Cluster Computing (2022) 25:2241–2270
https://doi.org/10.1007/s10586-021-03411-3

# SPIDE: sybil-proof, incentivized data exchange

Rafał Skowroński[1] · Jerzy Brzeziński[1]

Received: 15 February 2021 / Revised: 14 July 2021 / Accepted: 1 September 2021 / Published online: 20 November 2021


**Abstract**
Decentralized, open-access blockchain systems opened up new, exciting possibilities—all without reliance on trusted third parties. Regardless of the employed consensus protocol, the overall security, decentralization and effectiveness of such systems, largely depend upon properly structured incentives. Indeed, as has been previously spotted by Babaiaff et al. Bitcoin-like systems, oftentimes lack some of these. Specifically, current blockchain-systems fail to incentivize one of their crucial aspects–the underlying data exchange. As we rationalize, proper incentivization of that layer could lead to lower transactions' confirmation-times, improved finalization guarantees and at the same time to discouragement of malicious behaviours such as block-withholding attacks. Indeed, incentivization of the data-exchange layer allows the system to remain operational when all agents, including routing nodes, are assumed to be rational. In this work, while focusing on the problem of sybil-proof data exchange, we revisit previous approaches, showcasing their shortcomings and lay forward the first information exchange framework; with integrated routing and reward-function mechanics, provably secure in thwarting Sybil-nodes in 1-connected or eclipsed networks. The framework neither requires nor assumes any kind of constraints in regard to the network's topology (i.e. the network is modelled as a random-connected graph) and rewards information propagators through a system-intrinsic virtual asset maintained by the decentralized state-machine. The proposal, while being storage and transmission efficient is suitable for rewarding not only consensus-related datagrams (both data-blocks and transactions) but consensus-extrinsic information as well, thus facilitating an universal sybil-proof data-exchange apparatus, provably valid under the assumption of existence of a data store whose property of non-malleability emerges as time approaches infinity. Our research was conducted under two scenarios—with round leader known and unknown in advance of each transactional round.

**Keywords** Blockchain · Networking · Decentralization · Peer to peer

## 1 Introduction

Recently, cryptographic protocols allowing for a decentralized consensus, opened-up new, exciting possibilities. Their attractiveness does not stem from new performance breakthroughs. On the contrary—they may be orders of magnitudes slower than their centralized counterparts. Still, these protocols exhibit a remarkable quality—namely, their ability to carry out decentralized Turing-complete computations without imposing trust requirements among the involved agents. In such an open, widely deployed ecosystems, when it comes to the topmost State-Domain [1], we may need to trust the majority, but never a single, or easily countable number of system-intrinsic agents.

Indeed, in [1] we proposed a new, exciting family of Open-Blockchain Aided Multi-Agent Cyber-Physical Systems (OBAMA-CPS), highlighting new possibilities, while depicting limitations, shortcomings and laying forward some of the essential guidelines. We highlighted that resilience of incentivized, decentralized data-exchange could be used to aid innovative autonomous environments by focusing on the case the power-industry in [2]. The sole existence of such decentralized systems would not be possible without one utmost important, yet easily overlooked, probably due to its proliferation and ubiquity aspect—the underlying data exchange. In this paper we put our attention towards incentivization of data-exchange throughout decentralized, cryptographically secure decentralized state machines, oftentimes colloquially nicknamed

✉ Rafał Skowroński
rafal.skowronski@put.poznan.pl

[1] Poznan University of Technology, Poznan, Poland

as open-ledger systems or 'blockchains'. Here, besides analysing the system's intrinsic, consensus-related data-exchange (data blocks and transactions) we look at how a decentralized state-machine could be used to incentivize propagation of *consensus-extrinsic* information as well. Indeed, from such a perspective this effectively constitutes the problem of facilitating incentivized data-transmission in any open computer network, under the premise of existence of a decentralized non-volatile data store whose property of non-malleability emerges as time approaches infinity. As we further explain, proper incentivization of data exchange could benefit anyone. Leading to better qualities of the overall decentralized environment.

With that said, the proposed mechanics would need to assure fair assignment of rewards,—imposing verifiability and accountability of the traversed data paths. On top of everything, the solution would need to be sybil-proof. The protocol needs to assure that attempts to cheat the system, by gaining unwarranted rewards, fall short as counterproductive and effectively end up as an exercise in futility. Further, the solution should allow for a feasible implementation and exhibit acceptable transmission and data-storage overheads. Here, nodes are either rewarded or penalized through a virtual, transitive asset—one maintained by the decentralized state-machine. Throughput the paper we will be steadily introducing reader to data-propagation algorithms. These are used for ensuring fair incentivized, Sybil-proof, both system intrinsic (transactions/data blocks) and system-extrinsic data exchange. Each algorithm comes with precise assumptions and a Threat Model. We shall finish by providing the reward assignment function and proving it to be Sybil-proof.

## 1.1 Rationalization of why 'sybil-nodes' are bad

We assume that a legitimate data-path is one which does not include any sybil identities (Fig. 1), thus upholding a one-to-one relationship between an agent and its virtual, logical identity.

In a decentralized environment, requiring routers to be compensated for their efforts demands a decentralized reward-apparatus. Under the premise of agents being driven by profit, it is logical to presume they would be trying to deceive others, especially ones conducting assignment of rewards, into believing in a distorted state of the system—a state which would yield the dishonest—rewards higher than expected from an upright obedience to the protocol. Apparently, rational, yet unfair behaviour could come at the cost of the others—was the reward pool to be finite.

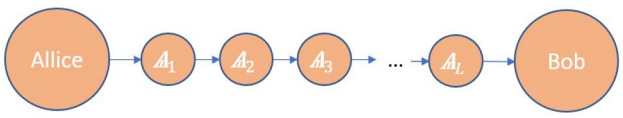



**Fig. 1** Sample legitimate data path with no Sybil-nodes

Now, let us imagine the remuneration apparatus to be pretty naïve. Each agent is to obtain a share from a finite reward pool ($R_N$) based on its identifier—one added to the message's (M) endorsement (E) once published. Now, looking at the figure below, suppose the second agent becomes '*overly*' greedy, and so to receive an unjust extra reward, he injects an additional mischievous identifier into the endorsement (E) accompanying M, resulting in a situation depicted in (Fig. 2).

Now, once M arrives at the agent responsible for assignment of rewards, meaning one capable of affecting the topmost State-Domain [1], the agent looks at the attached endorsement (E) and distributes rewards based on identifiers included within it. With our limited assumptions in mind, by looking at (Fig. 2) it becomes apparent that agent $(\Lambda)$ operating the second router introduced an artificial identity (denoted by $\mathbb{A}_i$ allowing for collection of an additional reward for identity $\mathbb{A}_{2'}$—thus effectively collecting rewards for both of the owned identities $\mathbb{A}_2$ and $\mathbb{A}_{2'}$. Notice how this wickedly reduced portion of the reward pool available to others (agents owning identities $\mathbb{A}_i \forall i \geq 3$). This simplistic example illustrates precisely why 'sybil nodes' are 'bad'. They induce a distorted, artificial perspective onto the state of the system, without improving network's connectivity. Such an unjust yet possibly rational misbehaviour of falsifying the world-view as perceived by others, affects both computations and decision making, here resulting in an unfair spread of rewards. One may conclude that this could lead to messages not being delivered at all—shall the remainder of reward-pool dwindle below a certain 'attractiveness' threshold as seen by further intermediaries. Now, since we require the protocol to be game-theoretically sound—its operational logic needs to discourage inclusion of unrighteous, artificial identities. Further, we shall require the mechanics to be 'sybil-proof' under the general assumption of agents willing to maximize profits.

### 1.1.1 Routing (why integrated)

When designing data-exchange protocols for decentralized environments, aiming to maintain the property of *sybil-proofness*, one needs to recognize whom to reward and/or whom to penalize. Let us notice that in game theoretical terms, propagation of information constitutes work (at least covering energy losses) performed by intermediary agents, i.e. routers. Thus, under our assumptions, without rewards of any kind, the rational decision for autonomous, independent agents would be not to propagate information at all. On the contrary, if we do decide to reward propagators,

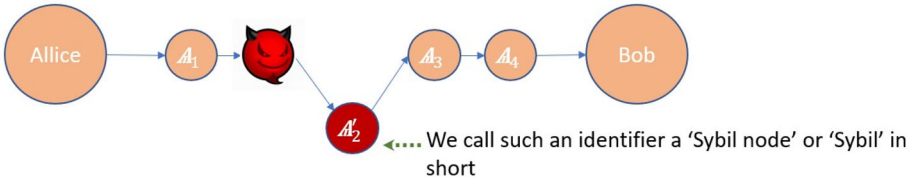



**Fig. 2** Data path with an artificial identity

then from the perspective of intermediaries—the rational decision would be to attempt to trick the system into receiving additional, 'unjust' rewards.

Rewarding intermediaries stands in need for a path assurance apparatus. It needs to enable for non-malleable sequences of nodes' identifiers. We require these embodiments of identities—of those who took part in data-exchange to be delivered to agent(s) responsible for assignment and distribution of rewards. Typically, this means agents operating the consensus layer, thus capable of affecting the top-most State-Domain [1]. Understandably, this leads to a tight coupling between reward-issuance and data-routing/propagation components. Data-propagation may need to account for delicate, game-theoretical assumptions associated with the transmitted information. Once we consider game-theoretical nature of the transmitted information itself—things become even more interesting.

Now, let us wonder—in whose intention would it be for the information to be delivered? Is the information useful for the intermediaries themselves? Does the information have a specific destination? Is the information routable? For an agent issuing a crypto-currency transaction it might be desirable to have it delivered to another agent who *could* affect the top-most State Domain, since it might be unlikely for the former to have the transaction confirmed and included within the decentralized storage all by himself. In systems relying on Proof-of-Work, this might be due to his limited computational capabilities. Such a client would need to incentivize others to distribute the transaction further across the network in hope for it to come about a round-leader who confirms it. Routers want to be paid. Nodes operating the top-most state-domain want to be paid as well. The round-leader *may not* be known in advance. Notice that knowledge of the round-leader's identity/location could be used to optimize transactions' propagation and transform it away from a gossip-like dissemination, towards a more efficient routing protocol. In our work, we abstract away from the notion of 'cryptocurrency transactions' and portrait these as authenticated data-structures describing and *instructing* for a transition of the decentralized state-machine into a new state, thus affecting at least one of its internal variables.

### 1.1.2 Eclipsed networks

A situation might arise when a data-path turns out to be the only one between sender and recipient. We call this an eclipsed communication link or a one-connected network. We require a game-theoretically compatible discouragement of agents, including additional Sybil identities, also in such a scenario. The requirement not met by previous works, which fall short in achieving this for networks of an arbitrary topology.

### 1.1.3 The nature of information

Once agents are assumed as rational, everything is about the *incentives*. What could be portrait as a generalized problem of incentivized data-exchange, in reality might require specialized game-theoretical treatment, depending on the accompanying game-theoretical nature of the exchanged information itself. Addressed or not by the protocol's designers, it would surely be considered and exploited by intermediaries, shall it affect their expected returns. Abstracting away from computer systems and allowing ourselves for a sloppy anecdote,—undeniably transport of *gold* might evoke other passions than transport of fertilizers. Similarly, stepping aside from the problematics of transactions' deliveries let us consider distribution of already mined and confirmed data-blocks. Is there anyone in whose intention would it be to have the transaction delivered? Could the reward and data-propagation rules be structured differently so to better accommodate the actual game-theoretical circumstances? For this, we might consider whether the transmitted information would be of use to anyone. Is it a precious secret? Would possession of the information improve one's situation? Indeed, an entity taking part as a data-router, capable of extending upon the accountable history of events, building upon the current state of the decentralized state-machine, may indeed want to get to know the propagated data-block so to extend upon it. Once it succeeds in 'confirming' the received block (the just acquired information), it may receive a reward (as seen fit by the 'consensus' mechanics i.e. the majority of peers participating in voting-such would be the case in the majority of current 'blockchain' systems).

When deciding whether to retransmit, the agent would first assess its probability of winning. The sources of data-blocks i.e. successful round leaders, may want to have the

data-blocks propagated across the network so as to have their confirmation-rewards acknowledged by others. Here, when designing propagation rewards, we *might not* require data sources to be rewarding intermediaries explicitly. Instead, we *might* make routers receive fractions of block confirmation rewards coming from new blocks—ones extending those they helped to deliver. Notice the ubiquitous accountability requirements. Now, going beyond the internals, towards system extrinsic data propagation—a love letter from Bob to Alice may not need to affect the decentralized consensus, thus for it to arrive at a round-leader is needless. Still, for it to reach Alice, Bob may need to assure its delivery by properly motivating autonomous, independent, profit-driven propagators, and these propagators need to be assured that the reward mechanics are fool proof and fair.

Apparently, a close-cooperation or even integration of reward-assignment and routing algorithms is conspicuous. Data-paths need to be temper-proof and undeniable. A mechanism meeting the just mentioned requirements had been already proposed in [3]—the idea was for each node ($N_i$) on a data-path, starting from source—the current leader, to be adding identifiers of consecutive intermediaries ($N_{i+1}$) to the endorsement (E) of the propagated data-block in an undeniable and temper-proof manner. This was achieved by making identifier of $N_{i+1}$ signed by $N_i$—and since the identifier of the first intermediary was included within the block itself, this rendered the data-block inseparable from the legitimate sequence on intermediaries. Each time this required $N_i$ to get to known the identifier of $N_{i+1}$ first. Any consecutive agent willing to extend upon the received data-block was literally *forced* to work upon the endorsed version of it and thus the reward function executed by round leader could account for all of the intermediaries leading to it. Another approach to path assurance was introduced in [4]. There, each node added itself to an onion-like data-structure, with each layer encrypted during data retransmission with a secret at its centre circumventing separation of the endorsement from the transmitted information. Yet another approach [5] relied upon hashing together consecutive nodes' identifiers, resulting in lower transmission overheads compared to 'chained-identifiers' solutions.

In a system where agents are rationale, introduction of an additional incentive might affect existing ones. The problem of a block-withholding attack has been well studied and known across the literature [6]. If we provide round-leaders with an additional, explicit encouragement for retransmission of newly minted data-blocks—the expected benefits from conducting an attack could dwindle, making others waste fewer resources (time, computational power, opportunities etc.). Now, while data-propagation rewards are welcome by those who propagate, they require someone willing to cover the costs. Were we to reward intermediaries 'out of thin air', then even equipped with a sybil-proof data-transmission apparatus, *we would still be encouraging agents to exchange dummy datagrams*. It may work for *proof-of-work authenticated* data-blocks (as often is the case) as every agent's production capacity is capped by his computational power and thus by the laws of physics, but not for 'transactions' whose production is not computationally bound in a significant manner. On the upside, were we to make sender responsible for rewarding of propagation of these, there would be no retransmission were no rewards assured for intermediaries; consequently leading to dramatically lowered network's susceptibility to spam and/or denial-of-service attacks.

The presumption of agents being driven by profits could be infamously considered as 'selfish' and supposedly as leading to deprivation of open community 'spirits'. Still, the free markets' approach has been widely proclaimed as the actual basis of any healthy economy [7, 8]. Relationships based on mutual benefits have been vastly observed across nature [9–12]. Still, leaving aside far-fetched, implicit reasoning and rationalizations—in the current decentralized state-machines' implementations, there is no tangible incentive for transactions and/or data-blocks to be retransmitted across the network. Let alone for an incentive for current leaders to provide new nodes with older data-blocks so they could re-validate entire history of events leading to the current state—the process which truly adds to the overall security of any open, decentralized state machine. If strength of the system was to rely upon individual decisions, there would be little to no benefit from a myriad of nodes blindly following a few. We as humans currently are used to paying centralized entities for data deliveries—often sacrificing our freedom and privacy.

## 2 Related work

The problem of motivating data-exchange within blockchain-environments is not new and to the best of our knowledge was spotted first in [13] by Babaiaff et al. Traditional defences against Sybil attacks rely on validated identities issued by a trusted authority. Recently an exciting research paper [14] tackled the problematics of Sybil-proofness in a slightly different context (social networks based on Transitive Trust [15] upon which [14] is based—not decentralized state-machines with global convergence to same state across all of the nodes). The paper made it also into the comparative table below, still probably due to the mentioned differences authors did not compare their approach with any of the other major works referenced here-in; it is worth to notice that [14] requires agents to trust that others present honest timestamps: "(..)The only

necessity is that something induces an order on the set of interactions of each agent(..)". Recently, also there was a very interesting work [16] tackling the problematics of Sybil-proofness, however in an entirely different context (CTI systems). Their work showed how a virtual asset *itself* could be used to minimize Sybil-incentives within an external system—such as falsified data-uploads to a CTI system, by introducing costs of doing so.

In previous works, because of reasons which have been discussed, often, the routing mechanism, the reward function and the blockchain architecture itself were tightly coupled to maintain both the game-theoretical and security assumptions. That is also true in our work. For instance, the property of the routing mechanisms assuring identifiers of the neighbouring nodes *and* the investments made by them to be blinded—is required to uphold game-theoretical properties leading to high-finalization guaranties. A simplified high-level comparison, of previous approaches towards blockchain related incentivized data-exchange, in respect to a few of their most important properties, is presented in (Table 1), below:

Please do notice that the novelty in Spide, as seen from the table above is that only it ascertains Sybil-proofness in one-connected and/or eclipsed networks. Also, it is the only one to assure such property when there are no constraints on the network's topology i.e. when the network is modelled as a random d-regular graph.

Looking at the above, Moshe Babaioff et al. were the first to spot Bitcoin's [17] lack of incentives for transactions' propagation and proposed a reward scheme. They proved the effectiveness of it in a rather constrained model in which the peer-to-peer network was assumed to constitute a regular d-ary tree of height H. They did not discuss how to enforce the proposed reward division. Their scheme imposes high reward overheads. If a transaction gives one unit of fee to its successful miner, up to $log(H)$ units of rewards have to be given to propagators [3]. Respectfully, there are three additional works by Oguzhan Ersoy et al. [5, 18, 19], where [5] and [18] provide treatment for transaction propagation, while assuming no topology constraints. To be sybil-proof their proposal requires statistical information to be available throughout the network. Both proposals are valid for a specific blockchain architecture where leader is known first before transaction dissemination begins. In [19] the design is detached from the previously employed leader-first-then-block *modus operandi* and it is proposed to reward propagation of 'transaction sets' through introduction of 'transaction-collectors', effectively proposing a new overall architecture.

# 3 Our contributions

## 3.1 Contribution statement

Herein, we lay forward an innovative and comprehensive sybil-proof data exchange incentivization framework. The framework covers multiple blockchain architectures, comes with integrated routing and path- assurance mechanics and provides a compatible sybil-proof reward assignment function. Overall, the contributed algorithms ascertain sybil-proof, incentivized propagation of information relevant for intrinsic mechanics of an open, decentralized state-machine (blocks and transactions) but uphold the same for propagation of system-extrinsic, arbitrary information as well. The framework is valid for both 'Nakamoto' and = 'leader-known-first' consensus-protocols. Previous works did not assure sybil-proofness in 1-connected or eclipsed networks while, in some cases, providing valid under their respective assumptions impossibility theorem [5, 13]. Here, by assuming reliance on even stronger assumptions and incorporating yet another degree of freedom—an unprecedented in this context concept of reliance on system-intrinsic investments, we propose the first protocol, provably effective in thwarting Sybil nodes in 1-connected or eclipsed networks.

## 3.2 High-level view

Before we head towards depicting details of our contributions, let us give a brief, high-level outlook on how the mechanics are structured. Here, we assume existence of a system intrinsic virtual asset, represented as a variable in the domain of unsigned natural numbers, assigned to every State-Domain [1], with values under a limited control (cannot be arbitrarily incremented) by agents owning the former; with modification rights (ex. 'value transfer' operation) under full provisioning of nodes maintaining consensus at the particular level of the State-Domains' hierarchy tree at which these domains are located. Value-transfers are initiated directly or indirectly (through autonomous code maintained within State-Domains) by agents after providing appropriate authentication to the decentralized consensus. The consensus, as long as it is reached, can cause arbitrary operations on any of the domains' variables (including reward assignment/penalties). Indeed, such a definition is compatible with what is widely known as a 'cryptocurrency'. Agents capable of voting and of over-watching consensus decisions can uprightly be deemed as guardians of a certain kind of a monetary-policy [20].

First off, all agents start of as being equal. There are no external points of trust. We separate the notion of agents

**Table 1** Comparison of previous approaches

| Paper | Transaction propagation (data that ends up in blockchain) | Block propagation (puzzles) | External data propagation when target known (data does not end up in blockchain) | External data propagation target not known (data does not end up in blockchain) | Allows to discourage sybils in 1 connected/ eclipsed networks | Model |
|---|---|---|---|---|---|---|
| Bitcoin and Red Balloons | Sybil-proof only under a very specific network topology model | − | − | − | No. Their model under which they provide proof simply does not cover this scenario. (d-array Tree $d > 1$) | d-array Tree with $d > 1$ |
| Information Propagation on Permissionless Blockchains | + Routing algorithm (with path assurance) + sybil-proof reward function | Proposed to use the same reward function, compatible routing/path assurance algorithm not provided | − | − | No. Their model **does** cover the situation and yet they provide a **valid** impossibility theorem **valid** *under their assumptions* | Random d-regular graphs |
| Transaction Propagation on Permissionless Blockchains: Incentive and Routing Mechanisms (same author as above) | Same as above | Same as above | − | − | Same as above (same author) | Random d-regular graphs |
| Tulip (same author as above) | Same as above; requires 'transaction collectors' elected through Proof-of-Stake | ± (*Very specific* nomenclature of Transaction Sets departure from 'Blocks') | − | − | − | Random d-regular graphs |
| Fully Distributed GRIDNET Protocol, with No Trusted Authorities | − | − | + | + | − | Random d-regular graphs |
| Solidus | Supported compatible path assurance algorithm provided. Reward function is based on a free market approach | Supported-compatible path assurance algorithm provided. Reward function is based on a free market approach | | | | Random d-regular graphs |
| TrustChain | Ensures an upper bound on a Sybil attack. Still, not fully Sybil-proof from a game-theoretical viewpoint | Ensures an upper bound on a Sybil attack. Still, not fully Sybil-proof from a game-theoretical viewpoint | − | − | Not supported | 'TrustChain p2p social graph' |
| Spide | Same as above | Same as above | + (Compatible with previous works) | + (Compatible with previous works) | Supported—Sybil proof in 1-connected and or/ eclipsed networks | Random d-regular graphs |

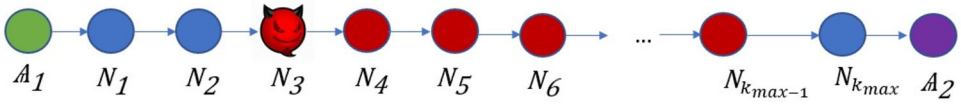



**Fig. 3** We've shown that's entirely possible in [5, 13]

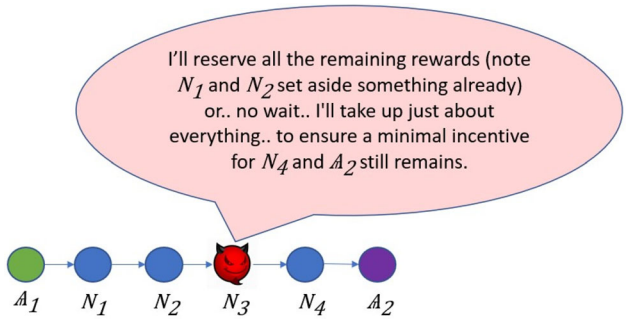
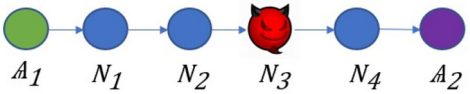



**Fig. 4** That's entirely possible in [3]

from the owned by them identities, allowing for their one-to-many relationship to be depicted within equations. All agents are assumed to be taking part in a game. The aim of the game is to maximize profits. Agents are assumed as rationale and thus driven by profits. The game theoretical *and* technical properties of algorithms to follow were designed to thwart and discourage,—through game rules—when technical mitigation was impossible due to openness of the system,—protocol-compatible yet assumed as unfair actions—such as issuance of unfair rewards.

Second, we make each identity owned by an agent require an initial investment made in system-intrinsic cryptocurrency. Identities of intermediaries are being added into the endorsement of a transmitted information as it traverses the network. Identity on behalf of which a bigger investment is made is expected to gain higher cumulative reward from participation in *random* game-rounds,—compared to an identity with a *lower* associated initial investment value. Similarly, to discourage possession of multiple-identities, investment made on behalf of a single identity should yield higher expected cumulative reward than the same amount of investment spread across multiple identities. Thus, inclusion of an identity into a data-path entails cost, whereas agents hope for profit. Rewards and losses are calculated based on initial investment-values associated with identities encountered within a data path. Calculation and assignment of rewards is carried out by nodes maintaining the consensus. Identities encountered on a data-path, with highest and lowest investment values receive highest and lowest rewards respectfully. In order to maintain high-finalization guarantees, data-routing agents should *not* be able pin-point and participate only in selected game-rounds with identity dealings deemed as favourable. To assure that, we design the game to reassemble a random blinded lottery [21]. Round-leader is rewarded for choosing shortest path to save up on data storage of endorsements within the decentralized data-store. This stays in contrast with other works, where choosing of the shortest data-path was used to guarantee sybil-proofness, yet, such approach and treatment *did not* allow for sybil-proofness in one-connected and/or eclipsed networks, as per impossibility theorem presented in [18]—valid under their limited assumptions.

According to previous works in 1-connected networks:




$N_3$ should keep including Sybil-identities as depicted in figure below, with Sybil-identities denoted in red (Fig. 3).

Where $N_i \in \mathbb{A}_S \forall i > 3 \leq k_{max-1}$; where $\mathbb{A}_S$ is the set of *Sybil* identities.

Now, in previous protocols when assigning rewards based on sequences of encountered identities [5, 13], $N_3$ would continue as long as it believed to be taking part within the shortest data-path, a path whose length does not exceed $k_{max}$, which is a maximum acceptable (by the consensus) data-path length.

Conversely, in protocols [3] where agents have the ability of making on-the-spot (when re-propagating), arbitrary reservations of chosen fractions of the reward-pool before handing information over to neighbours—$N_3$ might say (Fig. 4).

Further notice that, in previous works, $N_3$ would be rewarded *only* if the re-transmitted information reached a full node, as the algorithms were analysed only for propagation of system-intrinsic transactions and/or data blocks. Now, as time was of the essence,- since the shortest path *and only it* was rewarded, $N_3$ would have to use his knowledge of the network's topology before including a sybil-identity hoping to receive additional *'unjust'* rewards with a satisfactory probability. It is worthwhile to note that reserving arbitrary high fractions of the reward pool might had resulted in information not being delivered at all. Our proposal is immune to this type of attacks since it handles 1-connected/eclipsed networks with ease. Here, rewards are assigned solely based on the data-paths' lengths and $\mathbb{N}_i^{\mathbb{D}_I}$ values,—meaning investment-values, made in the name of $\mathbb{N}_i$ encountered within endorsement E ($\forall N_i \in E$). Going back to our previous example, in this work the only way for an agent to increase its expected reward would be to forward more information using identity $N_3$ or to create another identity $N_3'$ whose $\mathbb{D}_I$ value would be greater

$(N_{3'}^{Ъ_I} > N_3^{Ъ_I})$. Thus, it is always best to have higher Ъ as it increases the probability of dominating any encountered identity. As per our to be introduced reward-assignment function (Sect. 6), when having an A priori choice on how to spread investment capital among identities, using two or more identities (ex. $N_3$ and $N_3'$) in the same data-path would yield lower cumulative reward than using a just a single identity $N_3''$ whose $Ъ_I = N_3^{Ъ_I} + N_{3'}^{Ъ_I}$. Thus it is always better to have a single identity within a single data path, one with highest $Ъ_I$ value one can afford over a portfolio of multiple.

The value of $N_i^{Ъ_I}$ lowers with each retransmission (by $\Gamma$) $\forall N$—facilitating kind of a blind lottery-participation fee. It is assumed that $Ъ_I \gg$ reward from participation in any single data retransmission.

After considering the presumable innocence of the problem of incentive-compatible data exchange, it quickly turns out that a concrete protocol design might be far from obvious as some imminent problems arise, namely:

- How to make the protocol universal enough to handle both intrinsic and extrinsic data exchange? How to reward propagation of transactions /and/or bundles of these (i.e. blocks) but also of information extrinsic to the decentralized state-machine's ex. internet traffic?
- How to incentivize agents to invest into a single identity instead of multiple in order to discourage playing multiple hands at the same time (Sybil-nodes)?
- How to disincentivize agents from picking and participating only in favourable dealings (in order to maintain high finalization guaranties)?
- How to design a compatible routing scheme?
- How to carry on with propagation of system-extrinsic information with temporary lack of access to nodes maintaining the consensus?
- *And last but not least,*—how to make the potential solution storage and transmission efficient[1] at high data rates (especially important for system-extrinsic data propagation)

## 4 Problem statement

Let us formulate the problem as: "How to facilitate incentivized, sybil-proof propagation of information both extrinsic and intrinsic to the decentralized state-machine, in the presence of a decentralized data store comprising part of the machine; with the property of non-malleability as time approaches infinity?".

### 4.1 System model/network model/agents and identities

#### 4.1.1 Multi-agent approach

We model problematics assuming a multi-agent approach towards system's design. In particular, each agent $\Lambda$ might own multiple identities $|\mathbb{A}| \geq 0$. We assume all agents being rationale and willing to maximize their expected returns. Thus, we call them 'expectation maximisers' in short. Further, in the context of networking, we might relate to agents simply as 'nodes'. When we emphasize the fact of an identity being used in a specific role/context we shall indicate it in subscript, ex. $\mathbb{A}_C$—with '$C$' for client/producer of transactions (see Definitions).

As previously noted, the system is an open one, thus in order to achieve sybil-proofness, each agent needs to be motivated to transform his identity-portfolio into a singleton (Fig. 5). Further, due to the lack of authorities—the transformation needs to occur autonomously i.e. by each agent's own will. Recall that as long as there is something to gain (i.e. when a maximum data path length was not

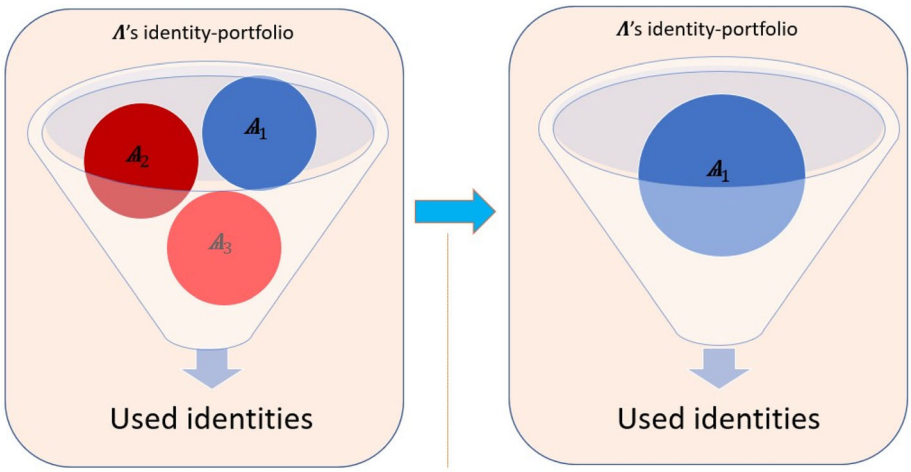



**Fig. 5** Transformation of agent's portfolio of identities into a singleton

reached and assuming immutability of its already reported identifiers)—the incentive to include Sybil-nodes would remain. Because of the aforementioned overall openness, nothing *technically* prevents formation of new identities. Similarly—nothing prevents data creation and/or data-copy operations as these are purely physical processes outside of the system's accountability *operandi*. Thus, without constraints on network's topology, one needs to look for *statistically and game-theoretically sound* game-rules—ones promoting possession of a single identity over a multiple. Effective actions undertaken by agents obeying the protocol need to be accountable and verifiable for. Once we assume,- due to the openness of the system, that it is indeed infeasible to prevent inclusion of Sybil identities into information's endorsement at each clock-tick, we might still be able to design the game in such a way so as to affect the expected returns from cumulative individual actions, effectively discouraging Sybil-identities in the long run. Indeed, under the assumption that nodes are expectation maximisers deprived from the ability of pin-pointing favourable dealings we *may* affect agents' individual decisions during each game-round i.e. at each clock–tick, effectively eliminating Sybil-nodes entirely. Similarly, Bitcoin [17] mitigates Byzantine participants not by solving the Byzantine Consensus problem itself, but rather by providing an efficient probabilistic workaround i.e. practical solution which affects unitary choices of participants. Now, relating to game-theory—we identify multiple simultaneous games: the game between agents trying to become a round leader, the game between clients willing to commit transactions, the game between leaders willing to propagate confirmed data-blocks and the game between agents trying to incentivize propagation of consensus-extrinsic information. By the end of the paper, we shall conclude our communication logic—used to incentivize data-exchange among participants of these games, to be sybil-proof as per the further-introduced *operandi*.

### 4.2 Threat model

In this work, we assume that there are *no trusted entities*. All entities are assumed to be expectation maximisers instead. Parties will strictly execute the designed protocol's specifications, but may try to infer information related to others and to the network's topology—by analysing and intercepting data during communication processes and available within the decentralized state-machine's data-store. Gathered information may then be used to conduct unfair actions, ones *technically* allowed by the protocol specification, yet *not allowed* under game-theoretic rules governed by the majority of agents. It is assumed that due to the openness of the system the only way to prevent inappropriate behaviours is through properly structured intrinsic incentives governed through the conjunction of data-transmission protocols and consensus governing the state of the decentralized state-machine. We assume all agents to be capable of colluding, the fact which should *not* affect game-theoretical guaranties provided by the protocol, as long as, the number of colluding nodes does not exceed the majority of all agents. Data exchange between parties may be intercepted. The numerosity of agents and thus identities is assumed as very high, making inferring relationships between them—unfeasible.

## 5 Definitions

*Blockchain*. We model 'blockchain' as a decentralized probabilistic state-machine where leader of each game-round is responsible for *proposing* and of carrying out transitions to new states in cooperation with the rest of the network. The decentralized state machine's consecutive state representations are buried within a sequence of entangled data-blocks. It is assumed that altering of that sequence(s) of data-blocks/states is prohibitive in the long run [1]. Anyone can propose particular variables of the state machine and/or values of these through 'transactions'. We model leader(s) as unitary independent agent(s), although the results available herein might be applicable to protocols where a consortium(s) constitutes leader(s)—the case which *may* be compatible yet not modelled by degrees of freedom employed within our analysis. Further we may relate to is simply as a state machine.

*Anonymity*. We consider anonymity as routers' inability of getting to know the system-intrinsic identifiers (ex. public-keys and/or identifiers generated based on these) of other data-path members. Such defined *anonymity* allows for, loosely defined, high-finalization guaranties in our apparatus, although *it is not* required to uphold the sybil-proofness guaranties.

*Identity*. $\mathbb{A}$ is created by $\Lambda$ and registered within the decentralized state machine. $\Lambda$ can own multiple identities. $\mathbb{A}$ is assigned an Identity Token describing a Token Pool ($\mathfrak{B}$). Initial parameters of $\mathbb{A}$ and thus $\mathfrak{B}$ are set by $\Lambda$ upon its registration within the state machine. Only $\mathbb{A}$ associated with $\mathfrak{B}_I > 0$ and $\mathfrak{B}$ which is not 'depleted' can effectively participate in a data exchange. It is assumed agents can query any identities' public properties from the state machine (Fig. 6).

- **Dominance**

  We employ the nomenclature of a 'dominance relation'. We say that there is a dominance relation between $\mathbb{A}_i$ and $\mathbb{A}_{i+1}$ and thus $\mathbb{A}_i$ dominates $\mathbb{A}_{i+1}$, if and only if, $\mathbb{A}_i$ has higher *expected* cumulative reward for propagation of information than $\mathbb{A}_{i+1}$.

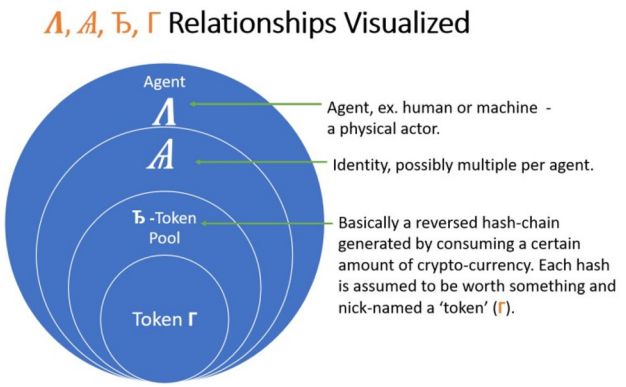



**Fig. 6** Relationship between an agent, identity, token pool and a unitary token

- **Sybil-Proofness**
  We say that a protocol is sybil-proof if it discourages inclusion of artificial identities which do not improve network's connectivity. This definition is compatible with definitions used either explicitly or implicitly in previous works mentioned herein.
- **2-or-more connected networks**
  In 2-or-more connected networks there is over one data-path between source and destination. Previous works achieved sybil-proofness in multiply connected networks but not in 1-connected/eclipsed networks (defined below).
- **1-connected/eclipsed networks**

Conversely, in a 1-connected network there is only one data-path between source and destination. This might be due to low network's connectivity or a result of an eclipsing attack [5].

When depicting communication protocols, for clarity of visual presentation we adopt the following general semantics:

- M—capital latter, when not postfixed by a non-alphanumerical character—the transmitted information—possibly a data structure (ex. M—for a message/payload)
- [A, B, C]—a data-structure here,—a concatenation of information A, B and C, the order is of no significance.
- $F_{A,B..}(x_0 \ldots x_n)$—round brackets—the result of performing transformation F over $x_0 \ldots x_n$, where A and B may indicate additional, function specific parameters. For example: $Sig_{SK}(M)$—results in a cryptographic signature computed over message $\boldsymbol{M}$ and secret SK. The result does *not* include $x_0 \ldots x_n$.
- $F_{A,B..}[x_0 \ldots x_n]$—same as above with the exception that here—the result is a data structure—comprising *both* $x_0 \ldots x_n$ *and* the result of $F_{A,B..}(x_0 \ldots x_n)$. For example: $Sig_{SK}[M] = [\mathrm{M}, Sig_{SK}(\mathrm{M})]$. The order of concatenations is of no significance yet assumed as constant throughout agents executing the protocol.

## 5.1 Association symbol ~

We define the symbol of association '~' to depict a functional relationship between two data fields or structures. For instance, $\mathbb{A}_1^{ESK} \sim M_i PE_k$ describes that $\mathbb{A}_1^{ESK}$ is an ephemeral secret key issued by $\mathbb{A}_1^{ESK}$, having a functional applicability to data structure $M_i PE_k$ such as the ability of decrypting information contained within it.

## 5.2 Data transformations

Within algorithms' definitions we will be often transforming data-structures by including additional fields ex. M = [M, $N_0^{PK}$]. For clarity of presentation, we may treat data fields and/or structures as sets ex.$N_i^{AD} = \varnothing$ depicting that $N_i^{AD}$ is empty.

*Token pool* (Ъ) is a data structure representing a store of value, with a unitary, derived token (Γ) as the smallest spendable unit of its total value. The pool is stored within decentralized datastore. A one-to-one relationship between the owning identity $\mathbb{A}_i^S$ is assumed. The structure is signed with $\mathbb{A}_i^{SK}$. The value of a single Γ is computed as $\frac{Ъ_S}{Ъ_C}$. Let $S_H$ be a secret 256-bit value. Hash-values representatives of each token are computed in an ascending manner over $\boldsymbol{i}$ through:

$$\Gamma_1 = Hash(S_H)\ \Gamma_i = Hash(\Gamma_{i-1})\ ; \forall n > 1 \leq Ъ_C.$$

The value of $\Gamma_{Ъ_C}$ is part of Ъ and public, whereas only a limited portion of bits of $S_H$ may be made public and stored within Ъ. The amount of revealed bits should not make brute-forcing $S_H$ feasible. The total number of hashes comprising the Token Pool is stored publicly. In order to make payments or to prove its identity Λ steadily releases $\Gamma_i$ values in an descending order on $i$. Here, the security relies upon the assumed infeasibility of computing $\Gamma_i$ based on $\Gamma_{i+1}$ i.e. security of a one-way hash function. Note that the spender of assets may release any number of consecutive hashes and do so efficiently by transmitting only the last to be revealed, consequently revealing all the previous to it. If done so, the resulting value of a transmission token accumulates. Double spends are detected through keeping track of the used-up depth within the state-machine. If, a token at depth $i$ is made public it is assumed tokens of $i' > i$ have also been spent already as well. Consequently, when $\Gamma_1$ is made public, the Ъ is assumed to be used up in its entirety.

- *Possibly compromised*—in threat models to follow we say that the reward process is possibly compromised if data allowing to include Sybil-identities had been disclosed. Thus, possibly under additional requirements and/or circumstances—making compromising of the integrity of legitimate endorsement E feasible. Conversely, if data securing the reward process could be considered as immutable (ex. data buried at a blockchain depth assumed as irreversible), then and only then,—the reward process *would not* be considered as *possibly compromised*. In our work, if nodes obey the protocol, the possibly compromised scenario does never arise.
- *Finalization guaranties*—we assume that the inability of agents to choose between favourable and non-favourable routing decisions increases the likelihood of the routed information M to arrive at its destination. That is under the assumption that agents are willing to maximize their profits and thus are ready to take a minimal amount of risk. Conversely, the ability to discriminate on less favourable routing-requests obviously results in lower probability of data delivery to succeed, thus—in lower finalization guaranties (Table 2).

Definitions used further throughout the paper are listed in table below:

The rationale behind $\mathfrak{Tb}_{\mathrm{I}}$ is the same for all algorithms and represents an investment each intermediary needs to make to participate in the routing process. That is to prevent the nothing-at-stake problem. Recall that token-pool's tokens represent fractions of its entire value and thus are considered spent once revealed and once the corresponding token-pool's usage metrics are updated—all this needs to be performed at the discretion of decentralized consensus.

Value marked with a colon (ex. $\mathrm{X}^{'}$) represents the transformed, oftentimes *received* information $\mathrm{X}^{'}$ that logically corresponds to X but it *may* happen that $\mathrm{X}^{'} \neq X$ after a possibly malicious, extrinsic modification done to X.

*Path accountability* We say that a path assurance algorithm thwarts tempering with the content of payload's endorsement if payload delivered alongside the modified endorsement cannot affect the state of the decentralized state-machine.

## 6 Reward function

Before proceeding with introduction of the main body of the reward function, we shall first introduce some of its coefficients and describe two variations of the identity rating function across two dominance-relations.

Let the Identity Rating Function $\epsilon$ be defined as $\epsilon\left(\mathbb{A}_{\mathrm{i}}^{\mathfrak{Tb}_{\mathrm{I}}}, L, \beta\right) = L^{\beta \mathbb{A}_{\mathrm{i}}^{\mathfrak{Tb}_{\mathrm{I}}}}$, where $\beta \geq 1, \mathfrak{Tb}_{\mathrm{I}} \geq 1, \mathfrak{Tb}_{\mathrm{I}}, \mathbb{A}_{\mathrm{i}}^{\mathfrak{Tb}_{\mathrm{I}}} \in \mathbb{N}, \beta \in \mathbb{R}$. $\beta$ is decided upon by leader of each round. i.e. must be a result of decentralized voting. The function assures exponential growth from investment and leads to exponential growth in probability of dominating any set of encountered identities. It also has the property of $\epsilon(\left(\mathbb{A}_i^{\mathfrak{Tb}}\right))$ being always greater than $\sum_1^L \mathbb{A}_i^{\mathfrak{Tb}-1}$ for any data-path length.

Let us now bring into picture the Reward Function $F_{\mathbb{A}i}$ defined over two *dominance relations:*

- **Global Dominance**

$$\forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X,\ F_{\mathbb{A}_i}(\mathbb{A}_i^{\mathfrak{Tb}_I}) = \frac{\epsilon\left(\mathbb{A}_i^{\mathfrak{Tb}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathbb{A}_i^{\mathfrak{Tb}_I}\right)}$$

- **Local Dominance**

$$F_A(\mathbb{A}_i^{\mathfrak{Tb}_I}) = \frac{\beta * (L - \nabla_i * F)}{\sum_{i=1}^{L} \nabla_i}$$

In local dominance, index $\nabla_i$ is used instead of $\epsilon\left(\mathbb{A}_i^{\mathfrak{Tb}_{\mathrm{I}}}\right)$ when compared to *Global Dominance*. In the former, a mapping $\varnothing$ is employed to map values of $\mathbb{A}^{\mathfrak{Tb}_i I}$ onto a family of ascendingly ordered elements $\nabla$, with possibly multiple elements $\mathbb{A}_i$ $\forall i \leq k$ at each hierarchy level $\nabla_i$ $\forall i \leq k$; i.e. $\nabla_0 = max(\mathbb{A}^{\mathfrak{Tb}_i I})$. Identity rating function takes the hierarchy's index $\nabla_i$ as input. In Global Dominance, stakes are used directly as exponents. It better captures reward-rates when differences between consecutive stakes are low. Still leading to overflows when large values are used. When stakes are high, the discrepancies between rewards might be very large. Conversely, in Local Dominance stakes are not used *directly* as exponents. Instead, identities are sorted by stakes in a descending order (duplicates removed) with indexes fed into the rating function. Now, due to small exponents, the result is not susceptible to overflows. The spread of rewards is less 'bumpy', with more rewards assigned to low-bidders when compared to global dominance, especially when differences between encountered stakes are high.

### 6.1 Routing mechanism

With routing of information there is always the sender, to be transmitted information M and a single or multiple recipients. Because we will take use of a decentralized state machine and stemming from the assumption that routing of

**Table 2** Table with definitions

| | |
|---|---|
| $\Lambda$—an agent. For clarity, in equations used as a superset of all its identities | $(\mathbb{A}_i)$—identity ranking function in the domain of identities |
| $\mathbb{A}_i$—an i'th identity (with N as an alias for a 'data router') | $F_A(\mathbb{A}_i)$—reward assignment function in the domain of identities |
| $\mathbb{A}_i^{SK}$—i'th identity's secret key | L—data path length $L \in N > 0$ |
| $\mathbb{A}_i^{PK}$—i'th identity's public key | $\mathbb{T}$—Token Pool |
| $\mathbb{A}_M$—the set of legitimate identities. Takes only identities whose $\mathbb{T}_I > 0$ | $\mathbb{T}_I$—One-time investment made into a $\mathbb{T}$. $\mathbb{T}_S \in N > 0$ |
| $\mathbb{A}_S$—the set of Sybil identities. Takes only identities whose $\mathbb{T}_I > 0$ | $\mathbb{A}_i^{\mathbb{T}I}$ associated with i'th identity in a data path |
| $N_{E_i}$—the set of identities in the role of intermediaries in Epoch i | $\mathbb{T}_C$—declared number of usable 'tokens' from within a $\mathbb{T}$ |
| $PID_i$—identifier of an i'th data path | $\Gamma$—Token from a Token Pool |
| ${}^S\mathbb{A}_X$—the superset of ${}^S\mathbb{A}_M$ and ${}^S\mathbb{A}_S$ i.e. ${}^S\mathbb{A}_X = {}^S\mathbb{A}_M \cup {}^S\mathbb{A}_S$ | $\Gamma_V$—value of a single token, calculated as $\frac{\mathbb{T}_I}{\mathbb{T}_C}$ |
| $\Phi$—the set of strategies available to agents | $\mathbb{T}_\Gamma$—value of all tokens ($\Gamma$) within an endorsement |
| $\Phi_i$—an i'th strategy | $\mathbb{T}_U$—index of the last used $\Gamma$ from $\mathbb{T}$. When $\mathbb{T}_C = \mathbb{T}$, $\mathbb{T}$ is assumed to be depleted |
| $\mathbb{A}_C$—an identity in the role of a client i.e. producer of transactions | $\mathcal{L}_i$—specialization of $\Lambda$—the leader of i'th round i.e. the agent who proposed for a transition of the state machine into the i'th state |
| $B_i$—i'th blockchain block. The lower subscript, the earlier the block in time | E—an endorsement proving data path accountability; usually alongside M. In equations, treated as a set of all layers $E_i$ added by consecutive intermediaries $0 > i \leq k$ |
| $E_i^L$—the i'th 'layer' of an endorsement | $E_S$—a secret within an endorsement set by a producer of M |
| $\mathbb{A}_i^{AD}$—additional data related to i'th identity | ${}^C\mathbb{A}_T^S$—stake value returned to $\mathbb{A}_C$ on success |
| M—routable message (a payload) | $\epsilon$—identity rating exponential function |
| T—transaction (specialization of M) | $B_S$—blockchain depth assumed as stable |
| ${}^C\mathbb{A}_T^S$—identitie's stake associated with M (usually T or B). (provides an incentive for $\mathbb{A}$ to act in $epoch_2$) | $B_X$—block at any depth |
| k—data path length, without sybil nodes | $P_K^M$—a data path of length K that M has traversed |
| $f_i^k$—reward received by $\mathbb{A}$ at position $i$ in a data path of length k | $R_L$—value of reward pool for leader |
| F—cumulative reward available in a data path of length k i.e. $F = \sum_1^k f_i$. Declared by $\mathbb{A}_C$ | $R_N$—value of reward pool for routers |
| $F_{\mathbb{A}_i}^k$—total reward received by $\mathbb{A}_i$ in a data path of length k | TE- transaction's endorsement's data structure containing a data bundle together with an endorsement |
| P—authentication data prepared by sender that is to be stored within the decentralized storage alongside E | UD—'unlocking data' data-bundle. Sent by client in order to activate Epoch2 |
| MPE—[M,P,E] | PE—[P,E] |

information and methods of encryption heavily depend on whether *public key* of the leader is known—further we shall be considering two scenarios:

- "Block-First-Then-Leader" (BFTL)—where leader is unknown at the beginning of each transactional round, thus both transactions and blocks are disseminated though a gossip like protocol.
- "Leader-First-Then-Block" (LFTB)—where leader is known at the beginning of each round, thus, supposedly, transactions can be propagated more efficiently.

Here, let us note that statistical finalization guaranties differ and depend upon game theoretical characteristics of the transmitted information itself (transactions vs data-blocks vs state-machine extrinsic datagrams). They also depend upon the architecture—whether we are dealing with a LFTB or BFTL blockchain as the game-theoretical properties of the employed routing mechanism differ between the two architectures and from the viewpoint of a routing node—portrait divergent incentives for participation. Still, independently from the routing and path assurance methodology; the reward distribution function is

always the same and will be proven to be sybil-proof in Sect. 8. Recall that the reward distribution function requires certain guaranties from the underlying data-propagation sub-system in order for its sybil-proof reward distribution guaranties to hold. These include non-malleability of the traversed data path and the previously defined anonymity among protocol-compatible routers.

Let us now proceed by introducing two path assurance algorithms: $PA_1$ and $PA_2$. These will be introduced steadily due to their rather intricate nature. The algorithms do account for the transmitted payload $M$. In other words, they facilitate both path-assurance and data propagation apparatus. The algorithms *do not* sensu stricto prove the data-path traversed, as it would indeed be unfeasible to assess everyone physically involved in propagation of information especially of parties not interacting with the data/protocol itself—a situation which *could* arise if nodes were not interested in being rewarded. Still, such a situation is incompatible with our assumptions and thus outside of the scope of our analysis. We shall require algorithms to ensure non-malleability of the already traversed data path against future intermediaries. Note the sound orchestration between the data-path assurance algorithms, the sybil-proof reward function and the propagation of the payload itself.

From the game-theoretical perspective, we distinguish between three types of transmitted information:

o Information which is supposed to be assessed by the decentralized consensus and thus affect its state.
p Information whose image (ex. $SHA_{256}$(M)) is supposed to be assessed by the decentralized consensus.
q Information that needs to arrive at its destination but which is otherwise consensus agnostic.

The first two imply the necessity of either M or its image, respectfully, to be included into the decentralized data-store so that the resulting state-transitions can be verified by the current and future leaders. In order to account for imminent game-theoretical differences, the proposed path assurance algorithms support two, related in this context, propagation modes: open and sealed. In latter, there is an additional initial step in which $\Lambda_1$ queries $N_0$ for its identifier and performs transformation $\Phi(M, N_0^{PK})$ over M which irreversibly imposes $N_0's$ identifier onto M itself; effectively rendering M *game-theoretically* inseparable for applications involving the decentralized consensus under the premise that function $\Phi'$ verifying the bond's $(M \leftrightarrow E)$ integrity needs to succeed on each node making up the majority. To allow for equations and to achieve extendibility without data loss, M is treated as a data-structure or a sequence of values. For better visualization let us now provide some sample *sealing transformations* ($\Phi$) below:

- $M' = \Phi(M,..) = Sig_{\Lambda_1^{SK}}[M, ID = N_0^{PK}]$—the operation which makes M inseparable from $ID = N_0^{PK}$ under the premise that M needs to be accompanied by ID and signed together in the name of $\Lambda_1$. Naturally, the strength of this bonding depends on the strength of the signature.
- Similarly, assume $M' = \Phi(M,..) = [M, ID = N_0^{PK}, PoW([M, ID], \sigma)]$. Now, under the assumption that $\Phi'$ verifies presence of both ID and of the result of Proof-of-Work [22] (PoW) meeting certain difficulty $\sigma \geq \sigma_{min}$ done over [M, ID], then M is inseparable from its coefficients ID and PoW under the difficulty $\sigma$ used to compute the Proof-of-Work.

Now, the sole purpose of $\Phi$ is rendering M otherwise worthless for conceives involving the decentralized consensus when detached from E or shall E be replaced with $E'$. One such usage scenario would be prevention of M from being used by leaders when delivered through a noncompatible path assurance/route traversal protocol or when a legitimate path was mangled with.

The path traversal and path verification algorithm need to protect the integrity of previously traversed identities at all times in order to uphold sybil-proof guaranties of the data-traversal algorithm. Recall that to achieve sybil-proofness in one-connected and/or eclipsed networks, the reward assignment function needs to promote possession of a single identity over multiple so to discourage inclusion of sybil identities *even if* such an action would be compatible with the path traversal protocol itself i.e. *technically possible*. The problem of assuring that M cannot not be replaced from within the MPE and thus of assuring that the endorsement E cannot be used for retransmission of another $M'$ facilitates a separate dilemma we will need to tackle.

Recall that in order to achieve high finalization guaranties we strive to ensure anonymity between intermediaries during traversal of M under the assumption that agents are expectation maximisers and would omit dealings that are likely not to generate profit, were investment values of their neighbours be known.

The path assurance and data propagation algorithms together with the accompanying threat models and game-theoretical assumptions under which they are valid will be presented below. Note that for now we give generalized protocols without implying M to be a transaction or a data block. We will give case-specific treatments later on when extending upon what is being shown. While the purpose of the following algorithms is detecting if path within the endorsement E was tempered with, the very intentions of including Sybil nodes shall be thwarted by the sybil-proof reward function introduced later on in Sect. 8.

Shall $\Gamma_{N_i}$ be revealed but not make it into the decentralized storage and thus token pool's usage not be updated within it, the $\Gamma_{N_i}$ value might be consider as compromised and may be reused.

$PA_1$ supports two routing modes: open and anonymous, together with the already discussed information-sealing features.

#### 6.1.1 Path assurance algorithm $PA_1$

**Assumptions** Initially, $\Lambda_1$ is not required to be aware of any intermediaries besides the initial intermediary $N_0$. Now, we require $\Lambda_2$ to be able to get to know the identities of agents who helped with propagation of M. Let $\Lambda_1$ be the sender of information and let $\Lambda_2$ be the recipient. Assume $\mathbb{A}_1 \in \Lambda_1$ and $\mathbb{A}_2 \in \Lambda_2$. Now dependently on routing mode:

- Anonymous mode: $\Lambda_2$ is required to be known to $\Lambda_1$ by $PK^{\mathbb{A}_2} \in \mathbb{A}_2$
- Open mode: $\Lambda_2$ is assumed not be known to $\Lambda_1$

The above two modes are self-excluding.

Now, agnostically to routing mode, there are two optional features which can be enabled:

- Sealing-feature implying use of Φ(M, E)—used by sender and Φ′(M, E)—by verifier
- Public-Target known-feature—yet to be described.

Now, let us define transformation $\Phi_1$ rendering M inseparable from E, together with a function $\Phi'_1(\mathrm{M}')$ verifying bond's $(M \leftrightarrow E)$ integrity:

The purpose of a keyed-mac function and thus of R is to thwart intermediaries ability of getting to know $N_0^{PK}$ under the premise that all identity-related public keys are traversable within public storage. The value of HM does not leak any information regarding $N_0^{PK}$ under presumption of HMAC being a one way transformation. Thus, not allowing for associating with any of the registered public keys, as long as, R remains secret during path traversal. Here, bond's verification requires two round trips. During the second, the value of R is released, as required by $\Phi'_1$. Indeed, full validation cannot be performed on sight of $MPE_k$ alone. Here, note that M is not required by $\Phi'_1$. and its image is sufficient instead.

Now, let us define yet another bonding $(M \leftrightarrow E)$ transformation $\Phi_1$:

Here, even under the premise that public keys of identities are known, transformation $\Phi_2$ does not leak information regarding $\mathrm{N}_0^{\mathrm{PK}}$ thus not allowing for association of transmitted information M with identity-related $\mathrm{N}_0^{\mathrm{PK}}$. That is because HM is based solely on an ephemeral public key used by intermediaries and sender to encrypt endorsement's layers on the way to destination. Interestingly, both transformations irreversibly associate any future legitimate version of endorsement $E_k$ with M while only knowledge of a few initial, imminent variables is required. That is under the premise that path assurance algorithm assures non-malleability of $E_{i-1}$ against $N_i$. Note that verification $\Phi'_2$. can be performed on sight of $MPE_k$, requiring only $\mathbb{A}_2^{\mathrm{EPK}}$

$$\Phi_2\left(\mathrm{M}, \mathbb{A}_1^{\mathrm{SK}}, \mathbb{A}_2^{\mathrm{EPK}}\right) = [\mathrm{M}, \mathrm{sig} = Sig_{\mathbb{A}_1^{SK}}(\mathrm{HM})], \text{ where } \mathrm{HM} = \mathrm{MAC}\left(\mathrm{SHA}_{256}(M), \mathbb{A}_2^{\mathrm{EPK}}]\right)$$

$$\Phi'_2\left(\mathrm{SHA}_{256}(\mathrm{M}'), \mathbb{A}_1^{PK}, \mathbb{A}_2^{\mathrm{EPK}\prime}\right) = VerSig_{\mathbb{A}_1^{PK}}(sig \in \mathrm{M}, \mathrm{HM}') \text{ where } \mathrm{HM}' = \mathrm{MAC}\left(\mathrm{SHA}_{256}(M'), \mathbb{A}_2^{\mathrm{EPK}\prime} \in E'_0 \in E\right)$$

Let N be an ordered set of identities of agents participating in the routing process, thus |N|= k; where $\mathrm{N_i} \in \mathrm{N}$; $\mathrm{i} \geq 1$ represents a particular routing identity. It is assumed that $\mathbb{A}_2^{\mathrm{PK}} \in \Lambda_2$ is known to $\Lambda_1$.

*Threat model* ($TM_1$): Attackers are believed to be trying to replace one or more identities within the endorsement. Potential attackers include nodes encountered during traversal of M and the final destination $\Lambda_2$. The ability of

$$\Phi_1\left(\mathrm{M}, \mathbb{A}_1^{\mathrm{SK}}, \mathrm{N}_0^{\mathrm{PK}}, R\right) = [\mathrm{M}, \mathrm{sig} = Sig_{\mathbb{A}_1^{SK}}(\mathrm{HM})], \text{ where } \mathrm{HM} = \mathrm{HMAC}([\mathrm{N}_0^{\mathrm{PK}}, \mathrm{SHA}_{256}(\mathrm{M})], \mathrm{R}], \mathrm{R})$$

$$\Phi'_1\left(\mathrm{SHA}_{256}(\mathrm{M}'), \mathbb{A}_1^{PK}, sig \in \mathrm{M}, \mathrm{N}_0^{\mathrm{PK}\prime}, R\right) = VerSig_{\mathbb{A}_1^{PK}}(sig \in \mathrm{M}, \mathrm{HM}') \text{ where } \mathrm{HM}' = \mathrm{HMAC}\left([\mathrm{N}_0^{\mathrm{PK}\prime}, \mathrm{SHA}_{256}(\mathrm{M})], \mathrm{R}' \in E'_0 \in E\right)$$

the protocol to detect and thwart tempering with the content of E relies upon the strength of the applied signature function, together with the secrecy of $\mathbb{A}_1^{SK}$. Now dependently on enabled modes and features:

- **Anonymous mode**: we assume intermediaries to be trying to get to know identities of their peers in order to assess their likelihood of either winning or losing. This should be unfeasible under the assumed strength of the encryption function and secrecy of recipient's private key.
- **Open mode**: intermediaries are free to get to know identities of their peers by looking at endorsement E.
- **Seal-feature**: we require M to be usable *only* when accompanied by a legitimate E under the assumption that $\Phi(M, E)$ transforms M in a way that $\Phi'(\mathrm{M}, \mathrm{E}')$ succeeds *only if* $\mathrm{E}' = \mathrm{E}$. Intermediaries and $\Lambda_2$ are assumed to be willing to detach E from M in order to introduce Sybil nodes of their own—all of which should be mitigated and fall short under the requirement that $\Phi'(\mathrm{M}')$ needs to succeed at $\Lambda_2$.

We present the algorithm in (Fig. 7) below:

Note that the bond between M and E established through an image $SHA_{256}(\mathrm{M}) \in \mathrm{P}$(1st line) is insufficient when requiring M to be unusable without the presence of E, shall M arrive at leader by omitting to the described propagation scheme. That is when the bonding mode comes in handy. Further, we will take use of optional parameters in more complex use-case scenarios. Note that, although $PA_1$ requires round-trips between consecutive agents, no further interaction from $\Lambda_1$ is needed upon the release of $\mathrm{MPE}_0$. Also, as far as incentives go, the above algorithm does not provide any. In order to make the algorithm incentive-compatible we will equip it with the capability of rewarding intermediaries with a 'virtual asset' i.e. cryptocurrency. Yet, for this to be possible, the meta-data based upon which rewards are to be issued need to make it into the decentralized storage so to be assessed by the decentralized consensus. Specifically, the payload's accompanying endorsement has to—so that round leaders, operating the decentralized consensus mechanics can properly recognize the intermediaries and carry out reward assignments, on the decentralized state machine, as dictated by the sybil-proof reward function (Sect. 8). Therefore, in anonymous mode, an ephemeral key is employed to allow others to decrypt E without compromising the recipient's identity-related private key. Notice that for E to be made available within the decentralized data store, $\Lambda_2$ first needs to be incentivized to include E into a data-block. If $\Lambda_2$ is not a leader, we assume that he needs to be incentivized to motivate others (i.e. another leader which might or *might not* be known to $\Lambda_2$ *and* possibly additional intermediaries leading to it)—to help include E into a data-block. All this should happen at the expense of $\mathrm{A}_1$—in whose intention it is, as we assume, to deliver M to $\mathrm{A}_2$. Above all, the entire process needs to remain incentive-compatible.

Previous algorithm did not allow intermediaries to remain anonymous when recipient's public key remained unknown. Now, we will propose a data-routing and path assurance algorithm suitable for when $\mathbb{A}_2^{PK}$ is unknown to $\Lambda_1$ and for when anonymity among intermediaries is desirable. Here, the only sought for property of a potential recipient is for it be a potential leader. The mechanism requires two round trips between sender and—one or two possibly different leaders. We shall proceed by introducing a simplified non-incentive compatible version of the protocol.

### 6.1.2 Simplified Path Assurance Algorithm $\mathrm{PA}_2$ (two modes: open and sealed)

***Assumptions*** Let $\Lambda_1$be the sender of information and let $\Lambda_2$ be the recipient. Assume $\mathbb{A}_1 \in \Lambda_1$ and $\mathbb{A}_2 \in \Lambda_2$. Let N be an ordered set of identities of agents participating in the routing process, thus $|\mathrm{N}| = \mathrm{k}$; $\mathrm{N_i} \in \mathrm{N}$ where $\mathrm{i} \geq 1$ represents a particular routing identity. We assume all agents to be well connected to the network. The aim of the algorithm is to enable for path assurance and data delivery from $\Lambda_1$ to $\Lambda_2$ when $\mathbb{A}^{\mathrm{PK}}{}_2$ is unknown *under the premise that* $\Lambda_2$ *is a round leader*. Initially, $\Lambda_1$ is not obliged to know any intermediaries besides $\mathrm{N}_0$. We want transmitted information M to arrive at the ultimate destination as soon as possible and require the protocol to be effective under simplex connectivity between the sender and round leader who is assumed to be selected randomly with uniform distribution and unknown. That stems from an assumption that there might be no time to allow for additional round-trips between the two so to perform a Diffie–Hellman key-exchange and establish a secure channel as leaders might rapidly change.

*Threat model* ($TM_2$): Potential attackers are nodes encountered during traversal of M including the eventual destination $\Lambda_2$. Attackers are assumed to be trying to replace one or more $N_i^{PK} \in E$. The ability of the algorithm to detect and thwart tempering with E relies upon the presumed strength of encryption and secrecy of both $E_S$ and $\mathbb{A}_1^{ESK}$. That is until E becomes non-malleable and the two values are made public as part of the *Unlocking Data (UD)*. Information within *UD* allows to validate, decrypt and possibly rewrite E. In order to ensure non-malleability of E after M reaches its destination and thus to uphold sybil-proof properties of reward-mechanics-it is of an utmost important not to release UD too early, meaning not

**$Epoch_1$:**

**Required Preconditions:** $\Lambda_1$ prepares identity $\mathbb{A}_1(\Lambda_1 \ni \mathbb{A}_1)$ together with key-pair $\mathbb{A}_1^{\mathrm{SK}}/\mathbb{A}_1^{\mathrm{PK}}$, along with a payload M. Intermediaries prepare identities $\mathrm{N}_i$ together with public-keys $\mathrm{N}_\mathrm{i}^{\mathrm{PK}}$, $\mathrm{N}_\mathrm{i}^{\mathrm{AD}}$ $\forall \mathrm{N}_i; i \geq 0 \leq k$ ; $\Lambda_2$ prepares key-pair $\mathbb{A}_2^{\mathrm{SK}}/\mathbb{A}_2^{\mathrm{PK}}$.
**Optional:** $\Lambda_1^{AD}$, $\mathrm{N}_\mathrm{i}^{\mathrm{AD}}$– *Additional Data (AD)* included by sender/intermediary. (${\Lambda_1}^{\mathrm{AD}} = \mathrm{N}_\mathrm{i}^{\mathrm{AD}} = \emptyset$ by default); $\mathbb{A}_2^{ESK}/\mathbb{A}_2^{EPK}$ – ephemeral key-pair used in anonymous routing mode.

1. $\Lambda_1$ queries $\mathrm{N}_0$ for its $\mathrm{N}_0^{\mathrm{PK}}$ , thus

   $\mathrm{P} = Sig_{\mathbb{A}_1^{SK}}[\mathrm{SHA}_{256}(\mathrm{M}),\mathrm{SHA}_{256}(\mathrm{E}_0), \mathbb{A}_1^{\mathrm{PK}}, \Lambda_1^{AD}]$; *now dependently:*

   - ***Seal-feature:*** $\Lambda_1$ does $\mathrm{M}= \Phi_2(\mathrm{M}, \mathrm{N}_0^{\mathrm{PK}}, \mathbb{A}_2^{\mathrm{EPK}})$, $\mathrm{E}_0 = Enc_{\mathbb{A}_2^{\mathrm{EPK}}}(\mathrm{N}_0^{\mathrm{PK}}, \mathrm{R})$
   - ***Target-known feature****:* $\mathbb{A}_2^{\mathrm{EPK}} \in \Lambda_1^{AD}$*, defined by* $\mathbb{A}_1$
   - ***Open Mode:*** $\mathrm{E}_0=\mathrm{N}_0^{\mathrm{PK}}$, ***or:***
   - ***Anonymous Mode :*** $\mathrm{E}_0=Enc_{\mathbb{A}_2^{\mathrm{EPK}}}(\mathrm{N}_0^{\mathrm{PK}})$,

   2. $\Lambda_1$transmits $\mathrm{MPE}_0 = [M, P, \mathrm{E}_0]$ to $\mathrm{N}_0$
3. Each consecutive $\mathrm{N}_\mathrm{i}$ where $i \geq 0$ queries its neighbour for $N_{i+1}^{PK}$ before preparing

   - ***Open Mode:*** $\mathrm{E}_\mathrm{i} = Sig_{\mathbb{A}_1^{SK}}[\mathrm{E}_{\mathrm{i}-1}, \mathrm{N}_\mathrm{i}^{\mathrm{AD}}, \mathrm{N}_{\mathrm{i}+1}^{\mathrm{PK}}]$
   - ***Anonymous Mode****: :* $\mathrm{E}_i = Sig_{\mathbb{A}_1^{SK}}[Enc_{\mathbb{A}_2^{\mathrm{PK}}}(\mathrm{E}_{\mathrm{i}-1}, \mathrm{N}_\mathrm{i}^{\mathrm{AD}}, \mathrm{N}_{\mathrm{i}+1}^{\mathrm{PK}}])$

   Then further transmitting $\mathrm{MPE}_\mathrm{i}$ =[M,P, $\mathrm{E}_\mathrm{i}$] to $\mathrm{N}_{\mathrm{i}+1}$.
4. Once $\mathrm{MPE}_\mathrm{k}$ arrives at a final destination $\Lambda_2$, here the processing also depends on routing mode and enabled features:

   - **Anonymous Mode:** $\Lambda_2$ it extracts $\mathbb{A}_1^{\mathrm{PK}}$ and does $Dec_{\mathbb{A}_2^{ESK}}(\mathrm{E}_\mathrm{i}) \forall 0 \geq \mathrm{i} < \mathrm{k} \in \mathrm{E}$
   - **Sealed Feature:** performs $\Phi'_2(\mathrm{M}', \mathbb{A}_1^{PK\prime}, \mathbb{A}_2^{\mathrm{EPK}\prime})$on failure the protocol aborts.
   - ***Target-known feature:*** $\Lambda_2$ verifies to be the looked for target by verifying if
     $$\mathbb{A}_2^{\mathrm{PK}} \in \Lambda_1^{AD}$$
   - Open mode: processing proceeds.
5. Now, independently of mode used $\Lambda_2$ assumes $\mathbb{A}_\mathrm{X}^{\mathrm{PK}} = \mathbb{A}_1^{\mathrm{PK}}$ for $\mathrm{E}_0$ or $\mathbb{A}_\mathrm{X}^{\mathrm{PK}} = \mathrm{N}_\mathrm{i}^{\mathrm{PK}}$ $\forall \mathrm{E}_\mathrm{i} \forall 0 \geq \mathrm{i} < \mathrm{k}$ and then does $VerSig_{\mathbb{A}_\mathrm{X}^{\mathrm{PK}}}(\mathrm{E}_\mathrm{i}) \forall 0 \geq \mathrm{i} \leq \mathrm{k}$ - effectively verifying entire sequence of identities $N_i^{PK} \forall i < k \in \mathrm{E}$. Note that in the open mode, that is also possible for each intermediary. If positive, the algorithm concludes with a properly verified list of intermediaries.

**Fig. 7** High-level representation of path assurance algorithm $PA_1$

until E becomes non-malleable. The property of non-malleability is assumed to be acquired by E when stored beneath blockchain-depth $\geq B_S$ which is when $\Lambda_1$ can safely release the *UD*. Similarly, here, the algorithm also supports two propagation modes: *open* and *sealed*. In latter, transformation $\Phi$ irreversibly embarks identifier of the first intermediary $\mathrm{N}_0$ onto the transmitted information/data-structure M without revealing information regarding $\mathrm{N}_0$ during $\mathrm{Epoch}_\mathrm{i}$ to anyone.

The algorithm takes use of 'clearing transmissions' nomenclature—the concept which will be explained further.

Do note that the algorithm below is concerned with three epochs in time. The current ***Epoch$_i$***—in which sender is willing to dispatch data; ***Epoch$_{i-m}$*** which happened in the past—*m* epochs ago; and ***Epoch$_{i+n}$***—which happens in the future. Like previously, first let us proceed with a non-incentive compatible and thus *simplified* version of the algorithm presented in (Fig. 8) below:

Notice that letting the decentralized state-machine verify the path traversed always requires two round trips between the sender and any *effective* round leader. *Here* all we care for is for the decentralized consensus to assess the decrypted E—the identity of a particular leader is of no significance at all. The value $M_i^{ID}$ within *UD* released by $\Lambda_1$ in $Epoch_{i+n}$ allows $\Lambda_2$ to look up the corresponding $E_k$ from the decentralized storage, values $\Lambda_1^{ESK}$, $E_S$ allow $\Lambda_2$ to decrypt all the layers of $E_k$, while comparing $E_S$, with $E_{S'}$, allows round leader the ensure that the sequence of intermediaries was not tempered with. Indeed, mangling with any $E_i$, $\in$ E would result in $E_{S'} \neq E_S$. For the tempering to be unexposed by $\Lambda_2$, the attacker would need to become in possession of the secret value $E_S$. As a consequence, for the protocol to be valid under $TM_2$, $\Lambda_1$ should release $E_S$ only, as soon as, it is positive that E made it into the data block buried at least at blockchain depth $\geq B_S$. Indeed, it is crucial for $\Lambda_1$ to withhold $E_S$ until E makes its way towards the decentralized storage and becomes *immutable*. Indeed, $\Lambda_1$initiates $Epoch_{i+n}$ only *after* $PE_k \sim$ - had been buried deep, at the discretion of $\Lambda_2$, within one of the data blocks, but at least at blockchain depth $\geq B_S$.—the depth at which data blocks are deemed to be irreversible. Since, we do not want anyone to be able to trick the majority into believing in another version of E, once UD had been released, we await UD to make it deep into the history of events before proceeding to $Epoch_{i+n}$. Now, since $\Lambda_1$ needs to get a confirmation before proceeding;—stemming from the assumption that all nodes are well connected to the network, $\Lambda_1$ is presumed to be actively monitoring the incoming blockchain blocks, awaiting an appropriate moment to initiate the second phase of the protocol—$Epoch_{i+n}$.

Let us wonder; in Step 2 how is $\Lambda_1$ to be assured of the fact that $M_i$PE arrived at $\Lambda_2$? Here, up until now, there had been no incentive for $\Lambda_1$ to act in $Epoch_{i+n}$—had it not had any additional information to send; also no incentive for $\Lambda_2$ to include PE into the decentralized data-store for it to become public and non-malleable and also no incentive for intermediaries to act in Step 3 during $Epoch_{i+n}$. Additionally, how is $\Lambda_1$ to provide $\Lambda_2$ with *UD*? Even if, in $Epoch_i$, $\Lambda_1$ managed, somehow, to incentivize others to route $M_i PE_k$ towards $\Lambda_1$—the same data-path might become unavailable in $Epoch_{i+n}$. Thus, the second stage of the algorithm requires a way of providing incentive for additional, possibly unknown intermediaries encountered while M traverses the network. We need to modify the algorithm, let as proceed.

Now, while having in mind algorithm $PA_2$ let us bring into the picture the decentralized datastore and let us unfold the incentive-compatibility requirements by affixing $PA_2$ deeper into our rational, incentive-compatible realities.

***Assumptions*** It is in the intention of $\Lambda_1$ for $\Lambda_1$—the latter being any round leader, to receive information M; which is why $\Lambda_1$ is to cover the 'propagation fees' paid out to agents whose involvement is indispensable. These include intermediaries and round leaders from both $Epoch_i$ and $Epoch_{i+n}$ who are to include the necessary meta-data into the decentralized state machine's data-store. One can easily imagine an opposite scenario, say a content delivery network, in which case, it would be in the intention of $\Lambda_2$ to have M delivered from $\Lambda_1$. Here, we are not concerned with the nature of M and *do not* require M to be included within the data store comprising the decentralized state-machine. Additionally, $\Lambda_1$ needs not to be aware of any intermediaries besides the (sub)set of its imminent neighbours ($N_{E_1}$) chosen at the discretion of.

$\Lambda_1$. Further, if M is a 'transaction'—we desire it to be processable as soon as possible, thus M is delivered to destination already by the end of $Epoch_i$**.** Indeed, the sole purpose of $Epoch_{i+n}$ is to assure game-theoretical soundness of the protocol for everyone involved. It is assumed that M has a unique identifier $M_{ID} \in$ M. The protocol rewards intermediaries of a unitary successful data path *only*. In case of multiple data-paths, the shortest one should prevail. Our reward function (Sect. 6) does not rely on the property of path lengths in order to achieve sybil-proofness. Even though explicit preference of shortest data-paths could have the benefit of saving up on storage for the entire network—it is not enough to discourage sybil nodes in 1-connected networks, also we do not assume the encountered leaders to be willing to participate for more than a single round and so being agnostic to the long-term wellbeing of the entire system. Intuitively, under the additional global premise of agents being expectation maximisers we need to affect leaders' atomic decisions. We—as designers, have the power of choosing shortest-path incentive compatible.

$E^e$—optional endorsement when $PA_2$ invoked by external algorithm.

Let us assume the existence of the following sets of intermediaries:

**$Epoch_i$:**

***Required Preconditions***: $\Lambda_1 \ni \mathbb{A}_1$ prepares $\mathbb{A}_1^{\mathrm{PK}}/\mathbb{A}_1^{\mathrm{SK}}$, $\mathbb{A}_1^{\mathrm{ESK}}/\mathbb{A}_1^{\mathrm{EPK}}$, $\mathrm{M}_i^{ID} \in \mathrm{M}_i$, $\mathrm{E}_S$, intermediaries prepare $\mathrm{N_i^{PK}} \forall \mathrm{N}_i$; $i \geq 0 \leq k$

**Optional:** $\mathrm{N_i^{AD}}/\Lambda_1^{AD}$ – – arbitrary additional data included by intermediary/sender

1. $\Lambda_1$ prepares: $\mathrm{E}_0 = Enc_{\mathbb{A}_1^{ESK}}(\mathrm{E}_S)$

   Now, dependently on mode:

   - *sealed mode*: $\Lambda_1$queries $\mathrm{N}_0$ for $\mathrm{N_0^{PK}}$ and performs $\mathrm{M} = \Phi\left(\mathrm{M}, \mathrm{N_0^{PK}}, \mathrm{R} = \mathrm{E}_S, \mathbb{A}_1^{\mathrm{SK}}\right)$
   - *open mode:* the processing jumps to pt. 2

2. Has $\Lambda_1$not *cleared* previous transmission of information $M_{i-m}$to which $M_{i-m}^{ID}$ corresponds ~ $\boldsymbol{Epoch_{i-m}}$?
   - **False:** $\Lambda_1$ assumes $UD = \emptyset$, as there's nothing to 'unlock'
   - **True:** and only once $M_{i-m}PE_k$ to which $M_i^{ID}$ corresponds had been included within the decentralized store at depth $\geq \mathrm{B_s}$ by previous leader does $\Lambda_1$ assume $UD = Sig_{\mathbb{A}_1^{\mathrm{SK}}}\left[\mathrm{M_{i-m}^{ID}}, \mathbb{A}_1^{\mathrm{ESK}} \sim \mathrm{M_{i-m}PE}_k,\ E_S' \in M_{i-m}PE_k\right]$

     Note $\mathrm{E'_S} \neq \mathrm{E_S}$ is required.

In any case $\rightarrow P = Sig_{\mathbb{A}_1^{\mathrm{SK}}}\left[\mathbb{A}_1^{\mathrm{PK}}, UD, SHA_{256}(M), \Lambda_1^{AD}, Sig_{\mathbb{A}_1^{\mathrm{SK}}}(E_S)\right]$

and $\Lambda_1$ provides $\mathrm{N}_0$ with $M_iPE_0 = Sig_{\mathbb{A}_1^{\mathrm{SK}}}[M_i, P, E_0]$

3. Each $N_i$ where $0 \leq i < k$ does $E_i = Enc_{\mathbb{A}_1^{\mathrm{EPK}}}([E_{i-n}, N_i^{PK}, N_i^{AD}])$and retransmits $M_i\mathrm{PE}_i = [M_i, \mathrm{P}, \mathrm{E_i}]$ to $\mathrm{N_{i+1}}$

**$Epoch_{i+n}$:**

4. Once $M_i\mathrm{PE}_k$ arrives at round leader $\Lambda_2$:

- **Open mode:** $\Lambda_2$ includes $\mathrm{PE}_k$ into storage
- **Sealed mode**: $\Lambda_2$ does $\Phi'(M_i', \mathrm{E}')$ to verify the integrity of bonding $M_i' \leftrightarrow \mathrm{E}$ and includes either $M_i\mathrm{PE}_k$ or $[SHA_{256}(\mathrm{M}_i), P, \mathrm{E}_k]$ into storage, depending on whether M itself or its image is required by the consensus mechanics respectfully.

5. If, $\mathrm{UD} \in P \neq \emptyset \rightarrow \Lambda_2$: looks up $M_{i-m}\mathrm{PE}_k$ (from an epoch which happened $m$ epochs ago) by $M_{i-m}^{ID} \in \mathrm{UD} \in \mathrm{P}$, from within the decentralized storage, decrypts the content of $E \sim \boldsymbol{Epoch_{i-m}}$ by doing $Dec_{\mathbb{A}_1^{ESK'}}\left(\left[\mathrm{E_i}, \mathrm{N_i^{PK}}\right]\right) \forall i \leq \mathrm{k} \forall \mathrm{E_i} \in E$ effectively getting to know the identities of intermediaries from $\boldsymbol{Epoch_{i-m}}$. In sealed mode, $\Lambda_2$ also does $\Phi'(\mathrm{M}', \mathrm{E}')$ to verify the integrity of bonding $M_{i'-m} \leftrightarrow \mathrm{E}$ assuming $\mathrm{M}_i$ or its image had been previously included and now retrieved from storage. Also, $\Lambda_2$verifies $\mathrm{E}_{\boldsymbol{S}'}$ to be equal to the retrieved value $\mathrm{E}_{\boldsymbol{S}}$ found within $E_0 \sim M_{i'-m}$ to verify the integrity of E itself. If positive, the algorithm concludes with a properly verified list of intermediaries.

**Fig. 8** High-level representation of path assurance algorithm $PA_2$

$N_{E^e} = \sum N_i \in E^e$ when $E^e \neq \varnothing$ otherwise. It is the set of routing identities from an external invocation of another path assurance protocol.

$N_E \sim Epoch_i$—the set of routing identities from $Epoch_i$

$N_{E_k}$—the final set of rewarded intermediaries rewarded for propagation of M (including nested invocations of protocols) including propagation of possibly additional meta-data required to end-up within the decentralized storage.

$C$ is assumed to be a protocol defined constant such that $C < B_S$ due to an assumption that $\hbar \ll R_L$.

*Threat model* ($TM_3$): Extends $TM_2$. Intermediaries are to remain anonymous until E $\sim$ $\boldsymbol{Epoch_i}$ is buried deep within the history of events at depth assumed as irreversible ($\geq E_S$) which is when $Epoch_{i+n}$ begins. After that, nobody (including $\Lambda_1$ and $\Lambda_2$) should be able to trick the majority into believing in a version of $M_i\mathrm{PE}_k$ other than the one initially published by $\Lambda_1$. Under these assumptions and under validity of the sybil-proofness Theorem of the reward distribution function laid out in Sect. 8—the protocol should allow for a Sybil-proof distribution of rewards when $\Lambda_2$ remains unknown to $\Lambda_1$ at the beginning of $Epoch_i$. Shall the consensus mechanism fork out the already published block $\mathrm{B}_i$ once *unlocking data* had been already released—the reward process is to be considered as *possibly compromised*—shall an adversary come into possession of $UD^{out}_{\Lambda_1}$ and be able to rewrite the content of E, which is assumed as unfeasible under the assumption that data bellow $\geq E_S$ is deemed as irreversible. The secret ephemeral key pair used for encryption, which we employ to allow for reusability of the sender's identity-related key-pair, is to be compromised in $Epoch_{i+n}$ once everyone may decrypt the endorsement.

Now we may proceed with an incentive-compatible version of $PA_2$. Recall that $\mathbb{A}_1^{AD}/\mathrm{N}_i^{AD}$ represented optional, complementary sets of information. Thence, now these might encapsulate a complementary endorsement $E^e$, when invoked by another protocol. Sample case is yet to follow, namely for propagation of system extrinsic data-packets. Additionally, these fields are now used to carry incentive-related data-structures. Later, when we arrive at incentivized propagation of system-extrinsic datagrams, they will be used for recursive invocations of protocols (algorithms) (Fig. 9).

#### 6.1.3 Incentive compatible path assurance algorithm $\mathrm{PA}_2$ (two modes: open and sealed)

Agents may perform arbitrary consecutive invocations of the protocol, thus theoretically allowing for the protocol never to finish, with intermediaries not being rewarded at all. Note that such situation does not lead to a game-theoretical Nash equilibrium. That is because, under our global assumption of agents being expectation maximisers, unwilling to loose, it is rational to presume that senders would be willing to get back the frozen assets from previous epochs. Indeed, at any point in time, there will always be at least a single, set of intermediaries who did not get rewarded as of yet, with one set corresponding to the latest epoch. Here, with each invocation of the protocol, sender provides 'unlocking data' for its previous invocation. Note that PE values are signed by sender, thus when a full-node notices same $\Gamma_{N_i}$ being re-used for transmission for more than one M, it would include a proof of such fact (containing the two signed PE values), while issuing additional penalty to sender, with the on spotter receiving reward in the same amount the mischievous node is penalized with (preventing out-of-thin air reward formation).

Recall—other nodes maintaining consensus over state-domains would always verify updates to the decentralized State-Machine's variables based on data present within $B_i$ by reprocessing available information. In case of any discrepancies, as seen by the majority- consequently nobody would be rewarded.

In Step 4 the purpose of $C \times B_S$ delay is to allow for $\mathcal{L}_i$ to be penalized by $\mathcal{L}_{i'}$ (Step 9) for a possibly excessive path length. Separation of the protocol into two independent epochs enables for a-synchronicity. Notice that the decentralized state machine does not need to wait upon $\Lambda_1$ to release *UD*. First, $\Lambda_1$ needs to be assured $\mathrm{PE_K}$ had been buried deep within storage by $\Lambda_2$ before it initiates $\mathrm{Epoch}_2$. Otherwise, possibly nested endorsements within E would not become non-malleable opening the door to Sybil attacks if the history of events forked and attackers got to know [$\mathrm{E}_S$, $EphSK_C$]—the secrecy of which the sybil-proofness guaranties rely upon ($TM_4$).

After all the complexity of $\mathrm{PA}_2$ let us now proceed to an incentive compatible version of an earlier algorithm ($\mathrm{PA}_1$).

***Assumptions*** $\Lambda_2$ is assumed as a round leader known through identity $\mathbb{A}_2$ to which corresponds a public key $\mathbb{A}_2^{PK}$. It is in the intention of agent $\Lambda_1$ for $\Lambda_2$ to receive information M; which is why $\Lambda_1$ is to cover the 'propagation fees' paid out to agents whose involvement is indispensable. These include intermediaries and the final destination $\Lambda_2$.The protocol rewards intermediaries of a successful data path only, one chosen at the discretion of $\Lambda_2$ (in case of alternatives).

*Threat model* ($TM_5$): Extends $TM_1$. Intermediaries are assumed to be willing to get to know their peers so to be able to choose favourable dealings. When in anonymous data-routing mode we take use of the fact that intermediaries are unable to get to know their peers during $\boldsymbol{Epoch}_1$.

**$Epoch_i$:**

**Required Preconditions:** $\Lambda_1 \ni \mathbb{A}_1$ prepares: secret value Es, key pairs $\mathcal{A}_1^{EPK}/\mathcal{A}_1^{ESK}$ and $\mathcal{A}_1^{PK}/\mathcal{A}_1^{SK}$, $R_L, R_N, {}^{C}\mathbb{A}_T^S, \mathbb{A}_1^{AD}$, possibly $E' \in \mathbb{A}_1^{AD}$. Each intermediary node $N_i$ prepares: $N_i^{PK}/N_i^{SK}$ and $\Gamma_{N_i}$. Naming convention $E_{S'}$, $\mathcal{A}_1^{ESK'} \sim \boldsymbol{Epoch_i}$ represents retrieved, corresponding values from an $i^{th}$epoch.

1. $\Lambda_1$ picks a set of initial neighbors $N_{E_1}^0$ Also $\forall N_i \in N_{E_1}^0$; $\Lambda_1$ sets $PID_i$ and prepares $\Lambda_{AD_1} = [R_L, R_N, \mathbb{A}_{ST}]$
   $P = Sig_{\mathbb{A}_1^{SK}}[PID_i, SHA256(M), \Lambda_1^{AD}, \mathbb{A}_1^{PK}, Sig_{\mathbb{A}_1^{ESK}}(E_S)]$
   - *Participated in a previous transmission* ~ $\boldsymbol{Epoch_{i-n}}$*?*
     - True: $UD_{\Lambda_1}^{out} = \left[M_{i-n}^{ID},\ Sig_{\mathbb{A}_1^{SK}}\left[E_S \sim \boldsymbol{Epoch_{i-n}}, \mathbb{A}_1^{ESK} \sim \boldsymbol{Epoch_{i-n}}\right]\right]$
     - False: $UD_{\Lambda_1}^{out} = \emptyset$
   - *Sealing feature enabled?*
     - True: $M = \Phi\left(M_{i-n}^{ID}, N_0^{PK}, R, \mathbb{A}_1^{ESK}\right), E_0 = Enc_{\mathbb{A}_2^{PK}}(N_0^{PK}, R)$
     - False: $E_0 = Enc_{EphPK_C}(E_S, \mathbb{A}_1^{AD})$

   $M_i = \left[M_i, UD_{\Lambda_1}^{out}\right]$; $M_iPE_0^i = [M_i, P, E_i]$ and delivers $M_iPE_0^i$ to each corresponding $N_i \in N_{E_1}^0$
2. Then, each $\forall i \in \{1, .., k-1\}$ $N_i$computes $E_i = Enc_{\mathbb{A}_1^{EPK}}\left(E_{i-1}, N_i^{PK}, N_i^{AD} = \Gamma_{N_i}\right)$; $M_iPE_i = [M_i, P, E_i]$and transmits $M_iPE_i$ to$N_{i+1}$, so that$\exists E_i \forall i \in \{1, .., k\} \in E$, resulting in unique $E$ for any given data-path.
3. If $\Lambda_1$ has no more data to send it waits until it gets to know that Point 4 below accomplished. *(passive check).* Otherwise $\Lambda_1$ might continue dispatching new $M_iPE$ $; i \to \infty$ packages starting again from Step 1.
4. Once $M_iPE_k$arrives at round leader $\mathcal{L}_i = N_k$, $\mathcal{L}_i$ includes $PE_k = [P, E_k]$ within $B_X$. At this stage, the processing of $M$ might take place. If, $UD \in P \neq \emptyset \to \Lambda_2$ looks up $MPE_k' \sim \boldsymbol{Epoch_{i-n}}$ by$M_{i-n}^{\prime ID} \in UD \in P$ from within the decentralized storage, decrypts the content of $E \sim \boldsymbol{Epoch_{i-n}}$ by doing $Dec_{\mathbb{A}_1^{ESK'}}([E_i, N_i^{PK}]) \leq k \forall E_i \in E$effectively getting to know identities of intermediaries from $\boldsymbol{Epoch_{i-n}}$. In sealed mode $\Lambda_2$ also does $\Phi'(M)$to verify the integrity of bonding $M_{i-n} \leftrightarrow E$. Also $\Lambda_2$ verifies $E_{S'}$ to be equal to the retrieved value $E_S$ found within$E_0$ to verify the integrity of itself. Now, $\mathcal{L}_i$ issues reward $R_L$ to itself, the reward is not spendable for $C * B_S$rounds. It also deducts $R_L + {}^{C}\mathbb{A}_T^S + R_N$ from $\Lambda_1$, whose balance shall be insufficient, the protocol aborts.
   - **Optional:** if $E \sim \boldsymbol{Epoch_{i-n}} \sim \mathbb{A}_1^{AD}$and$\neq \emptyset$ and $R_T \in E' \to \mathcal{L}_i$ deducts $R_T$ from $\mathcal{L}_1$.
   - **Optional:** if $UD_{\Lambda_1}^{out} \neq \emptyset, \mathcal{L}_{i'} \to$verifies $E_{S'}$to be equal to$E_S \in B_i$. If not equal, the protocol aborts.

$\mathcal{L}_{i'}$ rewards intermediaries by issuing a set of self-signed transactions $T_i \forall N_i \in \{1, .. k-1\}$ as guided by the reward-distribution function (Section VIII) and depending on $N_{E_k}$**:**

- $N_{E'} \neq \emptyset \to R_{N_{E'}} = R_{N_{E_i}} = 0.5 * R_N \Rightarrow \mathbf{N_{E_k}} = N_{E'} \cup N_{E_i}$ and calculates $F_{\mathbb{A}}\left(N_i^{\mathfrak{T}_i}, R_{N_{E_i}}\right) \forall N_i \in N_{E_k}$
- $N_{E'} = \emptyset \Rightarrow \mathbf{N_{E_k}} = N_{E'} \cup N_{E_i} = N_{E_i}$

  $\mathcal{L}_i$ does $F_{\mathbb{A}}\left(N_i^{\mathfrak{T}_i}, R_N\right) \forall N_i \in N_{E_k}$

$\mathcal{L}_{i'}$ calculates a cryptocurrency-based penalty $\mathfrak{T} = \sum \Gamma_{N_i} \forall N_i \in N_{E_k}$ and increases its account balance by $(R_L - \mathfrak{T})$; also increments balance of $\mathbb{A}_1 \in \Lambda_1$ by ${}^{C}\mathbb{A}_T^S \in M_iPE_k'$(given back). Token pools of each intermediary $\in N_{E_k}$are marked as used up till $\Gamma_{N_{i+1}}$. All the updates are stored within $B_i$. In case $\mathcal{L}_{i'}$received $M_i$from multiple sources it prefers to choose one with higher $R_L$or if equal, the one with shorter endorsement since this results in a lower penalty. Thus the Threat Model assumptions in regard to the shortest data-path are upheld.

**Fig. 9** Incentive compatible path assurance algorithm $PA_2$

This leads to higher finalization guaranties under the assumed soundness of the Sybil-proof reward assignment function (Sect. 8). Note that in case the consensus mechanism forks out $B_i$, the reward mechanism is *not* considered as *potentially compromised*, under the presumed secrecy of $\mathbb{A}_1^{SK}$

Below, the processing related to various modes and features of $PA_1$ has been omitted for the clarity of

| $\boldsymbol{Epoch_1}$: |
|---|
| **Preconditions:** $\Lambda_1$ prepares key-pair $\mathbb{A}_1^{\mathrm{SK}}/\mathbb{A}_1^{\mathrm{PK}}$, payload $\mathrm{M_i}, \mathrm{R_L}, \mathrm{R_N}, \mathrm{R}, \Gamma_{\mathbb{A}_1} \neq \emptyset$ if $\Lambda_1$ wants to participate in the lottery; intermediaries prepare $N_i^{PK}, \Gamma_{N_i}$ ; $\Lambda_2$ prepares identity $\mathbb{A}_2$ along with an ephemeral key-pair $\mathbb{A}_2^{EPK}/\mathbb{A}_2^{ESK}$; $\Lambda_1^{AD} = [R_L, R_N]$; $N_i^{AD} = \Gamma_{N_i}$ |
| 1. $\Lambda_1$ queries $\mathrm{N_0}$ for its $\mathrm{N_0^{PK}}$<br>2. $\Lambda_1$ prepares $E_0 = Enc_{\mathbb{A}_2^{EPK}}\left(N_0^{PK}, \Gamma_{\mathbb{A}_1}, R\right)$ ; $\Lambda_1^{AD} = [R_L, R_N]$;<br>$P = Sig_{\mathbb{A}_1^{\mathrm{SK}}}\left[SHA_{256}(M_i), \mathbb{A}_1^{PK}, \Lambda_1^{AD}, \mathbb{A}_2^{\mathrm{PK}}, SHA_{256}(E_0)\right]$.<br>3. $\Lambda_1$ transmits $M_i PE_0 = [M_i, P, E_0]$ to $N_0$<br>4. Now recursively, each consecutive $N_i$ where $i \geq 0$ queries its neighbour for $N_i^{PK}$ while providing P to $N_i$. Each $N_i \in N_{E_2}^0$ sees $R_N$ and *might* look-up $\mathbb{A}_1$ within the decentralized datastore by $\mathbb{A}_1^{PK} \in P$ and verify balance of $\mathbb{A}_1$ to be sufficient. If $N_{i+1}$ agrees it accepts $M_i PE_i = Enc_{\mathbb{A}_2^{EPK}}\left(E_{i-1}, N_i^{AD} = \Gamma_{N_i}, N_{i+1}^{PK}\right)$ from $N_i$; if $N_i$ is not the destination -it attempts the same procedure with another neighbour $(N_{i+2})$<br>5. Once $M_i PE_k$ arrives at $\mathcal{L}_X$ - it extracts $\mathbb{A}_1^{PK} \in M_i PE_k$ and does $Dec_{\mathbb{A}_2^{ESK}}(E_i) \forall 0 \leq i \leq k \in E$.<br>6. If all the verifications of $PA_1$ succeed, $\mathcal{L}_i$ assigns rewards by issuing a set of self-signed transactions $T_i \forall N_i \in \{1, .. k\}$ as guided by the reward function. It calculates penalty $\text{Ћ} = \sum_1^k \Gamma_{N_i}$ also increases its account balance by $(R_L - \text{Ћ})$ and deducts $\sum F_A(R_N, N_i) \forall N_i \in E = R_N + R_L$ from $\Lambda_2$. In case $\text{Ћ} > R_L$ the protocol aborts.<br>Token-pools of each corresponding intermediary are marked to be used up until $\Gamma_{N_{i+1}}$.<br>Changes to internal variables of State Domains constituting the overall state of the decentralized state machine are contained within $B_X$.<br>7. The decisions made i.e. assigned rewards need to be verifiable by other nodes, thus $\Lambda_2$. Includes $[PE_k, \mathbb{A}_2^{ESK}]$ also with the owned ephemeral secret-key $\mathbb{A}_2^{ESK}$ within the block. $\mathbb{A}_2^{ESK} = \emptyset$ in non-anonymous mode. |

**Fig. 10** Iincentive compatible version of $PA_1$

presentation—we now focus on incentive-related *additions* (Fig. 10).

The so far introduced, incentive-compatible protocols allow for accountability of the traversed data-path. When combined with the proposed reward function from Sect. 8, they allow for Sybil-proof exchange of information when target $\Lambda_2$ is known or unknown. Still, so far, in all cases it used to be paramount for the destination of information to be a leader. Now, we shall extend upon what we have already proposed in order to introduce a mechanics suitable for when $\Lambda_2$ *is not* a leader thus allowing for a Sybil proof exchange of arbitrary, consensus extrinsic information also in eclipsed and 1-connected networks in the presence of a data-store with the property of non-malleability as time approaches infinity. Indeed, now the only requirement from $\Lambda_2$ will be to own an identity ($\mathbb{A}_2$) with a corresponding public key ($\mathbb{A}_2^{PK}$).

Thence, allowing for decoupling of the role of a recipient from that of a round leader. Still, the involvement of the latter and thus of the decentralized state-machine is compulsory. Here, potential leaders are the ones maintaining the decentralized data-store with the property of non- malleability as time approaches infinity. Additionally, we shall be accounting for both scenarios in which leader may be known or unknown in the wake of each round to $\Lambda_2$.

Here, efficiency of storage makes up an important dilemma. As has been described in our previous works where we introduced the notion of State-Full and State-Less blockchain channels,- including information into the

decentralized state-machine regarding every data-packet exchanged between source and destination would impose an unbearable amount of storage-overhead at high redundancy. To cope with that, here it is a wonderful opportunity to employ the previously introduced State-Full channels. The sender of information will reward parties involved through rewards described by frozen assets constituting fractions of previously 'bought' Transmission Pools. The assets will be steadily released to parties involved by uncovering secret hashes through Transmission Tokens. Parties involved can detect a change in the data-path by looking at E containing encrypted identities. Closing the state-channel and submitting [P,E] into the state-machine entails cost, thus it pays to wait as long as possible or until the path changes before doing so. State-channel can be closed by any party involved. All it takes is to submit [P,E], which contains sender's signature, into the full node who would do the processing. PE is included alongside every M and contains the first and the most recently released Transmission Token, also the number of tokens released in total during particular state-channel for efficiency of verification at the leader. Thanks to this number leader can verify if ending hash is at the supposed position without performing unnecessary blind brute-forcing until it comes at the supposed, claimed final value. In that sense PE constitutes the Transit Token described earlier. The total reward pool used to reward parties involved is calculated through accumulation of Transmission Tokens through the entire data transmission.

#### 6.1.4 Data routing algorithm $PA_3$

***Assumptions*** As previously, it is possible for intermediaries to be rewarded only if E successfully made it into the data store. We require M to be accessible to $\Lambda_2$ as soon as possible, thus no protocol-intrinsic encryption of M against $\Lambda_2$ exists. The algorithm requires target $\Lambda_2$ to be known to $\Lambda_1$ by its public key. Now, supporting extrinsic targets, the target destination *may not* have the ability to affect State-Domains on its own; thus now we provide $\Lambda_2$ with the ability to incentivize additional, possibly unknown intermediaries to deliver E to a possibly unknown leader. In order to maintain game-theoretical soundness and under the global assumption of all agents, also those extrinsic to consensus-mechanics, being expectation maximisers, we introduce yet another reward $R_T$ paid out to $\Lambda_2$—a reward covered at the expense of $\Lambda_1$, stemming from the assumption that $\Lambda_2$ is agnostic to M and that the incentive of delivery resides at source,—assuring that an incentive for $\Lambda_2$ to proceed *after* it received M exists. Note that $R_T$ had already been accounted for in the embodiments of previous algorithms. Again, once information reaches destination we need to incentivize destination to route E across additional intermediaries, who also need to be incentivized on the way to a leader. Here it is assumed that $R_N$ constitutes a serialized representation of a Transmission Token one most recently used in a data transmission between $\Lambda_1$ and$\Lambda_2$. At any time, agents may verify usage of corresponding Token Pools for double spends.

*Threat model* ($TM_6$): Since the protocol *may* take use of both $PA_1$ and $PA_2$, corresponding specialized threat models apply. No reward shall be issued to any agent *unless* M reaches its destination ($\Lambda_2$) Onlookers can verify reward values at any time since $R_L$, $R_T$, $R_N$ values are public (Fig. 11).

Note that the value of $R_N$ accumulates over time as sender of information makes the most recent Transmission Token represent higher and higher subrange of the corresponding Token Pool. The purpose of $\mathbb{A}_2^{Sig} = Sig_{\mathbb{A}_1^{SK}}(ME_k)$ is to prove, in the eyes of the leader, that M reached target $\Lambda_2$. Assets from $R_N$ will be used to incentivize all intermediaries involved in transmission between $\Lambda_1$, $\Lambda_2$ and $\mathcal{L}_i$(possibly also $\mathcal{L}_{i'}$).Thus, the total reward issued to intermediaries is equal to $R_N$. The reward assignment function takes into account *all* intermediaries. Note that in Step 6, $\Lambda_2$ is obliged to specify $\mathbb{A}_T$, $R_{L_{1,2}}$, and $R_N$. While the last two are copied from $ME_0$, the first is at disposal of $\Lambda_2$ and its $\Lambda_2$ who will be responsible for participating in a potential $\boldsymbol{Epoch}_2$ of $PA_3$.

The benefits include the possibility for intermediaries to abort the procedure at any point by verifying consistency endorsement E. Intermediaries are not blinded i.e. encrypted, thus, the finalization guaranties are lower (further discussion ahead) compared to the protocol presented below (Fig. 12):

We now have a solid groundwork for facilitating exchange of transactions, data blocks and consensus extrinsic information for both BFTL and LFTB decentralized state-machines. Let us now allow for some game-theoretical divagations and visualizations.

### 6.2 Transactions

Thanks to $PA_1$ and $PA_2$ intermediaries do not know who their neighbours are. Thus, routers cannot guess what rewards their neighbours are to claim. These properties were required to maintain high finalization guaranties. The gist is that during $\boldsymbol{epoch}_1$ T from $\mathbb{A}_C$ needs to be delivered to $\mathscr{L}_1$. Intermediary nodes include themselves into E. Consequently, in accordance to $PA_2$, in $\boldsymbol{epoch}_{i+1}\mathbb{A}_C$ needs to release the endorsement's embedded secret $E_S$ and $EphSK_C$ *so that the* endorsement can be decrypted by $\mathscr{L}_1$. When M is assumed to make up transaction T both $PA_1$ and $PA_2$ can be used to facilitate sybil-proof propagation in both BFTL and LFTB machines.

***Epoch*$_1$:**

**Preconditions:** $\Lambda_1$ prepares key-pair $\mathbb{A}_1^{\mathrm{PK}}$ / $\mathbb{A}_1^{\mathrm{SK}}$ payload, $\mathrm{M_i}, \mathrm{R_L}, \mathrm{R_T}, \mathrm{R_N}$ ; intermediaries prepare: public-keys $N_i^{PK}$ and $\Gamma_{N_i}$ $\Lambda_2$ prepares key-pair $\mathbb{A}_2^{\mathrm{PK}}$ /$\mathbb{A}_2^{\mathrm{SK}}$

1. Now, using ***PA*$_1$**: $\Lambda_1$ queries $\mathrm{N}_0$ for its $N_0^{PK}$
   - *sealed mode:* $\Lambda_1$ does $\mathrm{M_i} = \Phi(\mathrm{M_i}, \mathrm{N_0^{PK}})$,
   - *open mode:* processing proceeds to Point 2
2. $\Lambda_1$ prepares $E_0 = N_0^{PK}, \Lambda_1^{AD} = [\mathrm{R_L}, \mathrm{R_T}, \mathrm{R_N}]$

   $P = Sig_{\mathbb{A}_1^{SK}}[SHA_{256}(M_i), \Lambda_1^{AD}, \mathbb{A}_1^{\mathrm{PK}}, \mathbb{A}_2^{\mathrm{PK}}, E_0]$ and delivers $M_i\mathrm{PE}_0 = [M_i, P, E_0]$to $N_0$

   ***Note*:** $M_i$ is *not required* for $VerSig_{\mathbb{A}_1^{\mathrm{PK}}}(P)$.
3. Each consecutive $N_i$ where $i \geq 0$ queries its neighbour for $N_{i+1}^{PK}$ before preparing $E_i = Sig_{\mathbb{A}_1^{\mathrm{SK}}}\left[E_{i-1}, N_i^{AD} = \Gamma_{\mathrm{N}_{i+1}}, N_{i+1}^{PK}\right]$ and retransmitting $M_i\mathrm{PE}_i = [M_i, P, E_i]$ to $N_{i+1}$, **Note:** indexes are scoped to specific variables.
4. Each intermediary, including $\Lambda_2$ assumes $\mathbb{A}_X^{PK} = \mathbb{A}_1^{PK}$ for $E_0$ and $\mathbb{A}_X^{PK} = N_i^{PK} \forall E_i; 0 \leq i \leq k$ and does $VerSig_{\mathbb{A}_X^{PK}}(E_i) \forall 0 \leq i \leq k$, effectively verifying the entire sequence of identities $N_i^{PK} \forall i \leq k \in E$.
   - If the sequence is invalid, the protocol *aborts*.
   - Otherwise, the protocol proceeds
5. Once $M_i PE_k$ arrives at $\Lambda_2$ and if $\Lambda_2$ happens to be the round leader, the processing proceeds to Step 7 (assuming $\mathcal{L}_i = \Lambda_2$ )
6. Now, $\Lambda_2$ chooses between two methods for propagation of $M_i$ :

- ***Leader unknown*:** $\Lambda_2$ employs $PA_3$ to incentivize transmission of $PE_k$ towards a round leader. $\Lambda_2$ parametrizes invocation of $PA_3$ with $\Lambda_1^{AD} = Sig_{\mathbb{A}_2^{\mathrm{SK}}}\left[P, {}^{C}\mathbb{A}_T^S\right]$Note $\Lambda_2$ just made an obligation to participate in possibly two epochs of $PA_3$ by laying out a signature. The transmission commences to conclude at Step 7. Note:$M_i$ is omitted as it already reached its destination.
- ***Leader known*:** $\Lambda_2$ assumes $\Lambda_2^{AD} = Sig_{\mathbb{A}_2^{\mathrm{SK}}}\left[P, {}^{C}\mathbb{A}_T^S\right]$ , queries its neighbor for $N_{i+1}^{PK}$before preparing$E_{\Lambda_2} = Sig_{\mathbb{A}_1^{\mathrm{SK}}}\left[E_{i-1}, N_i^{AD} = \Lambda_2^{AD}, N_{i+1}^{PK}\right]$ and transmitting $M_i$, $E_{\Lambda_2}$ to$N_{i+1}$. Further intermediaries proceed with $PA_1$ from Point 3 and after $|E_2|$ additional hops possibly arriving at a round leader, thus resulting in $M_i$ ,$E_k'$ where $k' = k + |E_2|$Note: $\Lambda_2$ not counted.

7. Round leader(s) taking care of the process (possibly two if $PA_3$ was employed in Step 6 do integrity validation and checks if final E contains $\mathbb{A}_2^{Sig}$, if not the processing aborts. Otherwise, all intermediaries are rewarded as dictated by the Sybil-proof reward function (Section 8) using value represented by$R_N$, leaders are rewarded from $R_L$, as per the chosen traversal method (Point 6). Note that balance of $\mathbb{A}_1$ will decrease by $\sum F_A(R_N, N_i) \forall N_i \in E = \mathrm{R_L}, \mathrm{R_T}, \mathrm{R_N}$ while balance of $\mathbb{A}_2$ increase by $R_T$. Additionally, Token-Pools of each corresponding intermediary will be marked to be used up until $\Gamma_{N_{i+1}}$. Leader token-pools' usage for double-spends.

**Fig. 11** Data routing algorithm $PA_3$

### 6.3 Game-theory behind *Epoch*$_i$ and *Epoch*$_{i+1}$

In ***Epoch***$_1$ it is obvious that $\mathbb{A}_C$ has an incentive to take part since he wants to have the transaction processed. Routers have an incentive to participate since they want a cut from $R_N$ (Fig. 13).

| $\boldsymbol{Epoch_1}$: |
|---|
| **Preconditions:** $\Lambda_1$ prepares key-pair $\mathbb{A}_1^{\mathrm{SK}}/\mathbb{A}_1^{\mathrm{PK}}$payload M, reward pools$\mathrm{R_L}, \mathrm{R_N}, \mathrm{R}_T$,also a large pseudo-random number R (key), intermediaries prepare public-keys $N_i^{PK}$ representing their identities. |
| 1. $\Lambda_1$queries $N_0$ for its $N_0^{PK}$<br>2. $\Lambda_1$ does $M = \Phi\left(M, N_0^{PK}, R, \mathbb{A}_1^{SK}\right)$, prepares $E_0 = Enc_{\mathbb{A}_2^{PK}}(N_0^{PK}, R)$ *note*: $N_0^{PK}$and R cannot be recovered by reading M.<br>3. $\Lambda_1$ transmits M$E_0 = Sig_{\mathbb{A}_1^{SK}}[M, \mathbb{A}_1^{PK}, E_0]$ to $N_0$<br>4. Each consecutive $N_i$ where $i \geq 0$ queries its neighbour for $\mathrm{N_i^{PK}}$ before preparing $E_i = Enc_{\mathbb{A}_2^{\mathrm{PK}}}\left(E_{i-1}, N_i^{AD}, N_i^{PK}\right)$and transmitting $ME_i = \left[M, \mathbb{A}_1^{\mathrm{PK}}, E_i\right]$ to $N_{i+1}$<br>5. Once $ME_k$ arrives at $\Lambda_2$ it extracts $\mathbb{A}_1^{\mathrm{PK}}$ and does $Dec_{\mathbb{A}_2^{\mathrm{PK}}}(E_i)\forall 0 \leq i < k \in E$also retrieves R from$E_0$ and verifies M by comparing HM with$HMAC(N_0^{PK}, R)$ , effectively verifying the entire sequence of identities $N_i^{PK}\forall i \leq k-1 \in E$.<br>• On failure the protocol aborts.<br>• Otherwise, the protocol proceeds<br>6. If, $\Lambda_2$ happens to be a round leader, the processing jumps to step X ($\mathcal{L}_i$=$\Lambda_2$)<br>7. Now, depending the situation:<br>• Leader unknown: $\Lambda_2$ uses $PA_3$ to incentivize further transmission of M = ME$_k$ towards unknown leader. But first assigns $ASD = Dec_{\mathbb{A}_2^{\mathrm{PK}}}(E_i)\forall 0 \leq i < k \in E$<br>• Leader known: $\Lambda_2$ re-encrypts $E_k = Enc_{\mathcal{L}^{\mathrm{SK}}}\left(Dec_{\mathbb{A}_2^{\mathrm{PK}}}(E_i)\right)\forall 0 \leq i < k \in E$ and additional intermediaries perform same operation depicted in pt. 4.<br>8. M after $\lvert E_2 \rvert$additional hops possibly arriving at round leader $\mathcal{L}$, results in ME$_k'$, where $k' = k + \lvert E_2 \rvert$**Note:** $\Lambda_2$ is not counted.<br>9. $\mathcal{L}$ does same validations as $\Lambda_2$ in step 4, and rewards intermediaries as dictated by$F_A(R_N, N_i)\forall N_i \in E$, also $\Lambda_2$ assigns itself reward $R_L$ (or partial reward as dictated by $PA_3$ in case $PA_3$ was used in 7th step), assigns reward $R_T$ to $\Lambda_2$ and decreases balance of $\Lambda_1$ by $\sum F_A(\mathrm{R_N}, N_i)\forall\, N_i \in \mathrm{E} + \mathrm{R_L}+\mathrm{R_T}$ token-pools of each corresponding intermediary are marked to be used up until $\Gamma_{N_{i+1}}$. |

**Fig. 12** Blinded version of $PA_3$

### 6.4 Game-theory at $N_i$

Each agent can see $R_N$ and decide whether it wants to risk value of $\Gamma$ by including it within the endorsement. So whether or not to retransmit is a function of $(\mathbb{A}_i^{\mathfrak{B}\mathrm{I}} \in N_i, L, R_{L_{1,2}}, {}^{C}\mathbb{A}_T^S)$ *i.*

Technically, $N_i$ can always include a Sybil but:

- At low $\mathbb{A}_i^{\mathfrak{B}\mathrm{I}}$ he is most likely not to receive rewards at all and loose
- No matter the number of Sybil nodes, the ones with highest $\mathfrak{B}_I$ would profit most
- $N_i$ can judge the current path length by looking at the endorsement

### 6.5 Data arrived at $_1$

Leader $\mathscr{L}_1$ chooses T endorsed by highest $R_{L_{1,2}}$. He cannot decide based on routers since at this point they are unknown. Leader will be rewarded with the value of $0.5 \times (R_{L_{1,2}} - \mathfrak{B})$ thus the incentive is to choose the shortest path.

Game Theory during $\boldsymbol{Epoch_2}$

- The precondition for initiation of $\boldsymbol{Epoch_{i+1}}$ is that $\boldsymbol{TE_k}$ is buried at a sufficient blockchain depth which we assume as constant for the protocol, but the value can be easily decided upon by the decentralized governance. In $\boldsymbol{Epoch_{i+1}}$, client $\mathbb{A}_C$ has an incentive to release

$[EphSK_C, S]$ since otherwise it would loose ${}^{C}\mathbb{A}_{T}^{S}$. Also, $\mathcal{L}_2$ has an incentive to participate since it receives $0.5 \times (R_L - \text{Ħ})$.

## 6.6 Data-blocks

Transactions are data structures encapsulating instructions changing one or more variables of the decentralized state-machine. Naturally, the incentive to act, is on the side of the client who wants to make the particular change. Typical blockchain implementations incentivize leaders to act upon the received transactions by introducing a protocol-intrinsic incentive in the form of transaction-fees specified and covered by clients. Still, as has been described already, these implementations fail to incentivize intermediaries responsible for transaction deliveries. Also there is no incentive for potential leaders to transmit transactions to other nodes if for whatever reason they are unable to do the processing. In our proposal we have already introduced algorithms that provide additional Sybil proof incentive for intermediaries (under validity of the Sybil-proofness Theorem of the reward function described in Sect. 8) within any open-system where leader is either known or unknown in advance. Another important blockchain intrinsic aspect is the propagation of data blocks. Let us proceed with our proposal targeting the matter. Here, we employ the nomenclature of Block Sets (BS) to underline full support of systems in which leader may release multiple consecutive blocks until another leader is elected.

### 6.6.1 Data Blocks Routing Algorithm (path assurance + incentives)

***Assumptions*** Routers have an incentive to propagate data blocks since they want to have their re-transmission rewards recognized by other nodes. The more routers propagate, the higher the expected rewards as proven further. When re-transmitting, inseparable meta-data is generated and re-transmitted along the data-block. Once both (the meta-data and the data-block) arrive at a leader it issues propagation-rewards accordingly. Nodes/leaders seeing that meta-data alter the state of variables within their local instances of the decentralized state-machine based on data contained within these very blocks thus taking the rewards into effect, updating balances of routers' accounts. By propagating they also increase the chance of reaching a round leader in the first place. There might be other incentives influencing withholding of data-blocks depending on other blockchain's specific incentives and network conditions [6]. It is assumed that longest and/or heaviest path wins, thus other nodes have an incentive to build upon blocks published by others. As has been shown in other works, this depends on nodes' capa-

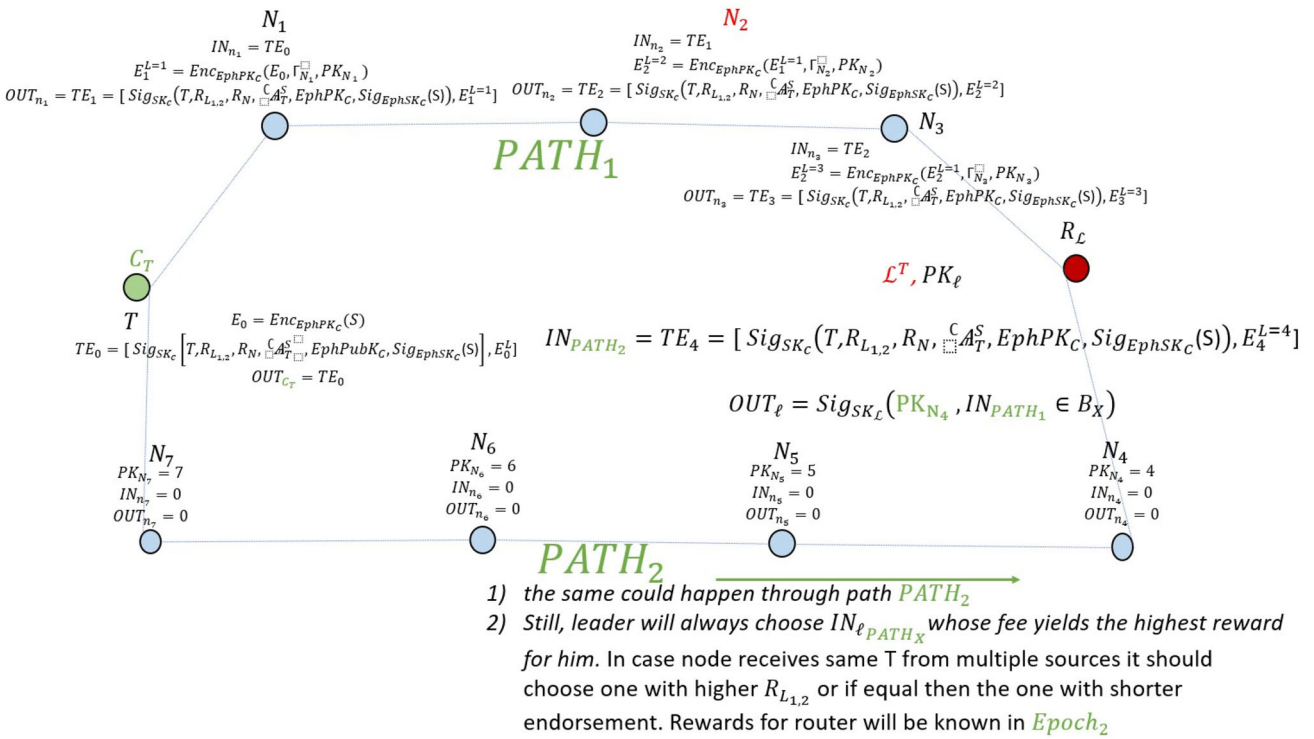



**Fig. 13** Visualization of transaction arriving at leader through two alternative data paths

bilities versus the rest of the network. These game-theoretical rationale have been well depicted in [5].

*Threat model*: As usually, misbehaviours are driven by incentives. Here, intermediaries might be trying to replace the path the data-block took in an attempt to introduce Sybil-identities. Also, the recipients (potential leaders) might be trying to separate M = B from the endorsement E, in order to replace identities of intermediaries with their own. We assume intermediaries to be trying to get to know their peers during blocks' propagation so to increase their chances of winning while decreasing their chances of losing (Fig. 14).

## 7 Proof of sybil-proofness

We now proceed with proving that the reward function employed by the previously depicted algorithms is sybil-proof indeed. Consequently, under the premise that the reward function is being used to reward data-propagation within the previously depicted games, it renders data-exchange within these games sybil-proof under validity of the provided theorem.

### 7.1 Preliminaries

The game agents participate in can be compared to a First-Price Sealed-Bid Auction (FPSBA) [14, 23–25] where the auctioneer is the round leader $\mathcal{L}$ and the item being auctioned off is the propagation fee $F$ set by $^{C}\mathbb{A}$. That is with some notable differences. Here, agents receive fractions of $F$, in proportion to how they are rated *in scope of a path* by an exponential identity rating function $\epsilon$. All bids are collected by $\mathcal{L}$. Agent would lose if his reward is dominated by others to turn out to be below $\Gamma_V$ (*the payoff might be negative*). $\mathcal{L}$ does not know the bids and cannot issue rewards unless in $\boldsymbol{epoch_i}$ ($^{C}\mathbb{A}$ *needs to release* $E_S$). Agents bid with $\Gamma_V$ which is constant for a given $\mathbb{A}$ (*for simplicity agents allowed to associate only a single Identity Token with any given identity*).

Notice that the uncertainty about the other agent's $\mathfrak{Tb}_I$, alongside the exponential reward drives $\mathfrak{Tb}_I$ to higher values and prevents inclusion of Sybil identities under the assumption of agents willing to make profit.

Assume nodes cannot get to know $\mathfrak{Tb}_I$ of other identities included in a data-path. *Assume agents act rationally. They are expectation maximisers and use all of their available resources to gain profit.* Without loss of generality, the fact that inclusion of an identity which further discourages inclusion of nodes owning low $\mathfrak{Tb}_I$ has been omitted. (Note that this further improves the Sybil-proofness property). Penalty for a leader to include paths of higher lengths also has been omitted. Let $\mathbb{A}_M$ be a set of legitimate identities owned by $\Lambda$, $\mathbb{A}_S$—the set of Sybil identities, R—the achievable reward, $R^{\Lambda}_{max}$—$\Lambda'$*s* amount of limited, system-intrinsic currency.

**Lemma 1** *In previous works, sybil-proofness is often achieved by the* $f_i^k \geq f_i^{k+1} + f_{i+1}^{k+1}$ *requirement. In our proposal the order of inclusion into E is irrelevant, on the contrary,—the amount of investment (*$\mathfrak{Tb}_i$ i.e. $\mathbb{A}_i^{\mathfrak{Tb}I}$) is relevant instead, thus intuitively, we employ sets and require $\sum_{i=1}^{|\mathbb{A}_M|} \mathbb{A}_i^{\mathfrak{Tb}I} \geq \sum_{i=1}^{|\mathbb{A}_M \cup \mathbb{A}_S|} \mathbb{A}_i^{\mathfrak{Tb}I} \Rightarrow \mathrm{F}_A(\mathbb{A}_M) \geq \mathrm{F}_A(\mathbb{A}_M \cup \mathbb{A}_S)$ (*Lemma 1*) for sybil-proofness; i.e. greater sum of investments into $\mathbb{A}_M$ alone results in higher expected rewards.

Speaking plainly, we begin by ascertaining two sets $\mathbb{A}_M$ and $\mathbb{A}_S$, with the second deemed as 'Sybil/malicious' and see how the game-mechanics cause the overall number of identities to tend towards 1, effectively rendering arity of the 2nd set $\mathbb{A}_S$ to become 0. i.e. $|\mathbb{A}_S|$ tends towards 1 as $|\mathbb{A}_M|$ tends towards 0 the process which is equivalent to elimination of Sybil nodes/identities.

**Theorem 1** *Now, let us prove that a given reward mechanism with a given reward function leads to highest rewards for* $\Lambda$ *as the arity of* $\Lambda_S \in \Lambda$ *containing Sybil nodes tends towards* 0 *and* $\mathbb{A}_M$ is driven down to become a singleton for any path length $L$ for any number of agents; i.e. $|\mathbb{A}_S| = 0 \Rightarrow$ results in $max(F(R^{\Lambda}_{max}))$—highest expected reward. i.e. $|\mathbb{A}_S| = 0 \Rightarrow\Rightarrow max(F(R^{\Lambda}_{max}))$.

**Lemma 2** *Higher investment leads to higher ratings. To prove this lemma it suffices to note that the function grows exponentially with* $\mathbb{A}_i^{\mathfrak{Tb}I}$ for a given path length $L$.

**Corollary 1** *Higher rating value leads to higher values of the reward function. We employ a Reward Function defined as*:

(Definition of Reward Function)$F_{\mathbb{A}}(\mathbb{A}_i^{\mathfrak{Tb}_I}) = \frac{\epsilon\left(\mathbb{A}_i^{\mathfrak{Tb}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathbb{A}_i^{\mathfrak{Tb}_I}\right)}; L, F \geq 1; \mathfrak{Tb}_I \geq 1, \mathfrak{Tb}_I, \mathbb{A}_i^{\mathfrak{Tb}_I} \in N$

$F_{\mathbb{A}_i}(\mathbb{A}_i^{\mathfrak{Tb}_I})$ denotes reward assigned to an $i$th identity in a data path, $F_{\mathbb{A}}(X)$, when X is a set, the function denotes cumulative a reward i.e. $\sum_{i=0}^{i=|X|-1} F_{\mathbb{A}_i}(\mathbb{A}_i)\ \forall \mathbb{A}_i \in X$. Here, it suffices to note that the reward function is a strictly increasing monotonic function, thus producing higher values for higher input values.

**$Epoch_i$:**

1. $\mathcal{L}_i$ releases an ordered set of blocks BS. $\forall B \in BS$ there is an endorsement $E_0^{B_i}$. We assume |BS| ≥ 1
2. For each $B_i$, $\mathcal{L}_i$specifies $R_N$ and $R_{\mathcal{L}_{i'}}$(either explicit or as a fraction of a puzzle-reward). Now, $\mathcal{L}_i$ might choose from two possibilities:
   - specify an explicit reward for $\mathcal{L}_{i'}$- an arbitrary amount of assets transferred from $\mathcal{L}_i$ to $\mathcal{L}_{i'}$ when extending upon $EB_i$.
   - specify an implicit reward for $\mathcal{L}_{i'}$ - reward as a fraction of the reward associated with $B_i$.

   *Note*: it is required for the decentralized consensus to assure rewarded amounts not to be spendable neither by $\mathcal{L}_i$ or $\mathcal{L}_{i+1}$ during the *safety period* $B_S$
3. $\mathcal{L}_i$ prepares data blocks and delivers through either $PA_1$ or $PA_2$ depending on whether leader is known or unknown.
4. Since block $B_i$had been sealed, once $\mathcal{L}_{i+1}$ receives information $M = B_i$; it can work only on a *sealed* representation of Bi and not directly on the block/puzzle itself. Recall that in sealed mode $\Phi'(B_i, E)$ needs to succeed on other nodes as well, thus $\mathcal{L}_{i+1}$ would not be able to separate $B_i$ from a legitimate sequence of intermediaries. That is under the assumption that for $B_{i+1}$to be recognized and thus for $\mathcal{L}_{i+1}$ to receive reward, $B_i$ needs to make it into the data-store as well.
5. $\mathcal{L}_{i+1}$ extends on each/selected $B_i$, producing a new block $B_{i+1}$. Note: $\mathcal{L}_i$ might had been allowed to produce multiple blocks. Notice that once $\mathcal{L}_{i+1}$ extends upon $B_i$ blocks indexed < *i* are assumed valid as well.
6. $\mathcal{L}_i$ continues producing blocks while monitoring the network for other leaders to extend upon the previously released blocks *(passive check)*.Once $\mathcal{L}_i$ detects another block extending upon one of its own, it proceeds to $\boldsymbol{Epoch_{i+1}}$.

**$Epoch_{i+1}$:**

7. Now, in accordance to either $PA_1$ or $PA_2$, $\mathcal{L}_i$ provides initial nodes $N_0$, to which PIDs of identities $\forall E \forall B_i \in BS$ extended upon correspond to, with 'unlocking data'
8. Now, similarly to $DR_1$, there are two options for intermediaries:

- BFTL - *Leader not known:* $UD_{N_i}^{out} = Sig_{PK_{N_i}}\left[UD_{N_{i-1}}^{IN}, PK_{N_{i+1}}, \Gamma_{N_{i+1}}\right]$

  (*note:* intermediaries would be able to look-up $\boldsymbol{N_i^{\mathcal{T}_I}}$values and choose favourable dealings)
- LFTB - *Leader known* $UD_{N_i}^{out} = Sig_{PK_{N_i}}\left[UD_{N_{i-1}}^{IN}, Enc_{PK_{\mathcal{L}}}\left(\left[PK_{N_{i+1}}, \Gamma_{N_{i+1}}\right]\right)\right]$.

  (***note****:* here routers would be unable to choose favourable dealings, due to encryption)

9. Members of each E get rewarded, now block reward for $\mathcal{L}_{i'}$ is made spendable after $B_S$ consecutive blocks.

**Fig. 14** Data blocks routing algorithm

### 7.2 Proof of sybil-proofness (global dominance)

1. (*Entity Ranking Function*) $\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}, L, \beta\right) = L^{\beta \mathcal{A}_i^{\mathfrak{Ъ}_I}}; \beta \geq 1, \mathfrak{Ъ}_I, \mathcal{A}_i^{\mathfrak{Ъ}_I} \in N, \beta \in \mathbb{R}, \mathrm{L}$—path length
2. $\Rightarrow$(*Lemma 2*) $\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I+1}\right) > \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)$ (higher investment $\mathfrak{Ъ}_I$ leads to higher rating values for any given identity)

Now, let ${}^{S}\mathbb{A}_X$ denote a set identities within a data path of length L.

1. (*Definition of Reward Function*)

$$\forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X, F_{\mathcal{A}}\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}, F\right) = \frac{\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)};$$

;

2. (*Definition 1*) $\Rightarrow \sum_{i=1}^{\left|{}^{S}\mathbb{A}_X\right|} \frac{\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)} = F \forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X$ (the sum of member-rewards equals F—the achievable reward)
3. *Thus*, $F_{\mathcal{A}}\left({}^{S}\mathbb{A}_X\right) = \sum_{i=1}^{\left|{}^{S}\mathbb{A}_X\right|} \frac{\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)} = F$ (total reward for members of $\sum_{i=1}^{\left|{}^{S}\mathbb{A}_X\right|} \frac{\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)} = F \forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X$
4. (*Using 2*) summing $\forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X, \sum_{i=1}^{L}\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) \leq \sum_{i=1}^{1} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I+1}\right)$ (data-path of length *L* filled entirely with Sybil nodes will yield lower rank than a path with just a single node of $\mathfrak{Ъ}_I$ greater by 1)
5. Thus, (*using 4 and 1*): $\sum_{i=1}^{L} F_{\mathcal{A}}\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) \leq \sum_{i=1}^{1} F_{\mathcal{A}}\left(\mathcal{A}_i^{\mathfrak{Ъ}_I+1}\right) \forall \mathbb{A}_i \in \Lambda \, \forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X$ (this translates to a higher expected total reward for any path length and any number of agents and identities)
6. Assuming the strategy of $\Lambda$ it to maximize *F* (*definition of* $\Lambda$), (*Using 5*), node will prefer to accumulate $R_{max}^{\Lambda}$ within a single identity.
7. Now notice that $\sum_{i=1}^{|\mathbb{A}_S|} \mathcal{A}_i^{\mathfrak{Ъ}_I} + \sum_{i=1}^{|\mathbb{A}_M|} \mathcal{A}_i^{\mathfrak{Ъ}_I} = R_{max}^{\Lambda}$, thus $\left(\sum_{i=1}^{|\mathbb{A}_S|} \mathcal{A}_i^{\mathfrak{Ъ}_I}\right) - 1 = \left(\sum_{i=1}^{|\mathbb{A}_M|} \mathcal{A}_i^{\mathfrak{Ъ}_I}\right) + 1$ (and for $\mathbb{A}$ to exist i.e. be part of any set, and receive reward it needs to have $\mathfrak{Ъ}_I \geq 1$ definition of $\mathbb{A}_S$)
8. Thus, (*using 2, 5 and 6*) $\sum_{i=1}^{|\mathbb{A}_M|} \mathcal{A}_i^{\mathfrak{Ъ}_I} \geq \sum_{i=1}^{|\mathbb{A}_M \cup \mathbb{A}_S|} \mathcal{A}_i^{\mathfrak{Ъ}_I} \Rightarrow \mathrm{F}_{\mathbb{A}}(\mathbb{A}_M) \geq \mathrm{F}_{\mathbb{A}}(\mathbb{A}_M \cup \mathbb{A}_S)$ for any L. **(proved Lemma 1)**
9. Thus, (*using Lemma 1, 5 and 8*) $max\left(F(R_{max}^{\Lambda})\right) \Leftrightarrow \lim_{\mathcal{A}_M^{\mathfrak{Ъ}_I} \to R_{max}^{\Lambda}} \left(\mathbb{A}_S^{\mathfrak{Ъ}_I}\right) = 0 \Longrightarrow$ and thus (*using definition of* $\sum_{i=1}^{\left|{}^{S}\mathbb{A}_X\right|} \frac{\epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right) * F}{\sum_{i=1}^{L} \epsilon\left(\mathcal{A}_i^{\mathfrak{Ъ}_I}\right)} = F \forall \mathbb{A}_i \in {}^{S}\mathbb{A}_X$)
$\lim_{\mathbb{A}_S^{\mathfrak{Ъ}_I} \to 0} (|\mathbb{A}_S|) = 0 \Longrightarrow \lim_{|\mathbb{A}_S| \to 0} (|\mathbb{A}_M|) = 1$ or $\lim_{|\mathbb{A}_S| \to 0} (|\mathbb{A}_S|) = 0$ (from a game-theoretic view)
10. $\Rightarrow |\mathbb{A}_S| = 0 \Longrightarrow max\left(F(R_{max})\right)$**(proved Theorem 1)**

## 8 Conclusions

In this work, while focusing on the problem of sybil-proof data exchange, we have revisited previous approaches, showcasing their shortcomings, and laid forward the first information exchange framework; with integrated routing and reward-function mechanics, provably secure in thwarting Sybil-nodes in 1-connected or eclipsed networks. The framework neither requires nor assumes any kind of constraints regarding the network's topology (i.e. network is modelled as a random-connected graph). The proposal, while being storage and transmission efficient, is suitable for rewarding not only consensus-related datagrams (both data-blocks and transactions) but consensus-extrinsic information as well, thus facilitating an universal sybil-proof data exchange apparatus, provably valid under the assumption of existence of a data store whose property of non-malleability emerges as time approaches infinity. We conducted our research under two scenarios—with round leader known and unknown in advance of each transactional round.

## 9 Future work

As of this writing, during revision of this manuscript, the here-in described mechanics have been already implemented and made operational on the test-net of our large research project (GRIDNET OS, available at https://gridnet.org), which is doing more than fine, soon to-be-made open-source. We have further research already pending, allowing for rewarding of multiple peers simultaneously from a single data-structure registered on a Decentralized State Machine through the concept of the to-be-introduced *Multi-Dimensional Token Pools* Currently these are employed as a show-off functionality in the Single/Multi-player Snake game available on GRIDNET OS (for efficiently rewarding a large number of a-priori unknown

peers from an on-the-chain registered relatively small data-structure once the player collects in-game-coins, without imposing excessive burden onto the decentralized state-machine). The crypto-mechanics have also been implemented in JavaScript and thus may be operational across a variety of protocols such as WebRTC for incentivizing data propagation across web-browsers also aiding the well-known problematics for incentivizing WebRTC TURN servers. Currently, we are working on incorporating the mechanics into file and audio/video transmissions happening over onion-routed data transmissions across full-nodes and web-browsers. As of this writing, the Web-part of GRIDNET OS is pending public deployment. It might be interesting to note that large portions of the project have been implemented on YouTube live since the early 2017.



## References

1. Skowroński, R.: The open blockchain-aided multi-agent symbiotic cyber–physical systems. Futur. Gener. Comput. Syst. **94**, 430–443 (2019)
2. Skowronski, R.: On the applicability of the GRIDNET protocol to Smart Grid environments. In: 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm). IEEE, 2017
3. Abraham, I., et al.: Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus. CoRR, abs/1612.02916 (2016).
4. Skowronski, R.: Fully distributed GRIDNET protocol, with no trusted authorities. In: 2017 International Conference on Information Networking (ICOIN). IEEE, 2017
5. Ersoy, O., et al.: Information propagation on permissionless blockchains. arXiv preprint arXiv 1712 (2017)
6. Bag, S., Ruj, S., Sakurai, K.: Bitcoin block withholding attack: analysis and mitigation. IEEE Trans. Inf. Forensics Secur. **12**(8), 1967–1978 (2016)
7. Smith, A.: 1723–1790. The Wealth of Nations/Adam Smith; Introduction by Robert Reich; Edited, with Notes, Marginal Summary, and Enlarged Index by Edwin Cannan. New York: Modern Library, 2000.
8. Matthews, R., Brauer, P., Dunne, J.: Defense Offsets: Policy Versus Pragmatism. Routledge, London (2004)
9. Connor, R.C.: The benefits of mutualism: a conceptual framework. Biol. Rev. **70**(3), 427–457 (1995)
10. Archetti, M., et al.: Economic game theory for mutualism and cooperation. Ecol. Lett. **14**(12), 1300–1312 (2011)
11. Boucher, D.H. (ed.): The Biology of Mutualism: Ecology and Evolution. Oxford University Press, New York (1985)
12. Bronstein, J.L.: Our current understanding of mutualism. Q. Rev. Biol. **69**(1), 31–51 (1994)
13. Babaioff, M., et al.: On bitcoin and red balloons. In: Proceedings of the 13th ACM Conference on Electronic Commerce (2012)
14. Otte, P., de Vos, M., Pouwelse, J.: TrustChain: a sybil-resistant scalable blockchain. Future Gener. Comput. Syst. **107**, 770–780 (2020)
15. Seuken, S, Parkes, D.C.: Sybil-proof accounting mechanisms with transitive trust. In: Proceedings of the International Foundation for Autonomous Agents and Multiagent Systems (2014).
16. Gong, S., Lee, C.: Blocis: blockchain-based cyber threat intelligence sharing framework for sybil-resistance. Electronics **9**(3), 521 (2020)
17. Nakamoto, S.: "Bitcoin: A peer-to-peer electronic cash system." Decentralized Business Review, 21260 (2008)
18. Ersoy, O., et al.: Transaction propagation on permissionless blockchains: incentive and routing mechanisms. In: 2018 Crypto Valley Conference on Blockchain Technology (CVCBT). IEEE, 2018
19. Ersoy, O., Zekeriya, E., Lagendijk, R.L.: TULIP: a fully incentive compatible blockchain framework amortizing redundant communication. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). IEEE, 2019.
20. Sauer, B.: Virtual currencies, the money market, and monetary policy. Int. Adv. Econ. Res. **22**(2), 117–130 (2016)
21. Goldschlag, D.M., Stubblebine, S.G.: Publicly verifiable lotteries: applications of delaying functions. In: International Conference on Financial Cryptography. Springer, Berlin, 1998.
22. Liu, D., Camp, L.J.: Proof of Work can Work. In: WEIS. 2006.
23. Likelihood of winning and overbidding in first price auctions, Paolo Crosetto, Antonio Filippin, Peter Katuscak
24. McFadden, D.: The theory of first-price, sealed-bid auctions. Institute of Business and Economic Research, Economics at the University of California: Berkeley. Available in: http://eml.berkeley.edu/~mcfadden/eC103_f03/auctionlect.pdf. Accessed 7 Jan 2016 (2003). Auction Theory, Jonathan Levin
25. Predicting the probability of winning sealed bid auctions: the effects of outliers on bidding models, Skitmore, Martin



**Rafał Skowroński** is a researcher at the Poznan University of Technology, Poland where he is the head of the GRIDNET OS research project. He specializes in applied cryptography, network security, reverse engineering, blockchain technology, power industry (Smart Grid technology) and secure programming. Speaker at multiple conferences regarding blockchain technology: the IEEE SmartGridComm conference (Germany) and ICOIN (Vietnam 2018), BlockchainTechCongress and EuroPower (2018). Creator of the first entirely decentralized architecture for Smart Grids and creator of the first communication protocol which rewards intermediaries on a per-byte basis using blockchain-based technology. In 2019 authored research on the Open Blockchain-Aided Multi-Agent

Symbiotic Cyber-Physical Systems published by Future Generation Computer Systems.

**Jerzy Brzeziński** received M.Sc. degree in electrical engineering, and Ph.D. and Dr. Habil. degrees in computer science, all from Poznań University of Technology, in 1977, 1982, and 1989, respectively. Since April 1977, he has been with Poznan University of Technology where he is currently a Full Professor of Computer Science and head of the Laboratory of Computing Systems. His research interests include in general dependable distributed computing systems, and in particular: data-centric and client-centric consistency models of data replication, replication techniques and protocols, distributed checkpointing and rollback recovery, distributed deadlock and termination detection, reliable group communication in MANETs (mobile ad-hoc networks), deadlock prevention and avoidance in store-and-forward networks, distributed algorithms design and complexity analysis. He is the author and coauthor of two books, and over 130 research papers published, among others, in IEEE Transactions on Communications, IEEE Transactions on Computers, Journal of Parallel and Distributed Computing, European Journal of Operational Research, Annales des Télécommunications, Microprocessing and Microprogramming, Bulletin of the Polish Academy of Sciences, and proceeding of many international conferences. He has been involved in many international and national research projects: New information technologies for electronic economy and information society based on SOA paradigm, Eskulap—Hospital Information System, Wielkopolska Telemedicine Center—improvement of access to specialistic medical services and of treatment quality in Wielkopolska, Network management and diagnostics systems, among others. Currently he is teaching Operating Systems, Distributed Computing, and Computer Systems Architecture. Prof. Brzeziński is a member of the IEEE CS, ACM, Polish Information Processing Society, Computer Science Committee of the Polish Academy of Sciences, among other.