Fair and Resilient Incentive Tree Mechanisms *

Yuezhou Lv
Tsinghua University
Beijing, China
lvyz11@tsinghua.edu.cn

Thomas Moscibroda
Microsoft Research Asia & Tsinghua University
Beijing, China
moscitho@microsoft.com

ABSTRACT

We study *Incentive Trees* for motivating the participation of people in crowdsourcing or human tasking systems. In an Incentive Tree, each participant is rewarded for contributing to the system, as well as for soliciting new participants into the system, who then themselves contribute to it and/or themselves solicit new participants. An Incentive Tree mechanism is an algorithm that determines how much reward each individual participant receives based on all the participants' contributions, as well as the structure of the solicitation tree. The sum of rewards paid by the mechanism to all participants is linear in the sum of their total contribution.

In this paper, we investigate the possibilities and limitations of Incentive Trees via an axiomatic approach by defining a set of desirable properties that an incentive tree mechanism should satisfy. We give a mutual incompatibility result showing that there is no incentive tree mechanism that simultaneously achieves all the properties. We then present two novel families of incentive tree mechanisms. The first family of mechanisms achieves all desirable properties, except that it fails to protect against a certain strong form of multi-identity attack; the second set of mechanisms achieves all properties, including the strong multi-identity protection, but fails to give participants the opportunity to achieve unbounded reward. Given the above impossibility result, these two mechanisms are effectively the best we can hope for. Finally, our model and results generalize recent studies on multi-level marketing mechanisms.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

Keywords

multi-level marketing, incentive trees, reward mechanisms

*This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00302, the National Natural Science Foundation of China Grant 61033001, 61061130540, and the Hi-Tech research and Development Program of China Grant 2006AA10Z216.

1. INTRODUCTION

There has recently been substantial interest in crowdsourcing and human-computation systems. These systems are based on mobilizing and utilizing people's work in order to quickly and efficiently achieve certain tasks. Commercial offerings such as Gigwalk [1] or Amazon's Mechanical Turk [2] allow users to submit tasks and recruit people to complete those tasks. Crowdsourcing is increasingly being used as the method of choice to obtain large-scale user data, such as environmental data, application traces, or to generate indoor-localization maps, e.g. [15, 14]. One key challenge in successfully deploying any such system is the question of how to incentivize people to actually perform tasks and contribute meaningfully. In fact, the same challenge is found in many other systems that rely on user contributions. For example, systems such as social forums, file-sharing services, public computing projects (e.g. SETI@Home), collaborative reference work, etc suffer from the well-known network-effect bootstrapping problem. These systems can become self-sustaining when the scale of the participation list exceeds a certain threshold, but below this threshold, they may not provide sufficient inherent benefit for users to participate in.

One common type of incentive mechanisms for raising user participation in such systems is *Incentive Trees.* Incentive Trees are tree-based mechanisms in which (i) each participant is rewarded for contributing to the system, and in addition, (ii) a participant that has already joined the system can make referrals, and thereby solicit new participants to also join the system and contribute to it. The mechanism incentivizes such solicitations by making a solicitor's reward depend on the contributions (and recursively also on their further solicitations, etc) made by such solicitees. Incentive Trees have been widely used in a variety of domains and under different names, e.g., in *referral trees*, *multi-level marketing* schemes, affiliate marketing or even in the form of the infamous illegal Pyramid Schemes. The question of how people can be incentivized using Incentive Trees to participate in crowdsourcing or network-effect systems is of significant interest and—starting from the work on *Lottery Trees* in [9], and most prominently through the work by the MIT team on the Red Balloon Challenge [6]—has recently attracted significant interest from the research community.

In this work, we study the foundations of *Incentive Trees.* An *Incentive Tree Mechanism* takes as input a weighted tree, where each node's weight denotes its contribution to the system, and the tree structure reflects the solicitation history. Based on this input, the mechanism then computes a reward for each node in the tree in such a way that the sum of rewards is linear in the sum of contributions. The question is, how should this reward function look like? Ideally, an Incentive Tree Mechanism is constructed such that every participant is optimally incentivized to both i) contribute to the system as much as possible, and ii) solicit as many new and itself highly-contributing and highly-solicitating participants as possible. As we will see, simultaneously achieving both *contribution and solicitation incentive* is challenging, especially if the mechanism should satisfy additional properties, such as fairness or robustness to strategic behavior.

In this paper, we take an axiomatic approach. We define a set of basic, desirable properties which ideally an incentive tree mechanism should satisfy. These include trivial properties such as the continuing solicitation and continuing contribution incentive properties, as well as more sophisticated properties that relate to the mechanisms resilience to strategic behavior. These are critically important. In web-based campaigns for example, resilience to multi-identity (Sybil [7]) attacks is key.[1]

Results: We study 8 properties of Incentive Trees, that have also been studied in earlier work on incentive trees and multi-level marketing; and suitably generalize these properties to our more general model. We present two novel families of incentive tree reward mechanisms, both of which are based on algorithmic techniques previously unused in the literature on multi-level marketing or incentive trees. The first family of mechanisms achieves all desirable properties, except that it fails to protect against a certain strong form of Sybil attack (technically, it satisfies all properties except UGSA). The second family of mechanisms does yield protection against the strong form of Sybil attack, but fails to give participants the opportunity to achieve unbounded reward (technically, it satisfies all properties except PO/URO). Both mechanisms are resilient to the well-known multi-identity attacks discussed above. Finally, we show that under some mild assumptions, these two mechanisms are essentially the best we can hope for. Specifically, we give an impossibility result showing that no reward scheme can simultaneously achieve UGSA and PO/URO, while maintaining the other properties. Thus, our results imply that that both of our mechanisms achieve a notion of optimality relative to the axiomatic properties we define in this paper: The mechanisms are optimal in the sense that they achieve a maximal mutually satisfiable subset of properties.

1.1 Related Work

The two works most related to ours are the ones by Douceur and Moscibroda on Lottery Trees [9], and the work by Emek et al. on multi-level marketing schemes [10]. The former work is aimed at motivating people to participate in networked systems and bootstrap systems that rely on the network effect. The paper addresses the following question: Assume some system organizer is willing to spend a fixed amount of money in order to incentivize people to do a specific type of work, how should the system be organized to maximize the resulting work? The authors propose *Lottery Trees*, formalize a set of desirable properties, prove impossibility results, and devise two non-trivial mechanisms one of which achieves near-optimality in terms of achieved desirable properties. Our model differs from [9]: In our incentive tree model, the total amount of reward distributed to the participants *grows linearly* in the total contribution, whereas in [9], the total reward is a fixed, constant value. This difference significantly changes the achievable properties.

The work by Emek et al. [10] has initiated the algorithmic study on multi-level marketing mechanisms. It proposes mechanisms for a model in which users can purchase items (specifically, each user can purchase one item of a fixed unit price). Participants join the system by buying a product, and can then refer friends to also buy this product. The paper proves several properties of such unit-price multi-level marketing schemes and proposes mechanisms that achieve a subset of these properties. The incentive tree model we study in this paper can directly be translated into the multi-level marketing context. When viewed in this context, our work yields a substantially generalized version of the model in [10]: Participants correspond to buyers, and a participant's contribution corresponds to the amount of goods purchased. The difference is that whereas in [10], each buyer can only purchase a single item of unit price (i.e., each participant makes the same contribution to the system), in our model participants can make arbitrary contributions, i.e., each buyer can buy goods at arbitrary price. This generalized version of the problem yields a richer structure, and allows us to generalize the desirable properties in meaningful ways. The results in this paper directly apply to this generalized version of the multi-level marketing model.

In addition to these two works, there has recently been many other work on incentive systems [6]. For example, the Bitcoin system by Babaioff et al. [12] studies a problem similar to multi-level marketing. The paper uses a game-theoretic solution concept to study a problem in which agents are incentivized to forward sensitive information in such a way that the overall system performance is maximized. The work of Drucker and Fleischer [11] studies a multi-level marketing model with multi-items proving properties defined in [10]. Other related work such as [5] on query incentive networks, [4] on finding influential users in a social network, or [3] on the effects of social structure on behavior and norms, is only loosely related to our work. Finally, incentive mechanisms have also been used in mobile systems to recruit people [13] [14].

2. MODEL

In our model, participants can join a system and *contribute* to it (e.g. by doing work such as finding weather balloons, uploading crowd-sourced data, solving tasks, etc). For a participant u, we denote its contribution by $C(u)$, $C(u) \geq 0$. Participants can also *solicit* new participants. Such referrals induce a *referral forest* F. Each participant is a node $u \in F$, and there is a directed edge (u, v) between two participants u and v if v has joined the system in response to a solicitation by u. In other words, if u joins the system via a referral by v, it becomes a child-node of v in F. A new participant u who joins the system independently of any solicitation joins F as an independent node. For simplicity, we consider the equivalent *referral-tree* T, in which there is an imaginary root node r with contribution $C(r) = 0$, and all root-nodes in F are children of r. T is a weighted tree in which the weight of a node u is its contribution to the system $C(u)$. We denote by $C(T) = \sum_{u \in T} C(u)$ the total contribution in the system.

[1] In the commonly employed refer-a-friend programs, for example, it is often very easy to forge identities by creating new free email accounts, and then "referring oneself" in order to get extra reward.

A *reward mechanism* is a function that takes as input the weighted referral tree T, and computes for each $u \in T$ a non-negative real *reward*, denoted by $R(u)$. Following [10], we impose a *budget constraint* on this function: The system administrator is willing to spend no more than a certain fraction $\Phi \leq 1$ of the total accumulated contribution on rewarding participants. That is, the total reward $R(T) = \sum_{u \in T} R(u)$ paid to participants grows linearly in the total contribution, i.e., $R(T) \leq \Phi \cdot C(T)$. While in principle, any function satisfying these properties defines a possible reward mechanism, a well-functioning mechanism should maintain several desirable properties, which we define in Section 3.

Generalized Multi-Level Marketing When viewed in the context of multi-level marketing, our model generalizes the model of Emek et al. [10], allowing buyers to purchase not just a single item of unit price or multi-items, but purchase items at arbitrary prices. Buyers can purchase goods from a seller. For some buyer u, her contribution to the system $C(u)$ is the total *cost* of the goods purchased. The seller is willing to return a certain fraction of his total income in the form of rewards $R(u)$ to the buyers. Notice that in this context, the amount of money a buyer u effectively ends up paying for the goods is his *pay*, $Pay(u) = C(u) - R(u)$. And since a buyer's reward can potentially exceed his cost (if he accumulates many contributing descendants), we also consider the *profit* as $P(u) = R(u) - C(u)$.

Comparison to Existing Models: The two main parameters in our model are contribution and reward. Many existing models have restrictions on either or both parameters. The Pachira in [9], Geometric Mechanism in [10] as well as the winning strategy in the DARPA network challenge [6] demand the total reward to be fixed. In [6] [10] the contribution of each node is the same, while in [9], contributions are allowed to be variable. In previous multilevel marketing models [10] [11], the total reward is linear in the total contribution, but the contribution (payment) of each node is fixed. We generalize these models such that each participant can make different contributions of arbitrary size, and the total reward paid to participants is linear in the total system contribution.

Tree Notation: We use standard tree notation. T_u denotes the subtree rooted at node u. $p_T(u)$ denotes the parent of a node u in T. Finally, $dep_p(u)$ denotes the *depth* of u in T_p, i.e., the distance between u and p. To simplify notation, we define $dep_p(u) = -\infty$ if $u \notin T_p$.

3. DESIRABLE PROPERTIES

In this section, we define the set of desirable properties that an incentive tree mechanism should ideally satisfy. Several of these properties are inspired by related properties defined in [9] for lottery trees; others are taken from [10] and adjusted appropriately to the generalized model with arbitrary contributions.

3.1 Basic Properties

Continuing Contribution Incentive (CCI) [9]: A reward mechanism satisfies CCI if it provides a participant u with increasing reward in response to an increase of u's contribution. This encourages participants to continue contributing to the system (e.g., to continue purchasing goods from the seller). Formally, given a referral tree T. If a node $u \in T$ increases its contribution, $C'(u) > C(u)$, and the contribution of all other nodes $v \in T \setminus \{u\}$ remains the same, $C'(v) = C(v)$, then the reward of u increases: $R'(u) > R(u)$.

Continuing Solicitation Incentive (CSI) [9]: A reward mechanism satisfies CSI if every participant always has an incentive to solicit new participants. This encourages ongoing solicitation and ensures continuing growth of the system. Let T_u and T'_u be the subtree rooted at u before and after a new participant has joined the system in u's subtree. Then, $R'(u) > R(u)$.

Reward Proportional to Contribution (ϕ-RPC) [9]: This property suggests that a reward mechanism should maintain some basic notion of fairness among the participants, the degree of which is determined by the parameter ϕ. We say that a reward mechanism satisfies ϕ-RPC for some $0 \leq \phi \leq 1$, if a participant u who contributes $C(u)$, should at least receive a reward of $R(u) \geq \phi C(u)$. In other words, every participant should receive at least a ϕ-fraction of his contribution to the system. Note that we assume $\phi \leq \Phi$ since otherwise no reward mechanism can satisfy the ϕ-RPC property.

Unbounded Reward Opportunity (URO) [10]: This property demands that there should be no limit to the reward a participant can potentially receive, even when his own contribution is fixed by constant. Formally, a reward mechanism satisfies *URO* if for every positive real R, $C(u)$ and positive integer k, there exists k trees $T_1, \cdots, T_k$ attached to u in the referral tree such that $R(u) \geq R$.

Profitable Opportunity (PO): The PO property is a weaker version of URO. It suggests that a buyer with any positive contribution has the opportunity to get positive profit (reward minus contribution). Formally, a reward mechanism satisfies *PO* if for every positive real $C(u)$ and positive integer k, there exists k trees $T_1, \cdots, T_k$ attached to u in the referral tree such that $R(u) \geq C(u)$. A mechanism that satisfies URO satisfies PO.

Subtree Locality (SL) [10]: This property demands that the reward paid to a participant u is determined uniquely by its subtree T_u, $R(u) = f(T_u)$. The property ensures that each user is credited only for actions (contributions and solicitations) performed by itself, or its descendants. Violation of this property can have undesirable consequences. For example, the reward of a user could increase or decrease without him having taken any action (no new purchases or newly solicited buyers in his subtree). Note that as an important special case, the SL property subsumes the so-called *Unprofitable Solicitor Bypassing (USB)* property defined in [9]. This property demands that for a new participant, it should not matter where in the tree he joins, such that a new participant has no incentive to join the system as a child of someone other than his solicitor.

3.2 Sybil-Attack Resilience Properties

It is desirable that a reward mechanism is robust against strategic behavior by participants. In particular, we seek mechanisms that are resilient against *multi-identity attacks*, commonly known as Sybil-Attacks [7]. A participant who is able to forge multiple identities (which is typically simple in web-based applications) should not be able to use this ability and "cheat" the mechanism for his own benefit. Previous work has defined two different notions of Sybil resilience.

Unprofitable Sybil Attack (USA) [9]: This property is taken directly from [9], and it captures the classic notion of Sybil resilience. The USA property imposes that no participant can increase his profit purely by pretending to have multiple identities: A mechanism satisfies USA if a participant with a given contribution cannot increase his reward by joining the system as a set of Sybil nodes instead of joining as a single node. In other words, a participant who makes a certain contribution to the system should never have a benefit of "splitting" himself and its contribution up and making these contributions as two or more identities, even if these "Sybil identities" join the tree as if referring themselves.

Unprofitable Generalized Sybil Attack (UGSA): This property is strictly stronger than USA. It is a generalization of the so-called *Profitable Sybil Attack* or *Split Proof* property as defined in [10] for the restricted single-item multi-level marketing model. The property demands that a participant can never increase his profit by joining the tree as multiple identities, even if by doing so, he increases his contributions, i.e., purchases additional goods.

We can formally define USA and UGSA as follows. Given a tree T_0. Let u be a participant that joins the tree. Let T_1' be the tree that results when u joins T as a single node. Alternatively, u can join the tree as a set of Sybil nodes $S_u = \{u_1, \ldots, u_k\}$, which can be arbitrarily connected in the referral tree. Let T_1'' be the tree that results when u joins T as the Sybil node set S_u. Let $J = v_1, v_2, \ldots$ be an arbitrary sequence of new participants joining the tree, and let $T_1', T_2', \ldots$ and $T_1'', T_2'', \ldots$ be a sequence of trees resulting from these joins. Notice that in the case u joins as a set of Sybil nodes, there can be many different such sequences because any new child solicited by u can join as a child of any of the Sybil nodes $u_1, \ldots, u_k$. Finally, let $R_i'(u), C_i'(u)$ be the reward and cost of u in T_i', and let $R_i''(u) = \sum_{j=1..k} R_i''(u_j), C_i''(u) = \sum_{j=1..k} C_i''(u_j)$ be the total reward and cost of u in T_i'', respectively. We say that a reward mechanism satisfies USA if for any $i > 0$, $R_i'(u) \geq R_i''(u)$, if $C_i'(u) = C_i''(u)$. We say that a reward mechanism satisfies UGSA if for any $i > 0$, $R_i'(u) - C_i'(u) \geq R_i''(u) - C_i''(u)$, if $C_i'(u) \leq C_i''(u)$. Notice that the UGSA property strictly subsumes the USA property by taking $C_i'(u) = C_i''(u)$.

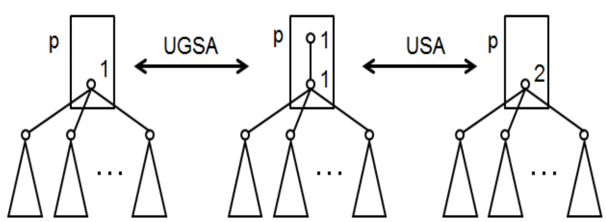

Figure 1: Participant p joining (left) as a single node with cost 1; (middle) as two Sybil nodes that refer one another, each with cost 1; and (right) as a single node with cost 2.

The difference between USA and UGSA is illustrated in Figure 1. USA requires that a participant who contributes a certain amount be unable to increase his reward by joining as multiple identities. Therefore, participant p in the right figure must receive at least as much reward as participant p in the middle figure. UGSA *additionally* demands that p's profit (=reward-cost) in the middle figure cannot exceed his profit in the left figure.

It is interesting to discuss the relative importance of these properties from the point of view of the system administrator or the seller in a multi-level marketing context. USA is clearly a desirable property from his point of view because if USA is violated, he will simply pay too much reward for no additional contribution. The case of UGSA is much less clear. In particular, it is possible that UGSA is violated even though the seller does not actually lose money (i.e., if the contribution exceeds the reward). This is possible if the Sybil buyer p increases his contribution not at the cost of the system administrator, but at the cost of other participants in the system, for instance the parent of p. Practically speaking, we therefore believe that USA is a more fundamental and important property than UGSA. When discussing our TDRM mechanism (end of Section 4.3), we will give a concrete example of TDRM violating UGSA.

4. REWARD MECHANISMS

We start by briefly reviewing existing (multi-level marketing and incentive tree) algorithms and analyze which desirable properties they achieve. We then give an impossibility proof showing that there can be no reward mechanism that simultaneously satisfies URO and UGSA. As the main technical contribution of this paper, we then present two novel reward mechanisms, both of which achieve a maximal subset of mutually satisfiable properties. The mechanism in Section 4.3 achieves all properties except UGSA, and the mechanism in Section 4.4 achieves all properties except URO/PO.

4.1 Existing Incentive Tree Mechanisms

Geometric Mechanism: The simple geometric reward mechanism is commonly used, e.g. in [6]. The idea is that a certain fraction a of a node's contribution "bubbles-up" to its parent, a fraction a^2 bubbles up to its grand-parents, etc. Given two constants $0 < a < 1$ and $b \geq \phi$ such that $b \leq (1-a)\Phi$, the reward of a participant u in the $(a, b) - geometric$ mechanism is defined as follows.

Algorithm 1: (a, b)-Geometric Mechanism
$R(u) = \sum_{v \in T_u} a^{dep_u(v)} \cdot b \cdot C(v)$;

The condition $b \leq (1-a)\Phi$ is to ensure the budget constraint. Specifically, the total reward that a node u is responsible for is at most $b\frac{1}{1-a}C(u)$, which should be less than $\Phi C(u)$. The fairness property $\phi - RPC$ is satisfied if we also set $b \geq \phi$. It is easy to derive the following theorem.

THEOREM 4.1. *The $(a, b) - GeometricMechanism$ with $\phi \leq b \leq (1-a)\Phi$ achieves all desirable properties, except USA and UGSA.*

The reason why USA (and thus, UGSA) is violated is also easy to see. A node can increase his reward by splitting itself into multiple Sybil nodes that are linked to each other as a chain. Some of the "bubbled-up" reward is then handed to other Sybil nodes of u and the total sum of rewards accumulated by u is larger than if u joins as a single node.

Multi-Level Marketing Mechanisms derived from Incentive Tree Mechanisms: In [9], two incentive tree mechanisms are given (called *Luxor* and *Pachira*) for a model in which the total reward in the system is a fixed constant. Any such incentive tree mechanism A for the fixed total reward model can be transformed into an incentive tree mechanism $L - A$ in our model by simply multiplying the reward paid to a user u by a factor of $\Phi C(T)$ (assuming that the total reward is normalized to 1). Applying this transformation to Luxor and Pachira yields two mechanisms L-Luxor and L-Pachira. As it turns out, L-Luxor is very similar to the $(a, b) - GeometricMechanism$, and achieves the same properties. On the other hand, L-Pachira is interesting. For two parameters $0 \le \beta \le 1$ and $\delta > 0$, the $(\beta, \delta) - L - PachiraMechanism$ is defined as follows.

Algorithm 2: (β, δ)-L-Pachira Mechanism

Let u be a participant with k children $q_1, \ldots, q_k$;
Define $\pi(x) = \beta x + (1-\beta)x^{1+\delta}$;
$R(u) = \Phi \cdot C(T) \cdot \left[\pi(\frac{C(T_u)}{C(T)}) - \sum_{i=1}^{k} \pi(\frac{C(T_{q_i})}{C(T)})\right]$;

It was shown in [9] that Pachira achieves USA, and the same proof carries over to L-Pachira as well. Moreover, $\phi - RPC$ can be satisfied by setting $\beta \ge \phi/\Phi$. Pachira does not satisfy the CSI property in the Incentive Tree model. But when transforming it into the multi-level marketing model, L-Pachira does achieve CSI, although the fact is not straightforward. On the other hand, it is easy to see that L-Pachira fails to satisfy the SL constraint, because of its dependency on the total system contribution $C(T)$.

THEOREM 4.2. *The $(\beta, \delta) - L - PachiraMechanism$ with $\beta \ge \phi/\Phi$ achieves all desirable properties, except SL and UGSA.*

Split-Proof Mechanism [10]: For the single-item multi-level marketing model studied in [10], Emek et al. give a mechanism that achieves several properties, including the single-item model equivalent of UGSA and URO. This algorithm is based on the idea of computing a deepest binary subtree of the referral tree and then computing the rewards based on that subtree. Unfortunately, this fails the basic CSI property because depending on the number of direct children it has, a node may no longer have an incentive to directly solicit additional children.

4.2 Impossibility Result

The subsequent constructions of our two new mechanisms are motivated by the following impossibility result, which suggests that *if a mechanism satisfies the SL property, then UGSA and PO (and thus URO) are mutually incompatible.* Since SL is a fundamental property, this result motivates our search for i) a mechanism that achieves all the properties except UGSA (Section 4.3) and ii) a mechanism that achieves all the properties except PO/URO (Section 4.4).

THEOREM 4.3. *There is no incentive tree mechanism that can simultaneously achieve SL, PO and UGSA.*

PROOF. We prove the theorem by contradiction. Suppose a mechanism A can achieve SL, PO and UGSA. In the following proof, all reward computations are done using mechanism A.

Consider a node v^* with $C(v^*) > 0$. According to **PO**, there exists a case in which v^* has one child tree, and yet v^*'s profit is positive, $P(v^*) = R(v^*) - C(v^*) > 0$. We denote the child tree as T^* and its root as u^*. Suppose the contribution of u^* is $C(u^*)$ and $T^* \setminus \{u^*\}$ forms a set of subtrees denoted as $T_1, \cdots, T_k$. According to **SL**, $R(v^*)$ only depends on $C(v^*)$ and T^*. We compare two cases. The first case is exactly as described above (Fig. 2 (left)). The profit of u^* is

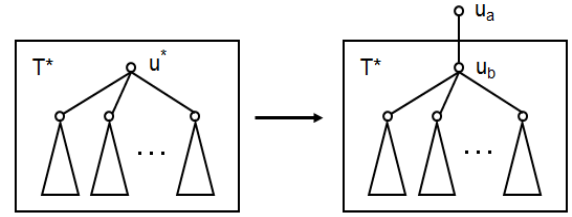

Figure 2: Illustration of notation used in the proof.

$P(u^*) = R(u^*) - C(u^*)$. In the second case (Fig. 2 (right)), node u^* launches a (generalized) Sybil attack by joining the referral tree as two nodes u_a and u_b with $C(u_a) = C(v^*)$ and $C(u_b) = C(u^*)$. Notice that the Sybil attack is generalized (i.e., of the USGA-type), since the total contribution of u_a and u_b exceeds the contribution of u^*. Further notice that in the second case, the root of v^*'s descendant tree is u_a; u_a is u_b's parent; and u_b is the parent of $T_1, \cdots, T_k$, i.e., we keep every node in T^* unchanged except u^*.

According to **SL**, it must hold that u_a has the same reward as v^* (with T^* attached to it), and for the same reason, u_b must have the same reward as u^*. Specifically, it holds that $R(u_a) = R(v^*)$ and $R(u_b) = R(u^*)$. The total profit of u^*'s two Sybil nodes u_a and u_b is thus $P'(u^*) = R(u_a) + R(u_b) - C(u_a) - C(u_b) = (R(v^*) - C(v^*)) + (R(u^*) - C(u^*)) > P(u^*)$. This implies that u^* can get more profit by contributing more, which violates **UGSA**. □

4.3 Satisfying All But UGSA: Topology -Dependent Reward Mechanisms (TDRM)

We construct the mechanism in two steps. We first give an intermediate mechanism which manages to satisfy USA, but does not satisfy budget constraint. This preliminary form of the mechanism could be turned into a feasible reward mechanism that satisfies the budget constraint, but doing so would violate Subtree Locality (SL). We then show how we can eliminate the shortcomings of this preliminary mechanism in such a way that both budget constraint and SL are satisfied.

As we discussed in the previous section, the reason why the simple Geometric Mechanism fails the USA property is that it is beneficial for a node to split up and accumulate its own "bubbled up" rewards. This can be avoided by *changing the linear dependency of a node's reward on its own and other node's contribution to a dependency that is of quadratic nature.* Specifically, when computing the reward of a participant u, we multiply u's contribution with the contribution of every node in u's subtree, *including itself.* In this way, even though u could still accumulate "bubbled-up" rewards from its own Sybil nodes, we can show that it is always beneficial for u to focus its total contribution in a single node. The resulting mechanism works as follows.

Algorithm 3: Preliminary Version of TDRM – Not a correct reward mechanism

$R(u) = C(u) \cdot \sum_{v \in T_u} a^{dep_u(v)} \cdot b \cdot C(v)$;

The problem is that while the structure of this quadratic geometric reward mechanism is such that it achieves USA, it is not in fact a feasible mechanism: It fails the budget constraint. On the positive, its structure is such that it does achieve USA. To see why, consider a node u. Suppose u can benefit from splitting itself into a set of Sybil nodes $u_1, \ldots, u_k$, such that $C(u) = \sum_{i=1..k} C(u_i)$. We can re-write the reward of u if it remains a single node as

$$R(u) = C(u)^2 + C(u) \sum_{v \in T_u \setminus u} a^{dep_u(v)} \cdot b \cdot C(v).$$

If it splits itself into Sybil nodes, its new reward is at most

$$R'(u) \leq [C(u_1) + \ldots + C(u_k)] \cdot \sum_{v \in T_u \setminus u} a^{dep_u(v)} \cdot b \cdot C(v) + (C(u_1) + \ldots + C(u_k))^2$$

because the distance between any descendant $v \in T_u$ to any of the Sybil nodes u_i is at least as big as the original distance between u and v in T. Comparing the two expressions, it can be seen that splitting u into multiple nodes $u_1, \ldots, u_k$ does neither increase the first summand (because of the quadratic term), nor the second.

The fundamental problem with this approach is that in order to stay within budget, we would need to scale down the rewards $R(u)$ that are distributed to the participants. However, the amount by which we would need to scale would depend on a global property of the referral tree, for example $C(T)$. Thus, such a scaling would fundamentally violate the SL property. In order to overcome this problem, we would like to constrain the reward a node can obtain. This will allow us to meet the budget constraint by scaling each node's reward by a constant factor, independent of $C(T)$. This could easily be achieved if there was a constant upper bound μ on the contribution $C(u)$ of every node $u \in T$. However, since our model allows a participant to potentially have an unlimited contribution, our mechanism simulates such an upper bound μ by splitting each participant with contribution exceeding μ into a set of nodes, each with contribution at most μ. The mechanism then computes the rewards in the resulting *Reward Computation Tree* (RCT), which may differ from the referral tree. In fact, one user can correspond to multiple nodes in the RCT. A participant's final reward is the sum of the rewards of his corresponding nodes in the RCT.

The effect of computing the rewards in the Reward Computation Tree in this way is that for participants with very large contribution, the algorithm effectively *linearizes* this node's reward with regard to its contribution. In the process, we need to be careful about not violating the USA property. Specifically, in order to make sure that this linearization does not thwart the USA-achieving structure of the quadratic reward computation, the mechanism must be careful about the way it splits participants with large contribution. In particular, our mechanism ensures that for any such split, it is the best possible split for such a participant. In other words, even though the splitting effectively reduces the reward of very large contributors (compared to the preliminary quadratic TDRM mechanism), participants can nevertheless *not benefit from a Sybil attack, because they are already given the best possible split.*

The TDRM mechanism works as follows. Given four parameters $\lambda < \Phi - \phi$, $\mu > 0$, a and b, such that $a + b < 1$, TDRM first transforms the referral tree T into a reward computation tree T', and then computes the rewards on T'. We denote by $C(u)$ and $C'(u)$ the contributions of a node u in T and T', respectively. For a participant $u \in T$, we define a chain CH_u of length N_u in T' as a sequence of nodes $m_1^u, \ldots, m_{N_u}^u$, such that m_i^u is the parent node of m_{i+1}^u, for all $i = 1...N_u - 1$. We call m_1^u and $m_{N_u}^u$ the head and the tail of the chain, respectively.

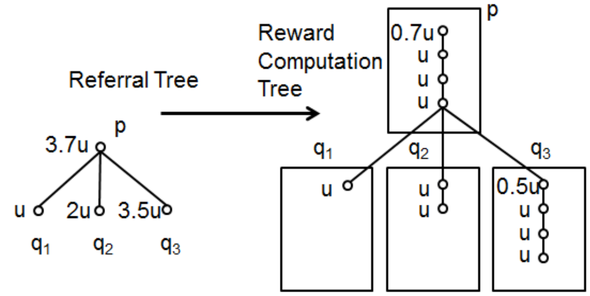

Figure 3: Transformation of a referral tree T into a reward computation tree T' by TDRM.

Algorithm 4: TDRM Mechanism

Transformation of T into T':
for $u \in T$ **do**
  $N_u = \lceil C(u)/\mu \rceil$;
  Create a chain CH_u of length N_u in T', such that
  $C'(m_i^u) = \begin{cases} C(u) - (N_u - 1)\mu & \text{, if } i = 1 \\ \mu & \text{, if } i > 1 \end{cases}$;
end
for *Every directed edge* $(u, v) \in T$ **do**
  Create a directed edge $(m_{N_u}^u, m_1^v)$ in T';
end
for $w \in T'$ **do**
  $R'(w) = \frac{\lambda}{\mu} C(w) \sum_{x \in T'_w} a^{dep_w(x)} b \cdot C(x) + \phi C(w)$;
  – *Reward Calculation in T'*
end
for $u \in T$ **do**
  $R(u) = \sum_{v \in CH_u} R'(v)$;
  – *Reward Calculation in T*
end

Figure 3 gives an example of how the mechanism transforms the referral tree T (left) into a corresponding reward computation tree T' (right). After this transformation, TDRM first computes the rewards for each node in T' according a function similar to the one given in the preliminary TDRM mechanism. Finally, the reward of a participant $u \in T$ is computed as the sum of all the nodes in the corresponding chain CH_u in T'. It remains to show that the mechanism meets the budget constraint – we do this in the next section. With this, we can prove the following key theorem.

THEOREM 4.4. *The TDRMmechanism with parameters $\lambda < \Phi - \phi$, $b < 1 - a$, and $\mu > 0$ achieves all desirable properties except UGSA.*

PROOF. It will be convenient to use the following definition. Let S_A, S_B be two subsets of T'. We define

$$B(S_A, S_B) = \sum_{u \in S_A} \sum_{v \in S_B} \frac{\lambda}{\mu} b \cdot C(u) C(v) a^{dep_u(v)}.$$

Intuitively, $B(S_A, S_B)$ is the sum of the rewards accumulated by nodes in S_A through nodes in S_B. Using this definition, we can reformulate the reward function $R(u)$ for $u \in T$ as $R(u) = B(CH_u, T'_{m_1^u}) + \phi C(u)$.

Budget Constraint: We start by proving that the mechanism meets the budget constraint. First, observe that the total rewards in the referral tree is equivalent to the total rewards in the reward computation tree. Then, in the reward computation tree T', it holds that

$$\begin{aligned} \sum_{u \in T'} R(u) &= \sum_{u \in T'} [C(u) \cdot \frac{\lambda}{\mu} \sum_{v \in T'} a^{dep_u(v)} \cdot bC(v) + \phi C(u)] \\ &< \sum_{v \in T'} [\lambda \cdot C(v) \sum_{i=0}^{\infty} a^i b] + \sum_{u \in T'} \phi C(u) \\ &< (\lambda + \phi) \sum_{u \in T'} C(u). \end{aligned}$$

By the constraint imposed on λ, this last expression is at most $\Phi \sum_{u \in T'} C(u)$, which is the budget.

We now prove the desirable properties one by one.

CCI: Consider a participant u, who increases his contribution from $C(u)$ to $C^*(u) = C(u) + \epsilon$. Let $R^*(u)$ and $CH_u^* = \{m_1^{*u}, \ldots, m_{N_u^*}^{*u}\}$ denote the new reward and the new corresponding chain, respectively. There are two cases depending on whether u's contribution increase leads to a change of its corresponding chain CH_u^* in the RCT, or not. We consider the two cases independently.

First, if $N_u^* = N_u$, then only the head-node m_1^u's contribution increases in T': $C(m_1^{*u}) = C(m_1^u) + \epsilon$. We can get $R^*(u) > R(u)$.

Second, if $N_u^* > N_u$, then we only need to consider the sub-chain in CH_u^* with N_u nodes from the leaf node up. As each node of the sub-chain has contribution μ, we get that $R^*(u) > R(u)$.

ϕ-RPC: By the definition of the $R(u)$, it holds that $R(u) = B(CH_u, T_{m_1^u}) + \phi C(u) > \phi C(u)$.

CSI: The property holds because by the definition of $R(u)$, the reward of a participant u is strictly increasing when a new node v attaches to u.

SL: The property holds because by the definition of $R(u)$, the reward of a participant u is independent of any node outside of T_u.

URO: Consider a participant u, whose contribution is $C(u) = s\mu + \epsilon$, for some integer s and $0 < \epsilon \leq \mu$, and suppose u has k children in the referral tree, namely there are k trees attached to u. Here s can be any non-negative integer and k can be any positive integer. We denote one of u's children as v and the corresponding subtree as T_v. Suppose v has ℓ children with contribution μ. It holds that $R(u)$ is at least $R'(m_{N_u}^u)$ in the reward computation tree. Calculating the value of $R'(m_{N_u}^u)$ using the definition, it can be shown that $R'(m_{N_u}^u) \geq \ell \cdot a^2 b \lambda \epsilon$. As ℓ can become arbitrarily large, the reward of $R(u)$ can increase to infinity.

USA: At the heart of our proof is that TDRM satisfies USA. To do so, we define an $\epsilon-chain$ as a chain in the reward computation tree of which only the head node can have contribution less than μ.

USA states that no participant can increase his reward by pretending to have multiple identities. Consider a participant u that joins the referral tree as j Sybil nodes ($j \geq 1$), $v_1, v_2, ..., v_j$, with total contribution $C(u)$. Further assume that u has s children, $q_1, \cdots, q_s$. Suppose $v_1, v_2, ..., v_j$ are transformed into k nodes $u_1, \cdots, u_k$ in the reward computation tree. By definition, it holds that $C(u) = \sum_{i=1}^{k} C(u_i)$ and $C(u_i) \leq \mu, i = 1, \cdots, k$. For $q_1, \cdots, q_s$, we denote the subtrees rooted at $q_1, \cdots, q_s$ in the reward computation tree

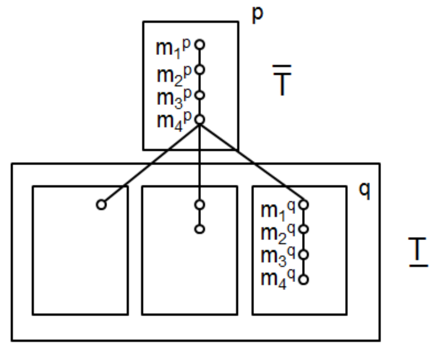

Figure 4: Illustration of notation used in the proof.

as $T_1, \cdots, T_s$. We define a *partition* as any configuration of nodes $u_1, \cdots, u_k$, subtrees $T_1, \cdots, T_s$, and contributions $C(u_i)$, $(i = 1, \cdots, k)$ in the reward computation tree that can feasibly result from node u joining the referral tree as a set of multiple Sybil nodes.

Our proof idea is the following: Consider the set of optimal partitions for u in the reward computation tree (partitions maximizing $R(u)$). We show that at least one optimal partition has the structure of a single $\epsilon - chain$ in the RCT. In other words, we show that u's best possible Sybil attack is to join in such a way that the resulting structure in the RCT is an $\epsilon - chain$. However, since the *TDRM mechanism transforms u into an $\epsilon - chain$ in the RCT even if u joins as a single node*, it follows that u has no benefit of joining the referral tree as multiple Sybil identities. The mechanism itself will give u the best possible split, thus giving u no incentive to split itself.

We formally prove this intuition by a sequel of structural lemmas. The lemmas describe the properties of a reward-maximizing partition $u_1, \cdots, u_k, T_1, \cdots, T_s$ in the RCT, ultimately showing that the optimal such partition is an $\epsilon - chain$. As a first step, notice that because we have proven SL in TDRM, we consider only $u_1, \cdots, u_k, T_1, \cdots, T_s$ in the RCT. All other nodes are irrelevant for u's reward. The first lemma shows that $u_1, \cdots, u_k, T_1, \cdots, T_s$ forms a tree. Here notice that according to the soliciting sequence, u_i can not be a child of T_j $(i = 1, 2..., k,\ j = 1, 2, ..., s)$.

LEMMA 4.5. *If $R(u)$ is maximized, $u_1, \cdots, u_k, T_1, \cdots, T_s$ forms a tree.*

PROOF. We prove the lemma by contradiction. Suppose $R(u)$ is maximized and $u_1, \cdots, u_k, T_1, \cdots, T_s$ forms a forest F with more than one tree. We pick any two trees T_α, T_β in F with roots α and β. As u is the parent of $q_1, \cdots, q_s$, it holds that $T_1, \cdots, T_s$ will be attached as subtrees to $u_1, \cdots, u_k$. Thus, $\alpha, \beta \in \{u_1, \cdots, u_k\}$. Now, assume that we attach T_β to α, thereby makeing it one tree. The attachment does not change the reward accumulated by nodes in T_β, but it strictly increases the rewards accumulated by α (due to the CSI property). This contradicts the assumption that $R(u)$ is maximized. □

Thus if $R(u)$ is maximized, $u_1, \cdots, u_k$, $T_1, \cdots, T_s$ forms a tree. We denote this tree as T_u, and define $\overline{T_u}$ as the tree induced by $u_1, \cdots, u_k$, and $\underline{T_u}$ as the forest induced by $T_1, \cdots, T_s$. With these definitions, we can write $R(u)$ as

$$R(u) = B(\overline{T_u}, \overline{T_u}) + B(\overline{T_u}, \underline{T_u}) + \phi C(u).$$

Before continuing the proof, we distinguish different parts of $R(u)$: The *inner reward* $R^i(u) = B(\overline{T_u}, \overline{T_u})$ which is the part of reward purely coming from u's own contribution, and the *external reward* $R^e(u) = B(\overline{T_u}, \underline{T_u})$ which is the part of reward coming from u's descendants. Then we can rewrite $R(u)$ as $R(u) = R^i(u) + R^e(u) + \phi C(u)$. According to our assumption that u has a fixed contribution, the third term $\phi C(u)$ is a constant and does not influence u's decision.

As mentioned before, we need to prove that the best partition of u, maximizing the reward, is an $\epsilon-chain$. Concretely, as $R(u) = R^i(u)+R^e(u)+\phi C(u)$, we show that u can maximize $R^i(u)$ and $R^e(u)$, respectively, if $\overline{T_u}$ is an $\epsilon - chain$. Our next step is to prove u's partition as an $\epsilon - chain$ will maximize $R^i(u)$. We transform the topology of $\overline{T_u}$ step by step. The lemma below shows that if u wants to maximize his inner reward $R^i(u)$ at most one node in $\overline{T_u}$ can have contribution less than μ.

LEMMA 4.6. *If $R^i(u)$ is maximized, there can be at most one node $v \in \overline{T_u}$ with contribution $C(v) < \mu$.*

PROOF. We prove the lemma by contradiction. Suppose there is more than one node with contribution less than μ. We denote two such nodes as x, y, i.e., $x, y \in \overline{T_u}$ with $C(x) < \mu$ and $C(y) < \mu$. Let $S_u = \overline{T_u} \setminus \{x, y\}$, and let P_x, P_y be the set of ancestors of x, y in the reward computation tree. The inner reward of u is $R^i(u) = B(\overline{T_u}, \overline{T_u})$
$= B(\{x,y\}, S_u)+B(S_u, S_u)+B(\{x,y\}, \{x,y\})+B(S_u, \{x,y\})$.

To simplify the calculation, we define a function $\gamma_p(S) = \sum_{v\in S} \frac{b\lambda}{\mu} C(v) \max\{a^{dep_p(v)}, a^{dep_v(p)}\}$ for any node p in $\overline{T_u}$. According to the definition of $B(\cdot)$, it can be shown that

$$\begin{aligned} B(\{x,y\}, S_u) &= B(x, S_u) + B(y, S_u) \\ &= C(x)\gamma_x(T_x \setminus \{x,y\}) + C(y)\gamma_y(T_y \setminus \{x,y\}), \\ B(S_u, \{x,y\}) &= B(S_u, x) + B(S_u, y) \\ &= C(x)\gamma_x(P_x \setminus \{x,y\}) + C(y)\gamma_y(P_y \setminus \{x,y\}), \\ B(\{x,y\}, &\{x,y\}) \\ &= \tfrac{b\lambda}{\mu}[(a^{dep_x(y)} + a^{dep_y(x)})C(x)C(y) + C(x)^2 + C(y)^2]. \end{aligned}$$

Expanding $R^i(u)$ and combining the above bounds, we get

$$\begin{aligned} R^i(u) = &C(x)\gamma_x((P_x \cup T_x) \setminus \{x,y\}) \\ &+C(y)\gamma_y((P_y \cup T_y) \setminus \{x,y)\}) \\ &+\tfrac{b\lambda}{\mu}[(a^{dep_x(y)} + a^{dep_y(x)})C(x)C(y) + C(x)^2 + C(y)^2] \\ &+B(S_u, S_u). \end{aligned} \quad (1)$$

Without loss of generality, suppose $\gamma_x((P_x \cup T_x) \setminus \{x,y\}) \geq \gamma_y((P_y \cup T_y) \setminus \{x,y\})$. Then, consider two cases:

a) If $C(x)+C(y) > \mu$, we can change $C(x)$ to μ and $C(y)$ to $C(x) + C(y) - \mu$.

b) If $C(x)+C(y) \leq \mu$, we can change $C(x)$ to $C(x)+C(y)$ and $C(y)$ to 0.

In both cases, the change does not have an impact on the total contribution, but it increases $R^i(u)$. Specifically, the sum of the first two expressions in (1) will increase due to the change. Then, using the fact that if for two reals A and B with $A > B, 0 < t < \frac{A-B}{2}, k < 2$ and $S_1 = A^2+B^2+kAB$ and $S_2 = (A-t)^2+(B+t)^2+k(A-t)(B+t)$, it holds that $S_1 > S_2$, it follows that the third expression in (1) also increases. Meanwhile, the forth expression is unchanged. This leads to a contradiction because it means that this hypothetic partition does not maximize the inner reward. □

Next, we characterize the *location* of the at most one node in $\overline{T_u}$ that has contribution less than μ. Due to space limitations, we omit full proofs, and instead sketch the main proof ideas.

LEMMA 4.7. *If $R^i(u)$ is maximized, $\overline{T_u}$ is an $\epsilon - chain$ or a chain in which only the leaf node has contribution less than μ.*

`Proof Sketch.` According to Lemma 4.6, if $R^i(u)$ is maximized, there is at most one node with contribution less than μ in $\overline{T_u}$. We call it $\epsilon - node$ and suppose its contribution is $\epsilon(< \mu)$. (Here we need to pay attention that the $\epsilon-node$ has contribution strictly less than μ.) We can prove the lemma by case analysis and contradiction.

a) Suppose $\overline{T_u}$ is not a chain. We distinguish three subcases.

a1) Suppose in $\overline{T_u}$, there is an $\epsilon-node$ and the $\epsilon-node$ is not a leaf node. We denote the $\epsilon-node$ as x with $C(x) = \epsilon$, and denote the leaf node which is a descendent of x as y with $C(y) = \mu$. If we change $C(x)$ to μ and $C(y)$ to ϵ, $R^i(u)$ increases which contradicts the assumption.

a2) Suppose in $\overline{T_u}$, there is an $\epsilon-node$ and the $\epsilon-node$ is a leaf node. In this case, $\overline{T_u}$ has at least two leaf nodes. At least one leaf node denoted as x has contribution μ. Then we can delete x, add a new node y with contribution $C(y) = \mu$ and make the remaining tree $\overline{T_u} \setminus \{x\}$ attached as a subtree to y. $R^i(u)$ will increase.

a3) Suppose in $\overline{T_u}$, there is no $\epsilon-node$. The proof method is the same as that in a2).

b) Suppose $\overline{T_u}$ is a chain and there is an $\epsilon - node$ which is neither the root nor the leaf of the chain. We denote the $\epsilon-node$ as x and the leaf of the chain as y. We can increase $R^i(u)$ by changing $C(x)$ to μ and changing $C(y)$ to ϵ which contradicts the assumption. □

Thus we have shown that an $\epsilon-chain$ in the reward computation tree maximizes $R^i(u)$. We will now prove that u's partition as an $\epsilon-chain$ also maximizes his external reward $R^e(u)$. The next lemma shows it is better to root each tree in $\underline{T_u}$ to one leaf node in $\overline{T_u}$.

LEMMA 4.8. *For any given topology $\overline{T_u}$, suppose $u_1, u_2, ...,$ u_k are the nodes in $\overline{T_u}$. There exists a partition that maximizes $R^e(u)$ in which each tree in $\underline{T_u}$ is attached to a single node u_i, for some $i = 1, 2, ..., k$.*

`Proof Sketch.` We denote the trees in $\underline{T_u}$ as $T_1, \cdots, T_s$. According to the definition of the external reward, it holds that

$$R^e(u) = B(\overline{T_u}, \underline{T_u}) = \sum_{i=1}^{s} B(\overline{T_u}, T_i).$$

Suppose that by attaching T_1 to u_t $(1 \leq t \leq k)$, $B(\overline{T_u}, T_1)$ can be maximized. The proof works by showing that by attaching each tree T_i $(i = 1, 2, ...s)$ to u_t, we can maximize $R^e(u)$. □

We now know that $R^e(u)$ can be maximized when each tree in $\underline{T_u}$ is attached to a single node in $\overline{T_u}$. For any given $\overline{T_u}$, in order to maximize $R^e(u)$, we thus only need to consider partition in which each tree in $\underline{T_u}$ is attached to some node u^* in $\overline{T_u}$. Then using this property, we show that an $\epsilon - chain$ is the best partition for maximizing u's external reward.

LEMMA 4.9. *If $R^e(u)$ is maximized, $\overline{T_u}$ must be an $\epsilon -$ chain and u^* is the leaf node of $\overline{T_u}$.*

`Proof Sketch.` The first step is to show that if $R^e(u)$ is maximized, $\overline{T_u}$ must be a chain and u^* is a leaf node in $\overline{T_u}$. We prove it by contradiction. Suppose $\overline{T_u}$ is not a chain or u^* is not a leaf node in $\overline{T_u}$. Then we find that there exists a leaf node u_L in $\overline{T_u}$ which is not u^*. As no tree in $\underline{T_u}$ is attached to u_L, it holds that $B(u_L, \underline{T_u}) = 0$. We delete u_L in $\overline{T_u}$ and relocate u_L to be the root of $\overline{T_u} \backslash u_L$. The external reward of u will increase due to this change. So if $R^e(u)$ is maximized, $\overline{T_u}$ must be a chain and u^* is a leaf node in $\overline{T_u}$.

Our next step is to show $\overline{T_u}$ is an $\epsilon - chain$. We also prove it by contradiction. Suppose $\overline{T_u}$ is a chain but not an $\epsilon - chain$ and u can get the maximum external reward. Then there exists a node x which is not the root node of $\overline{T_u}$ and has contribution $C(x) < \mu$. As x is not the root, we denote x's parent as y. Then we find that if we change $C(x)$ to $C(x) + \alpha$ and $C(y)$ to $C(y) - \alpha$, (The constraints are $\alpha < \mu - C(x)$ and $\alpha < C(y)$; we can take very small α.) u can get higher external reward. This establishes the contradiction. Thus, if u wants to maximize $R^e(u)$, he must join the referral tree in such a way that $\overline{T_u}$ results in an $\epsilon - chain$, and u^* is the leaf node of $\overline{T_u}$. □

With this, we are now in a position to complete the proof. By Lemmas 4.7 and 4.9, we know that the partition which makes $\overline{T_u}$ an $\epsilon - chain$, and in which all trees in $\underline{T_u}$ are attached to the tail node of $\overline{T_u}$, can maximize both $R^i(u)$ and $R^e(u)$. According to the definition that $R(u) = R^i(u) + R^e(u) + \phi C(u)$, we can infer that such a partition thus maximizes $R(u)$. However, if the participant u simply joins the referral tree as a single, non-Sybil node with its entire contribution, TDRM will automatically also transform u partition into the same $\epsilon - chain$ in the reward computation tree. Thus, u has no benefit from joining as multiple identities, which proves USA.

Example : To show that TDRM does indeed violate UGSA, consider the following counter-example. Let u be a participant with $C(u) = \frac{1}{2}\mu$ and let $v_1, \cdots, v_k$ be u's children with $C(v_1) = \cdots = C(v_k) = \mu$ $(k > \frac{1}{ab\lambda})$. The profit of u as computed by TDRM is $P(u) = \frac{1}{2}((ak+1)\lambda\mu b + \phi\mu - \mu)$. If we increase u's contribution to $C'(u) = \mu$, then we can show that the new profit of u is $P'(u) = R'(u) - C'(u) = (ak+1)\lambda\mu b + \phi\mu - \mu$, which is larger than $P(u)$. That is, by increasing his contribution u can increase his profit, which violates UGSA.

4.4 Satisfying All But URO: Contribution-Deterministic Reward Mechanisms

Given the impossibility results in Theorem 4.3, we cannot expect to achieve a mechanism that achieves all the desirable properties defined in this paper, in particular, we cannot hope to simultaneously achieve UGSA and URO. The TDRM mechanism in the previous section has achieved all, but UGSA. In this section, we show that we can also relax the other property, URO, and satisfy instead all the remaining properties. For this, however, entirely different algorithmic techniques are required.

The key idea is that whereas the previously discussed mechanisms are *topology-dependent* (i.e., the reward is among other things a function of the structural property of a node's descendant tree), we now consider mechanisms in which the reward of a participant u is independent of the topology of its subtree. In particular, we seek mechanisms in which the reward $R(u)$ is purely a function of u's own contribution and the *sum* $\sum_{v \in T_u} C(v)$ of the contributions in T_u. We show that this can yield a family of mechanisms that achieve UGSA, albeit at the cost of URO.

For ease of notation, define $x_p = C(p)$ and $y_p = C(T_p \backslash \{p\})$ for a participant $p \in T$. Then, we want that the reward function $R(p)$ is purely a function of x_p and y_p. What properties should this function $R(x_p, y_p)$ have in order to satisfy the desirable properties? The SL constraint is automatically satisfied by the definition of $R(x_p, y_p)$. The CCI property demands that $R(x_p, y_p)$ is increasing in x_p, i.e. $0 < \frac{dR(x_p,y_p)}{dx_p}$. In order to satisfy CSI, it should hold that an increase in y_p increases p's reward, hence $0 < \frac{dR(x_p,y_p)}{dy_p}$. If we want to globally ensure the budget constraint, one way to do this is to demand that $R(x_p, y_p) < \Phi x_p$, and similarly, the $\varphi - RPC$ property can be enforced by $\phi x_p < R(x_p, y_p)$. It is important to point out that demanding the budget constraint to be satisfied by means of $R(x_p, y_p) < \Phi x_p$ implies that we cannot achieve the unbounded reward property URO. The reason is that if URO were to be satisfied, $R(x_p, y_p)$ would need to be able to grow larger and larger as y_p increases, which would eventually violate this constraint. In order to also achieve USA, we need the condition that for any x'_p, x''_p such that $x'_p + x''_p = x_p$, it holds that $R(x_p, y_p) \geq R(x'_p, x''_p + y_p) + R(x''_p, y_p)$, and, finally, in order to achieve UGSA (under the assumption that we already have USA satisfied), we only need $\frac{dR(x_p,y_p)}{dx_p} < 1$.

Combining these observations, we can demand that a function $R(x_p, y_p)$ satisfy four properties. If it satisfies all of them, we call the function *successfully contribution -deterministic*. The properties are, for any $x_p > 0$, y_p:

$$i)\ 0 < \frac{dR(x_p, y_p)}{dx_p} < 1, \quad ii)\ 0 < \frac{dR(x_p, y_p)}{dy_p},$$
$$iii)\ \phi x_p < R(x_p, y_p) < \Phi x_p,$$
$$iv)\ R(x_p, y_p) \geq R(x'_p, x''_p + y_p) + R(x''_p, y_p),$$

for any x'_p, x''_p such that $x'_p + x''_p = x_p$.

THEOREM 4.10. *If $R(x_p, y_p)$ is a successfully contribution-deterministic function, then the reward mechanism that distributes rewards according to $R(x_p, y_p)$ achieves all properties, except URO.*

PROOF. The proof follows closely along the lines of how the properties are defined. The SL constraint is obviously satisfied. CCI is satisfied because $R(x_p, y_p)$ is increasing in x_p (Property i); CSI is satisfied because $R(x_p, y_p)$ is increasing in y_p (Property ii); and both $\phi - PPC$ and the budget constraint are clearly satisfied because of Property iii.

We prove that USA is satisfied by contradiction. Suppose there is a participant p that can maximize his reward by joining the system as $k \geq 2$ nodes, and assume that the cardinality k is minimal among all those maximal splits. Consider two of these Sybil nodes p_1 and p_2, and define $x_1 = C(p_1)$, $x_2 = C(p_2)$, $y_1 = C(T_{p_1}) - C(p_1)$ and $y_2 = C(T_{p_2}) - C(p_2)$. There are two cases:

a) p_1 is an ancestor of p_2 (or vice versa). Then we know that $y_1 \geq x_2 + y_2$, $0 < \frac{dR(x_p,y_p)}{dy_p}$, so for any x_p and y_p,

$$R(x_1, y_1) + R(x_2, y_2) \leq R(x_1, y_1) + R(x_2, y_1 - x_2).$$

According to Property iv defined above, we know that

$$R(x_1, y_1) + R(x_2, y_1 - x_2) \leq R(x_1 + x_2, y_1 - x_2).$$

Combining these two expressions implies that the following inequality holds:

$$R(x_1, y_1) + R(x_2, y_2) \leq R(x_1 + x_2, y_1 - x_2).$$

This means that p can get at least the same reward by merging p_1 and p_2 into one node, which contradicts our assumption.

b) p_1 is not an ancestor of p_2 (or vice versa). According to Property iv, it holds that

$$R(x_1 + x_2, y_1 + y_2) \geq R(x_1, y_1 + y_2 + x_2) + R(x_2, y_1 + y_2)$$

$$> R(x_1, y_1) + R(x_2, y_2).$$

Like in case a), this implies that p can get at least the same reward by merging p_1 and p_2 which contradicts our assumption. This concludes the proof that USA is satisfied.

Finally, we prove that UGSA is satisfied. Consider some participant p. We need to compare two cases. In the first case, p joins the system as k nodes, $p_1, ..., p_k$. In the second case, p joins the system as a single node. In order to prove UGSA, we need to show that for any k and any $\Sigma_{i=1}^{k} C(p_i)$ which is equal to or larger than $C(p)$, in the second case, p can get higher payoff, namely $\Sigma_{i=1}^{k}(C(p_i) - R(C(p_i), C(T_{p_i} \backslash p_i))) \geq C(p) - R(C(p), C(T_p \backslash p))$. According to the USA property, we know that any participant p with a fixed cost can get the highest reward by joining the system as single node. Therefore, we can assume that there is an optimal choice in the scenario in which $k = 1$.

It remains to prove that for any $\epsilon > 0$, it holds $x_p - R(x_p, y_p) < x_p + \epsilon - R(x_p + \epsilon, y_p)$. According to Property i, we know that for any x_p, y_p,

$$\frac{dR(x_p, y_p)}{dx_p} < 1.$$

Therefore, it follows that for any $\epsilon > 0$,

$$R(x_p + \epsilon, y_p) - R(x_p, y_p) < \epsilon$$

$$\Rightarrow x_p - R(x_p, y_p) < x_p + \epsilon - R(x_p + \epsilon, y_p).$$

As $\epsilon > 0$, the total profit decreases, which implies that UGSA is satisfied. □

4.4.1 CDRM Mechanisms

The properties derived in the previous section imply a family of reward mechanisms all of which achieve all properties except URO. It remains to find specific, practical functions that belong to this family. In this section, we give two examples. First, we set $R(x_p, y_p) = f(x_p, y_p)x_p$, so that the reward function is proportional to x_p.

Algorithm 5: Two examples of a CDRM Mechanism

i) $R(p) = (\Phi - \frac{\theta}{1+x_p+y_p})x_p$, for $\theta + \phi < \Phi$

ii) $R(p) = \Phi x_p + \theta \ln \frac{1+y_p}{x_p+y_p+1}$, for $\theta + \phi < \Phi$

In both cases, it is easy to verify that the reward function does satisfy all the properties stated in the theorem. Hence, both CDRM mechanisms satisfy all our desirable properties, except URO.

5. CONCLUSIONS

In this work, we have studied incentive tree mechanisms, thus formalizing and generalizing previous algorithmic work on referral trees, lottery trees [9, 6] and multi-level marketing mechanisms [10][11]. We design two families of incentive tree mechanisms, both of which achieve all but one among the set of axiomatic properties. Furthermore, our impossibility result suggests that this is optimal. We are encouraged that both of these mechanisms achieve the slightly weaker notion of unprofitable Sybil attack (USA). This shows that mechanisms can be designed that are provably resilient against basic forms of multi-identity attacks.

Any axiomatic approach based on a choice of desirable properties is questionable as different people may deem different properties to be more important. Indeed, as we point out, not all of the properties are equally relevant to the successful operation of an incentive tree scheme in practice. However, in ongoing work, we have been studying the effect of our mechanisms in practical deployments; and experience has strengthened our belief the properties defined in this paper are indeed of critical practical importance.

6. REFERENCES

[1] http:\\ www.gigwalk.com.
[2] http:\\ www.mturk.com.
[3] M. Tennenholtz. Convention Evolution in Organizations and Markets. *Computational and Mathematical Organization Theory*, 2, 1996.
[4] P. Domingos and M. Richardson. Mining the Network Value of Customers. In *Proc. of SIGKDD*, 2003.
[5] J. Kleinberg and P. Raghavan. Query Incentive Networks. In *Proc. of FOCS*, 2005.
[6] G. Pickard, W. Pan, I. Rahwan, M. Cebrian, R. Crane, A. Madan, and A. Pentland. Time Critical Social Mobilization. *Science*, 2011.
[7] J. Douceur. The Sybil Attack. In *Proc. of IPTPS*, 2002.
[8] A. Cheng and E. Friedman. Sybilproof Reputation Mechanism. In *Proc. of SIGCOMM*, 2005.
[9] J. Douceur and T. Moscibroda. Lottery Trees: Motivational Deployment of Networked Systems. In *Proc. of SIGCOMM*, 2007.
[10] Y. Emek, R. Karidi, M. Tennenholtz, and A. Zohar. Mechanisms for Multi-Level Marketing. In *Proc. of 12th ACM Conference on Electronic Commerce (EC)*, 2011.
[11] F. Drucker and L. Fleischer. Simpler Sybil-Proof Mechanisms for Multi-Level Marketing. In *Proc. of 13th ACM Conference on Electronic Commerce (EC)*, 2012.
[12] M. Babaioff, S. Dobzinski, S. Oren, and A. Zohar. On Bitcoin and Red Balloons. In *Proc. of 13th ACM Conference on Electronic Commerce (EC)*, 2012.
[13] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing. In *Proc. of Mobicom*, 2012.
[14] A. Rai, K. K. Chintalapudi, V. Padmanabhan, and R. Sen. Zee: Zero-Effort Crowdsourcing for Indoor Localization. In *Proc. of Mobicom*, 2012.
[15] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No Need to War-Drive: Unsupervised Indoor Localization. In *Proc. of MobiSys*, 2012.