposes a financial trust system with transitivity, with trust defined as lines-of-credit, similar to us. We extended their work by using the blockchain for automated proofs-of-risk, a feature not available to them at the time.

Our Conservative strategy and Transitive Game remind [24] which is also financially transitive and is used by [25], [26]. IOUs in those correspond to our edges of trust reversed. The critical difference is that our trust is expressed in a global currency and there is no money-as-debt. Furthermore, we proved that trust and maximum flows are equivalent, a direction not explored in their papers.

IX. FURTHER RESEARCH

Before a purchase, outgoing direct trust should be reduced as in (11). [27] discusses suitable algorithms.

Our game is static. In a dynamic setting, users will play simultaneously, freely join, depart or disconnect temporarily. In a future analysis we will model repeated purchases and treat trust as part of the utility function.

MaxFlow now needs full network knowledge, which defeats privacy [28]. Calculating the flows in zero knowledge remains an open question. [29] and [30] offer insight into how privacy can be achieved.

An open source implementation of an API in Bitcoin for OpenBazaar is under development. After deployment, analysis of the usage patterns will give insights into the social dynamics Trust is Risk promotes.

REFERENCES

- [1] Sanchez W.: Lines of Credit. <https://gist.github.com/drwash0/2c40b91e169f55988618#part-3-web-of-credit> (2016)
- [2] Nakamoto S.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
- [3] Antonopoulos A. M.: Mastering Bitcoin: Unlocking Digital Cryptocurrencies. O'Reilly Media, Inc. (2014)
- [4] Karlan D., Mobius M., Rosenblat T., Szeidl A.: Trust and social collateral. *The Quarterly Journal of Economics*: pp. 1307–1361 (2009)
- [5] Thyfronitis Litos O. S., Zindros D.: Trust Is Risk: A Decentralized Financial Trust Platform. *IACR: Cryptology ePrint Archive* (2017)
- [6] Cormen T. H., Leiserson C. E., Rivest R. L., Stein C.: Introduction to algorithms. MIT Press and McGraw-Hill: 3rd ed. (2009)
- [7] Orlin J. B.: Max Flows in $O(nm)$ Time, or Better. In *STOC '13 Proceedings of the forty-fifth annual ACM symposium on Theory of computing*: pp. 765–774: ACM, New York: doi:10.1145/2488608.2488705 (2013)
- [8] Douceur J. R.: The Sybil Attack. *International workshop on Peer-To-Peer Systems* (2002)
- [9] Zindros D.: Trust in Decentralized Anonymous Marketplaces (2015)
- [10] Caronni G.: Walking the web of trust. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, IEEE 9th International Workshops*: pp. 153–158 (2000)
- [11] Zimmermann P.: PGP Source Code and Internals. The MIT Press (1995)
- [12] Clarke I., Sandberg O., Wiley B., Hong T. W.: Freenet: A Distributed Anonymous Information Storage and Retrieval System. In H. Federrath (editor), *Designing Privacy Enhancing Technologies*: pp. 46–66: Springer-Verlag Berlin Heidelberg, Berkeley, USA (2001)
- [13] Mui L., Mohtashemi M., Halberstadt A.: A Computational Model of Trust and Reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*: pp. 2431–2439: IEEE (2002)
- [14] Josang A., Ismail R.: The Beta Reputation System. In *Proceedings of the 15th Bled Electronic Commerce Conference* (2002)
- [15] Vişan A., Pop F., Cristea V.: Decentralized Trust Management in Peer-to-Peer Systems. In *10th International Symposium on Parallel and Distributed Computing*: pp. 232–239 (2011)
- [16] Lamport L., Shostak R., Pease M.: The Byzantine Generals Problem. In *ACM Transaction on Programming Languages and Systems*: vol. 4.3: pp. 382–401 (1982)
- [17] Adams C., Lloyd S.: Understanding PKI: concepts, standards, and deployment considerations. Addison-Wesley Professional (2003)
- [18] Huynh T. D., Jennings N. R., Shadbolt N. R.: An Integrated Trust and Reputation Model for Open Multi-Agent Systems. *Autonomous Agents and Multi-Agent Systems*: vol. 13(2), pp. 119–154 (2006)
- [19] Michiardi P., Molva R.: Core: a Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad-hoc Networks. In *Advanced Communications and Multimedia Security*: pp. 107–121: Springer US (2002)
- [20] Grünert A., Hudert S., Köning S., Kaffile S., Wirtz G.: Decentralized Reputation Management for Cooperating Software Agents in Open Multi-Agent Systems. *ITSSA*: vol. 1(4), pp. 363–368 (2006)
- [21] Repantis T., Kalogeraki V.: Decentralized Trust Management for Ad-hoc Peer-to-Peer Networks. In *Proceedings of the 4th International Workshop of Middleware for Pervasive and Ad-hoc Computing*: p. 6: MPAC: ACM (2006)
- [22] Soska K., Kwon A., Christin N., Devadas S.: Beaver: A Decentralized Anonymous Marketplace with Secure Reputation (2016)
- [23] DeFigueiredo D. D. B., Barr E. T.: TrustDavis: A Non-Exploitable Online Reputation System. *CEC*: vol. 5, pp. 274–283 (2005)
- [24] Fugger R.: Money as IOUs in Social Trust Networks & A Proposal for a Decentralized Currency Network Protocol. <http://archive.ripple-project.org/decentralizedcurrency.pdf> (2004)
- [25] Schartz D., Youngs N., Britto A.: The Ripple protocol consensus algorithm. *Ripple Labs Inc White Paper*: vol. 5 (2014)
- [26] Mazieres D.: The stellar consensus protocol: A federated model for internet-level consensus. *Stellar Development Foundation* (2015)
- [27] Thyfronitis Litos O. S., Zindros D.: Trust Is Risk: A Decentralized Financial Trust Platform - Master Thesis (2017)
- [28] Narayanan A., Shmatikov V.: De-anonymizing Social Networks. In *Proceedings of the 30th Symposium on Security and Privacy*: pp. 173–187: IEEE: 10.1109/SP.2009.22 (2009)
- [29] Malavolta G., Moreno-Sanchez P., Kate A., Maffei M.: Silent-Whispers: Enforcing Security and Privacy in Decentralized Credit Networks (2016)
- [30] Moreno-Sanchez P., Kate A., Maffei M., Pecina K.: Privacy preserving payments in credit networks. In *Network and Distributed Security Symposium* (2015)