Resource Burning for Permissionless Systems (Invited Paper)

Diksha Gupta[1(✉)], Jared Saia[1], and Maxwell Young[2]

[1] Department of Computer Science, University of New Mexico,
Albuquerque, NM, USA
{dgupta,saia}@cs.unm.edu

[2] Department of Computer Science and Engineering, Mississippi State University,
Mississippi State, MS, USA
myoung@cse.msstate.edu

Abstract. Proof-of-work puzzles and CAPTCHAS consume enormous amounts of energy and time. These techniques are examples of resource burning: verifiable consumption of resources solely to convey information.

Can these costs be eliminated? It seems unlikely since resource burning shares similarities with "money burning" and "costly signaling", which are foundational to game theory, biology, and economics. Can these costs be reduced? Yes, research shows we can significantly lower the asymptotic costs of resource burning in many different settings.

In this paper, we survey the literature on resource burning; take positions based on predictions of how the tool is likely to evolve; and propose several open problems targeted at the theoretical distributed-computing research community.

> *"It's not about money, it's about sending a message."*
> The Joker [107]

1 Introduction

In 1993, Dwork and Naor proposed using computational puzzles to combat spam email [43]. In the ensuing three decades, ***resource burning***—verifiable consumption of resources—has become a well-established tool in distributed security. The resource consumed has broadened to include not just computational power, but also communication capacity, computer memory, and human effort.

The rise of ***permissionless systems*** has coincided with the recent increase in popularity of resource burning. In permissionless systems, any participant—represented by a virtual identifier (***ID***) in the system—is free to join and depart without scrutiny, while enjoying a high degree of anonymity. For example, an ID might be an IP address, a digital wallet, or a username.

In this setting, security challenges arise from the inability to link an ID to the corresponding user. A single malicious user may create a large number of

This work is supported by the National Science Foundation grants NSF-CNS-1816250 and NSF-CNS-1816076.

A. W. Richa and C. Scheideler (Eds.): SIROCCO 2020, LNCS 12156, pp. 19–44, 2020.
https://doi.org/10.1007/978-3-030-54921-3_2

accounts on a social media platform to wield greater influence; or present itself as multiple clients to disproportionately consume resources provided by the system; or inject many IDs in a peer-to-peer network to gain control over routing and content. This malicious behavior is referred to as the ***Sybil attack***, originally described by Douceur [41].

Such attacks are possible because users are not "ID-bounded" in a permissionless system; that is, there is no cost, and therefore no limit, to the number of IDs that the attacker (***adversary***) can generate. However, the adversary is often "resource-bounded", even if this bound is unknown. In particular, it may be constrained, for example, in the number of machines it controls, or total channel capacity to which it has access. Resource burning leverages this constraint, forcing IDs to prove their distinct provenance by producing work that no single attacker can perform.

Paper Overview. Resource burning is a critical tool for defending permissionless systems. In support of this claim, we survey an assortment of topics: distributed ledgers, application-layer distributed denial-of-service (DDoS) attacks, review spam, and secure distributed hash tables (DHTs). Using these examples, we highlight how results in these different areas have converged upon resource burning as a critical ingredient for achieving security; this is summarized in Table 1.

Table 1. Summary of the domains surveyed, along with the corresponding resources, and core functionality that is secured by resource burning. We also make conjectures on the algorithmic spend rate. Here, T is the adversary's spend rate; J_G is the join rate for good IDs; and P_G is the posting rate of good IDs. We elaborate on these notions in Sect. 2.5. The $\tilde{O}$ notation omits polylogarithmic factors.

Domain	Primary resource consumed	Mechanism	Enabled functionality	Conjectured cost
Blockchains	CPU	CPU puzzles	Distributed ledger	$O(\sqrt{TJ_G} + J_G)$
DHTs	CPU	CPU puzzles	Decentralized storage and search	$\tilde{O}(\sqrt{TJ_G} + J_G)$
DDoS attacks	Bandwidth/CPU	Messages/CPU puzzles	Fair allocation of server resources	No conjecture
Review spam	Human time	CAPTCHAS	Trusted consumer recommendations	$\tilde{O}(T^{2/3} + P_G)$

As prelude to this survey, we predict how resource-burning may evolve, and how systems may adapt to this technique. These predictions are distilled in four position statements below.

> **Position 1.** *Resource burning is a fundamental tool for defending permissionless systems.*

PoW and CAPTCHAs have been around now for decades, persisting despite concerns over scalability, resource consumption, security guarantees, and predicted obsolescence (see discussion under Position 2 and Sect. 3). The continued practical success of resource burning aligns with theoretical justification from game-theoretic results on "money-burning" and "costly signaling" (Sect. 2.1). Given the increasing popularity of permissionless systems, and the need to defend them, resource burning will likely only increase in prevalence.

Position 2. *Resource burning must be optimized.*

In May 2020, the annual energy consumption of Bitcoin was 57.92 terawatt-hours of electricity per year, which is comparable to the annual electricity consumption of Bangladesh; Ethereum was 7.9 terawatt-hours, comparable to that of Angola [39]. In 2012, humans spent an estimated 150, 000 hours per day solving CAPTCHAS [114,137]. The rise of permissionless systems will likely only increase these rates of resource burning.

On the positive side, recent theoretical results suggest that resource burning can be analyzed and optimized just like any other computational resource [59,61]. But there is significant work needed to: (1) develop a theory of resource burning focused on distributed security; and to (2) translate this theory into practical resource savings.

In this paper, we discuss current theoretical work on reducing resource-burning rates across multiple application: blockchains (Sect. 3); DHTs (Sect. 4); application-layer DDoS attacks (Sect. 5); and review spam (Sect. 6).

Position 3. *Reducing from permissionless to permissioned systems is important.*

Four decades of research have resulted in efficient and reliable algorithms for permissioned networks. We should leverage these results when addressing problems in new permissionless systems. One way to do this is to develop tools, based on resource burning, that bound the fraction of IDs controlled by the adversary (***bad IDs***) in permissionless systems. In Sect. 3, we discuss results on the problem GenID, which provides this bound for static, permissionless networks; and DefID which does so for permissionless networks with churn.

In Sects. 5 and 6, we discuss the threats posed by application-layer denial-of-service (DDoS) attacks and review spam. Neither problem aligns perfectly with a permissionless model. For example, servers are under administrative control, and online review systems often require credentials for account creation. However, these systems still remain vulnerable to malicious participants that are difficult to identify, and who monopolize system resources. We define a *hybrid* system model as one that contains both permissioned and permissionless properties. We note that any tools designed for permissionless systems will also work for hybrid systems. However, we would expect to be able to develop more efficient techniques to adapt tools from permissioned systems to these hybrid systems.

Position 4. *Theoretical guarantees should hold independently of the resource burned. Research should focus on both domain-specific and domain-generic problems.*

As theoreticians, we should generalize as much as possible. Algorithms that use resource burning should require a certain "cost" that specifies the amount of the resource to be consumed, but should allow for that resource to be anything: computation, computer memory, bandwidth, human effort, or some other resource yet to be defined. As much as possible, theoretical results should be stated in terms of this cost, irrespective of the resource consumed. This ensures our theoretical results will continue to be relevant, even as underlying technologies providing verifiable resource burning may change.

Additionally, a key research focus should be on problems that generalize across multiple domains. In this paper, we describe two examples: GenID and DefID (Sect. 3.1). Our remaining three examples are domain-specific. We believe it is important to work on both types of problems.

2 Background and Preliminaries

Resource burning has found application in various areas of computer security; indeed, its use was proposed by Douceur [41] as a defense against the Sybil attack [40,75,100,106]. However, resource burning has a broader history, with similar ideas appearing in several other scientific domains.

In Sect. 2.1, we present this background. In Sects. 2.2, 2.3, and 2.4, we elaborate on the notion of resource burning. Finally, in Sect. 2.5, we describe a general problem model that provides a unifying set of assumptions and terminology used throughout this document.

2.1 Game Theory, Biology and Economics

Resource burning is analogous to what is referred to as *money burning* in the game theory literature. To the best of our knowledge, the first significant algorithmic game theory study of money burning, due to Hartline and Roughgarden, analyzed the use of money burning in mechanism design [61]. Their main result is a near-optimal mechanism for multi-unit auctions, where the quantity optimized is ***social welfare*** or the sum of utilities of all players. They also give results showing that, under certain conditions, an auction utilizing money burning can obtain a $\Omega\left(\frac{1}{1+\log(n/k)}\right)$ fraction of the optimal social welfare, where the auction consists of n bidders who are bidding for k units. They conclude that *"the cost of implementing money-burning ... is relatively modest, provided an optimal money-burning mechanism is used"*.

Money burning is also known as *costly signaling* in the game theory literature, and it has two main uses in this context. First, it can signal commitment to a certain action, as is illustrated in the "lunch" game[1] [29,68]. Second, it can signal

[1] This is equivalent to what is referred to as the "battle of the sexes" game in [29].

the "type" of a player, as is in the"college" game [29]. We present these two games below.

Lunch Game. Two friends want to eat lunch together, but the first friend prefers option A and the second prefers option B. They each obtain payoff of -1 if they choose different locations. If they both pick option A, they obtain payoffs of 10 and 1 respectively. Conversely, if they both pick option B, they obtain payoffs of 1 and 10.

Now, if the first friend verifiably burns money equal to 1 unit of utility prior to playing the game, this signals a commitment to their preferred option, since if they were to choose the unpreferable option, their utility would now be at most 0. Thus, they would not have played the game. In this way, a friend who burns money can expect higher utility.

College Game. Each student is one of two types: *smart* or *daft*. Each student is considering college and can choose either the action *attend* or *not attend*. A smart student pays a cost of 1 (in terms of time and effort) to attend college, and a daft student pays a cost of 3 to attend college. We assume that the decision of the student to attend college is publicly known, but that otherwise, college has no impact: daft students stay daft even after attending.[2]

An employer wants to hire smart students. If the employer hires a smart student, their benefit is 2, and if they hire a daft student, their cost is 2. If a student is hired by the employer, they have a benefit of 2, and if they are not hired, they have a benefit of 0.

It is easy to verify that the following is a Nash equilibrium for this game:

- ○ Smart students attend college.
- ○ Daft students do not attend college.
- ○ The employer hires only students that attend college.

Here, smart students all choose to attend, *even though college has no intrinsic benefit*. Thus, the choice to attend college is a costly signal made by the smart students, and college itself is an example of resource burning.

If the option to attend college were removed from the game, and the fraction of smart students were less than 1/2, then a Nash equilibrium would be for the employer to never hire. In this case, the overall social welfare—the sum of expected benefits to all players—would decrease.

Biology. Costly signaling is a well-known phenomena in biology. A relevant example from animal behavior is *stotting*, in which quadrupeds, such as deer and gazelles, repeatedly jump high into the air. This is often done in view of a predator, suggesting that stotting is a costly signal to the predator that the prey is too healthy to catch [49]. Other examples occur in sexual-selection, where the use of plumage, large antlers, and loud cries are a costly signal of fitness [156].

Economics. In 1912, the economist Thostein Veblen coined the term "conspicuous consumption" to describe costly signaling used by people to advertise both

[2] On the positive side, smart students stay smart!.

wealth and leisure. For example, Veblen writes, *"The walking stick serves the purpose of an advertisement that the bearer's hands are employed otherwise than in useful effort, and it therefore has utility as an evidence of leisure"* [134]. Decades of economic studies suggest that conspicuous consumption is a critical part of historical and modern economies [98,113,118,119,131]. For example, Sundie et al write: *"Although showy spending is often perceived as wasteful, frivolous, and even narcissistic, an evolutionary perspective suggests that blatant displays of resources may serve an important function, namely, as a communication strategy designed to gain reproductive reward"*[131].

2.2 What is Resource Burning?

We define ***resource burning*** as the verifiable consumption of a resource. In particular, it is computationally easy to verify both the consumption of the resource, and also the ID that consumed the resource [6]. Below we describe several resource-burning techniques.

Proof-of-Work (PoW). PoW is arguably the current, best-known example of resource burning. Here, the resource is computational power. Proof-of-work has been proposed for spam-prevention [43,85,90]; blockchains [103]; and defense against Sybil attacks [10,88].

CAPTCHAs. A *completely automated public Turing test to tell computers and humans apart*, or a CAPTCHA, is a resource-burning tool where the resource is human effort [147]. CAPTCHAs may be based around text, images, or audio; however, several design and usability issues exist [148].

Proof-of-Space. Proof-of-space requires a prover to demonstrate utilization of a certain amount of storage space [1,13,42,44]. This approach is foundational for Spacemint cryptocurrency [111]. Like PoW, proof of space demonstrates the consumption of a certain amount of a physical resource, but can require less electrical power. A related proposal is "Proof of Space-Time" [102], which demands proof of consumption of a certain amount of storage space for a certain amount of time.

Resource Testing. Resource testing requires a prover to demonstrate utilization of a radio channel [55,56,101].[3] Consider a wireless setting where each device has a single radio that provides access to one of several channels. Thus, an adversary representing two bad IDs, but with a single device, can only listen to one channel at a time. A base station can assign each ID to separate channels; send a random message on one of these channels chosen randomly; and demand that the message be echoed back by the corresponding ID. Since the adversary can only listen to a one channel at a time, it will fail this test with probability at least 1/2.

[3] Resource burning refers to the game-theoretic money burning technique; resource testing refers to that technique specifically applied in the wireless domain.

2.3 What is *not* Resource Burning

Proof-of-Stake (PoS) is a defense for permissionless systems, wherein security relies on the adversary holding a minority stake in an abstract finite resource [2]. It has been proposed primarily for cryptocurrency systems (Sect. 3). When making a group decision, PoS ensures that each ID has voting weight proportional to the amount of cryptocurrency that ID holds. Well-known examples of such systems are ALGORAND [54], which employs PoS to form a committee, and Ouroboros [83], which elects leaders with probability proportional to their stake. Hybrid approaches using both PoW and PoS exist, including one proposed for the Ethereum system [8], and under the name "Proof of Activity" [27]. In contrast to the above examples, PoS involves a measurement, rather than a consumption of, a resource.

Disadvantages of Proof-of-Stake. Unfortunately, PoS can only be used in systems where the "stake" of each ID is globally known. Thus, it seems likely to remain relevant primarily in the domain of cryptocurrencies. Moreover, even within that community, there are concerns about proof-of-stake. To quote researcher Dahlia Malkhi: *"I think proof-of-stake is fundamentally vulnerable ... In my opinion, it's giving power to people who have lots of money"* [35].

2.4 Resource Burning Does Not Require *Waste* of the Resource

While resource burning requires verifiable *consumption* of a resource, it does not necessarily require *waste* of that resource. For example, Von Ahn et al. [137] developed the reCAPTCHA system which channeled human effort from solving CAPTCHAs into the problem of deciphering scanned words that could not be recognized by computer. Their system achieved an accuracy exceeding professional human transcribers, and was responsible for sucesssfully transcribing hundreds of millions of words from public domain books.

In 2018, Ball et al. developed proof-of-work puzzles whose hardness is based on worst-case assumptions [25]. These puzzles are based on the Orthogonal Vectors, 3SUM, and All-Pairs Shortest Path problems, and any problem that reduces to these problems, including deciding any graph property statable in first-order logic. Hence, their work enables design of PoW puzzles that can be useful for solving computational problems of practical importance.

In [126], Shoker developed proof-of-work puzzles that solve real-world matrix-based scientific computation problems. He named this technique "Proof of Exercise".

All algorithms discussed in this paper are compatible with this type of "useful" resource burning, where the consumption of the resource solves practical problems. Our only requirement of the resource burning mechanism is that the consumption of the resource be easily verifiable, which holds true for the above results.

2.5 A General Model

We discuss broad aspects of a general model for permissionless systems. This allows us to highlight commonalities between different application domains, while retaining the same terminology throughout.

The system consists of virtual ***identifiers (IDs)***. An ID is ***good*** if it obeys protocol and belongs to a unique user; otherwise, the ID is ***bad***. Good and bad IDs cannot necessarily be distinguished *a priori*.

Communication. Communication is synchronous and occurs either via point-to-point or via a broadcast primitive. The former is typical for peer-to-peer systems and the general client-server setting. The latter corresponds to permissionless blockchains, where it is a standard assumption that a good ID may send a value to all other good IDs within a known and bounded amount of time, despite an adversary; for examples, see [30,52,54,92] and see [97] for empirical justification.

Adversary. A single adversary controls all bad IDs; this pessimistically represents perfect collusion and coordination by malicious users. Bad IDs may arbitrarily deviate from our protocol, including sending incorrect or spurious messages. The adversary can send messages to any ID at will, and can view any communications sent by good IDs before sending its own. It knows when good IDs join and depart, but it does not know in advance the private random bits generated by any good ID.

Often, the adversary is assumed to control only an α-fraction of the network resources, for $\alpha > 0$. Generally, in settings where correctness is threatened, α must be a small constant; for example, often bounded below 1/3 or 1/4. Alternatively, there are settings where α can be any constant bounded away from 1; typically, this corresponds to problems of performance (rather than correctness).

Tunable Costs. We measure ***cost*** as the amount of resource consumed. Our model is agnostic with respect to the particular resource used. However, we assume that it is possible to arbitrarily tune the cost. In particular, we assume that, for any value x, an ID can be issued a ***challenge*** of difficulty x that will require consumption of x units of whatever resource is used.

Resources such as computation, computer memory, and bandwidth have inherently tunable costs. For CAPTCHAs, cost could be adjusted in two possible ways. First, by adjusting the difficulty of the puzzle, by either (1) adjusting the number of alphanumeric digits or the number of images to be classified; or (2) adjusting the difficulty of an individual recognition task as described in the ScatterType CAPTCHA system [24]. Second, by adjusting the *expected* difficulty by adjusting a probability of being required to solve a CAPTCHA.

Joins and Departures. Often, the system is dynamic, with IDs joining or departing over time. There is no *a priori* method for determining whether a joining ID is good or bad. Joins and departures by bad IDs may be scheduled in a worst-case fashion, and pessimistically we often assume the adversary also has a limited ability to schedule these events for good IDs. We will generally assume a lower bound on the number of IDs in the system, and that the lifetime of the system is polynomial in this lower bound.

Key Notation. Through out this work, let $\boldsymbol{T}$ denote the ***adversarial spending rate***, which is the cost to the adversary over the system lifetime divided by the lifetime of the system. Let the ***algorithmic spending rate***, $\boldsymbol{A}$, be the cost to all good IDs over the system lifetime divided by the lifetime of the system.

In the blockchain and DHT problems, we let $\boldsymbol{J_G}$ denote the ***good ID join rate***, which is the number of good IDs that join during the system lifetime divided by the lifetime of the system. Finally, for the review spam problem, we let $\boldsymbol{P_G}$ denote the ***good posting rate***, which is the number of posts made by good IDs during the system lifetime divided by lifetime of the system.

2.6 Game Theoretic Analysis

For many of our problems, we can analyze the defense of a system as a two-player zero sum game [45] as follows. There is an adversary that can choose to attack or not, and an algorithm that can choose to defend or not. There is a system invariant, which the algorithm seeks to protect, that has some value V. There is a function f that gives the cost incurred when the algorithm chooses to defend as follows: if the adversary spends T to attack, then the algorithm will spend $f(T)$ to defend. Thus the payoff matrix for the algorithm is given below.

		Adversary	
		Attack	¬Attack
Algorithm	Defend	$T - f(T)$	$-f(0)$
	¬Defend	$-V$	0

Solving this game, we get that in the Nash equilibrium, the algorithm player will defend with probability $p = \frac{V}{T - f(T) + f(0) + V}$. Thus, the expected utility of the game to the algorithm player will be $\frac{-Vf(0)}{T - f(T) + f(0) + V}$. In many of our problems, $f(T) = f(0) + o(T)$, and so we obtain a value that is $\Theta\left(\frac{-Vf(0)}{T+V}\right)$. Smaller T optimizes the utility for the adversary, in which case, the expected utility of the algorithm is $\Theta(-f(0))$.

3 Blockchains and Cryptocurrencies

A *blockchain* is a distributed ledger. In particular, it is a distributed data structure that stores transactions between IDs in a network. Each transaction represents flow of a resource from one ID to another. Every transaction added must be legitimate, in the sense that the source ID owns the resource to be transferred, as indicated by the distributed ledger, at the time of the transaction. Importantly, transactions can only be added to the blockchain, and once added, can never be deleted or edited.

3.1 GenID and DefID

Perhaps the current, most frequently-used application of resource-burning is for blockchains. Permissionless blockchains are vulnerable to Sybil attacks [89]. The next two problems use resource burning to defend against this. Recall that the adversary controls an α-fraction of the resource that is being burned.

The GenID Problem. The problem stated below, GenID, was first defined and studied by Aspnes, Jackson, and Krishnamurthy [11]. They proposed a solution with latency of 3 rounds, and $\tilde{O}(n^2)$ bits sent per good ID, at a burned resource cost of $O(1)$ per good ID.

Problem 1. **GenID**

Model: Initial set of IDs; n of which are good, with the rest are controlled by an adversary.
Goal: All good IDs decide on a set of IDs S such that: (1) all good IDs are in S; and (2) at most a $O(\alpha)$ fraction of the IDs in S are adversarial.

Several other solutions to GenID have been proposed in the literature [4,10,67,81]. Andrychowicz and Dziembowski described an algorithm with a latency of $\Theta(n)$ rounds; $\tilde{O}(n^2)$ bits sent per good ID; and a burned resource cost of $\tilde{O}(1)$ per good ID [10]. Concurrent to this work, Katz, Miller and Shi [81] proposed another solution with similar costs. Hao et al. [67] improved on these results via using a randomized leader election protocol. Their algorithm has, in expectation, a latency of $\Theta\left(\frac{\ln n}{\ln\ln n}\right)$ rounds; $\tilde{O}(n)$ bits sent per good ID; and a burned resource cost of $\Theta\left(\frac{\ln n}{\ln\ln n}\right)$ per good ID.

The most recent work in this domain is by Aggarwal et al. [4], which requires in expectation: $O(1)$ latency; $O(n)$ bits sent per good ID; and a burned resource cost of $O(1)$ per good ID.

It is still not known if these costs can be reduced for the general problem, or for an "almost-everywhere" versions of the problem, where all but a $o(1)$ fraction of the IDs must learn S. To the best of our knowledge, there are no current lower-bounds on the problem.

The DefID Problem. The following problem, called DefID, considers the GenID problem in the presence of churn.

Problem 2. **DefID**

Model: Stream of IDs joining and leaving a network.
Goal: At most an $O(\alpha)$-fraction of bad IDs in the network at any time.

A first algorithm to solve DefID was proposed in by Gupta, Saia and Young in [58]. It required algorithmic spend rate of $O(J_G + T)$; recall that J_G is the join rate of good IDs per time step, and T is the spend rate of the adversary. Note that this result holds without any additional assumptions. Gupta, Saia and Young further improved this result in [59,60] to $O(J_G + \sqrt{TJ_G})$, subject to two assumptions on the join rate of good IDs, which are found to be supported by real-world data [59].

Specifically, the assumptions needed are as follows. Define an ***epoch*** to be the length of time it takes for the fraction of good IDs to change by 3/4 fraction. First, the join rate for good IDs changes by at most a multiplicative factor between any two successive epochs. Second, in any epoch the actual join rate for good IDs over any "sufficiently large" period of time is within constant factors of the join rate for good IDs over the entire epoch.

An asymptotically matching lower bound was obtained for a large class of algorithms [59]. An open problem is to generalize this bound to all algorithms.

4 Distributed Hash Tables

Distributed hash tables (DHTs) are a popular P2P distributed data structure [3,80,87,96,117,129] with several implementations over the years [46,128,141]. Generally, the design entails hashing attributes of a user's machine to a key value (or ID) in a virtual space; similarly, for data items. The various DHT constructions differ in their overlay topologies, but typically IDs need only maintain state on a small number of neighbors, and routing is possible with a small number of messages, where small means at most logarithmic in the number of IDs in the system.

These systems are vulnerable to attack. A bad ID that participates in routing can drop or corrupt any message it receives. A good ID can be completely isolated from the rest of the network if all of its neighbors are bad; this is often referred to as an ***eclipse attack*** [63,127]. Finally, content can be compromised if bad IDs alone are responsible for storing a particular data item. Generally, the behavior of bad IDs is modeled by Byzantine faults. For almost two decades, there has been a sustained interest in the design of secure DHTs that can tolerate such attacks [135].

Byzantine Fault Tolerance in DHTs. A popular approach to tolerating bad IDs depends makes use of ***groups***: these are small sets of IDs, each of which have a good majority. Intuitively, a group is used in place of an individual peer, and the group members act by using majority action or secure multiparty computation to coordinate actions. For example, routing can be performed robustly via all-to-all communication between each pair of groups along the path from source to destination. Examples of group-based DHT constructions include [21–23,48,72, 105,122,125,151].

As an alternative to using groups, bad IDs may be tolerated by employing some form of redundant routing [32,65,74,78,82,104]. Several other results do not explicitly apply to DHTs, although they may be compatible. For example, the challenge of tolerating bad IDs is exacerbated in highly-dynamic P2P systems, and there is a growing body of work in this area [14–18,57]. Self-healing networks are another approach for achieving security, where bad IDs are identified and evicted [84,120,121].

In all of these works, a critical assumption is that the fraction of bad IDs is a small constant. However, given that DHTs are often permissionless, this assumption is easily violated via a Sybil attack. Thus, while many tools have been developed for securing DHTs against Byzantine faults, additional work is required to limit the fraction of bad IDs in the permissionless setting.

Sybil Resistance. Several approaches have been proposed for mitigating the Sybil attack. The influence of bad IDs can be limited via containment schemes that leverage the network topology in structured overlays [124] and in social networks [7,86,99,143,152–154]. However, the information required—particularly social networks—may not always be available.

An alternative defense is to use measurements of communication latency or wireless signal strength to verify the uniqueness of IDs [26,38,53,91,140]. However, these techniques are sensitive to measurement accuracy.

For DHTs, an early result by Danezis et al. [37] gives a heuristic to limit the impact of bad IDs using bootstrapping information, but unfortunately provides no formal guarantees. Results that employ resource burning are scarce. The use of computational puzzles in decentralized systems is explored by Borisov [31] and Tegeler and Fu [132] as a means for identifying and excluding bad IDs from the system. Computational puzzles are also used by Rowaihy et al. [116] to throttle the rate of bad IDs added to a structured P2P system; however, this does not limit their number. Arguably the best-known result is the SybilControl scheme by Li et al. [88], which provides for a DHT construction that limits the number of bad IDs through the use of computational puzzles. Good IDs periodically challenge their neighbors under the Chord DHT topology [129,130], and blacklist those who do not respond with a solution in time. Experimental results indicate that this approach, in conjunction with limited data replication, allows for almost all searches to succeed.

4.1 Why DefID is Not Enough

The DefID problem (Sect. 3.1) captures many of the challenges required for secure DHTs. However, current solutions to DefID depend heavily on a means to coordinate resource burning. The main approach is to use a committee—a small set of IDs with a good majority—which issue resource-burning challenges. To apply results on DefID to DHTs requires decentralizing the functionality provided by the committee.

Additionally, while DefID always guarantees a minority of bad IDs, this is not enough. In particular, to ensure reliable routing and protection from eclipse attacks, group-based approaches demand that all groups have a minority of bad IDs. Fortunately, there are already clever techniques to spread the bad IDs uniformly across the groups. Informally, when a new ID joins a group, some IDs in the group are evicted and resettled in random locations, and their replacements are selected uniformly at random [21–23,57].

Unfortunately, performing such shuffling for every joining ID, even when there are no bad IDs in the system, incurs large bandwidth costs. A major open problem is to devise an algorithm that minimizes both bandwidth and resource-burning costs, as a function of adversarial spend rate.

4.2 The Permissionless DHT Problem

Problem 3 gives our formal problem in this domain. It assumes that the adversary controls an $\alpha < 1/3$ fraction of the burnable resource. We now describe some ideas about how to solve it.

Problem 3. **A Secure DHT in the Permissionless Setting**

Model: The adversary has complete control over the scheduling of joins and departures for bad IDs and limited control for good IDs. There is no explicit assumption that the good IDs are in the majority at all times.
Goal: A DHT that enables secure and efficient routing between any two good IDs in the system.

Recall from Sect. 3.1 that DefID imposes a cost of $O(J_G + \sqrt{TJ_G})$ on the good IDs. Informally, a plausible extension to this result is for each group in the DHT to act as a committee that runs an algorithm to solve DefID. In many group-based constructions, a good ID belongs to a number of groups that is logarithmic in the system size. Consequently, the algorithmic spend rate is likely to increase by a logarithmic factor. This yields our conjectured bound of $\tilde{O}(\sqrt{TJ_G} + J_G)$. Note that this aligns with Position 2 since costs to the good IDs are low when the adversary expends little effort (or does not attack at all), and grows slowly relative to the adversary's cost when a significant attack occurs. In the absence of a single committee that can track global information (such as the join rate of IDs), setting the hardness of challenges is tricky, and new ideas are needed to obtain the conjectured upper bound.

Finally, while we have focused on DHTs, new defenses for them might generalize to providing security in permissionless settings for other structured P2P systems [12,20,47,51,62,71,158].

5 Application-Layer DDoS Attacks

A denial-of-service (DoS) attack prevents good IDs from accessing resources of a system. A distributed denial-of-service (DDoS) attack occurs when multiple bad IDs carry out a coordinated DoS attack. In an application-layer DDoS attacks, an adversary attacks by issuing many requests for system resources, as opposed to say swamping the network bandwidth. Here, we discuss defenses against application-layer DDoS attacks based on resource burning.

Filtering Methods. Many DDoS defenses rely on techniques for filtering out malicious traffic, including IP profiling [94,155]; CAPTCHAs [109,136]; capability-based schemes [9,149][4]; and anomaly detection [70]. An extensive survey of defenses can be found in [157]. Unfortunately, these techniques are imperfect, and an adversary may bypass them by issuing traffic that appears legit-

[4] Informally, this refers to a scheme where the source makes a "capability" request and, if approved by the receiver, will then obtain prioritized service from those routers along the path between the source and the receiver.

imate. This has led to resource-burning defenses against DDoS attacks, which are sometimes referred to in the literature as currency-based or resource-based schemes [139].

Resource-Burning Approaches. A number of proposed defenses require IDs to solve puzzles before their requests for service are honored [19,76,77,112]. A challenging aspect of these proposals is the lack of a theoretically-backed method to tune the puzzle difficulty. To address this issue, Mankins et al. [95] propose a pricing mechanism to set the difficulty based on the service-request type; however, the pricing functions are set by the server *a priori*, and may fail as the incentives or capabilities of the attacker change over time. A dynamic strategy to determine puzzle difficulty is given by Wang and Reiter [142]. A client requesting service chooses the puzzle difficulty based on the effort it is willing to expend, while the server prioritizes service according to the difficulty of the puzzles solved. However, this approach may starve IDs with limited resources, and requires the server to maintain state on the difficulty of the puzzles solved. Finally, Noureddine et al. [108] employ a game-theoretic model to pre-compute the difficulty of puzzles assuming all IDs (good and bad) are rational.

An alternative resource—communication capacity—is consumed by the *speak-up* defense of Walfish et al. [138]. During an attack, it is common for bad IDs to bombard the server with requests, using much (or all) of the data rate available to the adversary. Speak-up encourages good IDs to respond in kind by increasing their respective request rates. A front-end server known as a "thinner" randomly drops requests in order to impose a manageable service load. If the aggregate capacity of the good IDs is comparable to that of the bad IDs, then this resource-burning scheme can allow good IDs to obtain a commensurate amount of service.

5.1 The Application-Layer DDoS Problem

There are many similarities between the application-layer DDoS attack and the Sybil attack. The DDoS model is not purely permissionless, since the server is a trusted authority. However, the attacks involve IDs whose distinctness cannot be ascertained, and where an adversary may create many bad IDs to facilitate attacks. In this sense, the DDoS model is a hybrid of permissionless and permissioned systems. Thus, it is not surprising that resource burning would be useful to defend against DDoS attacks.

In this vein, we propose the open problem below.

Problem 4. **Application-Layer DDoS Attacks**

Model: There are n good client IDs and a good server. An adversary controls an α-fraction of the consumable resource, and can generate any number of bad client IDs. Client IDs can request service from the server at any time. The server must decide which requests to service based on its own limited resources.
Goal: The good clients obtain a $1-O(\alpha)$ fraction of the service provided by the server.

Problem 4 shares much in common with DefID (Sect. 3.1). Requests from client IDs correspond to join events; satisfying requests corresponds to departures. Here, α need not be bounded, since we are not making a correctness guarantee analogous to maintaining a good majority in DefID. Rather, our new requirement concerns performance: good IDs receive a $1 - O(\alpha)$ fraction of service. In this sense, Problem 3 seems strictly easier than DefID.

However, a new difficulty is heterogeneity: requests may differ in the amount of effort required to service them. Thus, enforcing a bound on the fraction of bad requests serviced does not ensure that the goal of Problem 4 will be met. In light of this issue, it may be helpful to consider a weighted version of DefID, and whether existing solutions can be extended to this more general setting. While we are optimistic that for large T, $o(T)$ is possible for Problem 4, a tight upper bound is an interesting direction for future work.

6 Review Spam

Online user-generated reviews play an important role in influencing the purchasing decisions of consumers. These systems are subject to manipulation where an adversary employs multiple accounts to create fake reviews that falsely promote or disparage a product [50]; this malicious behavior is often referred to as ***review spam***, but also goes by other labels such as *astroturfing* [133] and *opinion spam* [69].

Review spam threatens online retailers—such as Amazon or Walmart [33,50]—and merchants who depend on income from online sales. While online review systems typically have some form of admission control, such as requiring credentials for the creation of an account, this can be bypassed. For example, an attacker can hire users that possess a sufficient online presence in order to engage in review spam [36,66], and social-media credentials can be automatically generated [133]; examples of these attacks are described in [69,93].

In response to this threat, the research community has proposed various strategies for detecting fraudulent reviews; these employ a range of techniques including machine learning [34,73], anomaly detection [123,145,146], linguistic evaluation [79,115], graph analysis [5,28,66], and many others. A comprehensive overview of these techniques is given in [64,144,150].

Progress in this area offers the ability to classify a review as either spam or legitimate, with some small error probability; for example, the work in [110] achieve an accuracy of almost 90%. This classification functionality is a promising ingredient for designing more general tools for mitigating review spam.

6.1 The Review Spam Problem

The problem of review spam largely aligns with our general model in Sect. 2.5. While online systems often require some credentials for creating an account, this admission control can be circumvented, and the system is effectively permissionless. However, the review spam model has some novel features. IDs join the system, but they may never formally depart. Even IDs that are regularly in use may have periods where the corresponding user is offline. Thus, any attempt to simultaneously challenge all IDs, in order to reveal some as bad, will fail.

On the positive side, as noted above, machine learning can now help. In particular, we may assume a classifier that correctly classifies reviews as spam or not with some fixed probability of error. Over a sufficiently large number of reviews, this classifier can be used to obtain a good approximation of the current fraction of spam reviews, and this information can be used to set the amount of resource burning required to post a review. Our conjecture of $O(T^{2/3} + P_G)$ in Table 1 follows from a preliminary analysis that leverages a classifier in this way. Informally, we increase the cost for posting a review when a significant attack is ongoing—that is, many reviews are diagnosed as spam by the classifier. Otherwise, we reset the cost to the lowest level.

We formalize the challenge of review spam as Problem 5.

Problem 5. **Review Spam**

Model: IDs post reviews online. A classifier labels each post as legitimate or as spam, with some fixed error probability. Each spam post has unit cost, reflecting its negative impact on system usability. The algorithm can also set an arbitrary resource-burning cost for each new post, based on the classification of past posts.
Goal: Minimize costs due to spam posts plus resource-burning costs incurred from legitimate posts.

7 Conclusion

In this paper, we surveyed the literature on resource burning and established it as critical a tool for securing permissionless systems. We described results from four domains: blockchains, DHTs, application-layer distributed DDoS attacks, and review spam. We noted shared security vulnerabilities in both permissionless and hybrid systems, and how resource burning is well-suited for addressing common threats.

We observed that resource burning costs are prohibitively high for most current systems. Thus, a high-priority area for theoretical research is the design of resource-burning defenses that reduce these costs. In particular, whenever possible, good IDs should spend at a rate which is asymptotically less than the adversary when the system is under attack. To encourage research efforts, we defined several open problems, along with conjectured upper bounds for these problems.

Acknowledgements. We are grateful to the organizers of SIROCCO 2020 for inviting this paper, and we thank Valerie King for helpful feedback on our manuscript.

References

1. Abadi, M., Burrows, M., Manasse, M., Wobber, T.: Moderately hard, memory-bound functions. ACM Trans. Internet Technol. (TOIT) **5**(2), 299–327 (2005)
2. Abraham, I., Malkhi, D.: The Blockchain Consensus Layer and BFT. Bull. EATCS: Distrib. Comput. Column (2017)
3. Abraham, I., Malkhi, D., Dobzinski, O.: Land: stretch $(1 + \epsilon)$ locality-aware networks for DHTs. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete algorithms (SODA), pp. 550–559 (2004)
4. Aggarwal, A., Movahedi, M., Saia, J., Zamani, M.: Bootstrapping public blockchains without a trusted setup. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, pp. 366–368. ACM (2019)
5. Akoglu, L., Chandy, R., Faloutsos, C.: Opinion fraud detection in online reviews by network effects. In: Seventh International AAAI Conference on Weblogs and Social Media (2013)
6. Ali, I.M., Caprolu, M., Pietro, R.D.: Foundations, properties, and security applications of puzzles: a survey. CoRR abs/1904.10164 (2019). http://arxiv.org/abs/1904.10164
7. Alvisi, L., Clement, A., Epasto, A., Lattanzi, S., Panconesi, A.: SoK: the evolution of sybil defense via social networks. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 382–396 (2013)
8. Hertig, A.: Ethereum's big switch: the new roadmap to proof-of-stake (2017). urlwww.coindesk.com/ethereums-big-switch-the-new-roadmap-to-proof-of-stake/. Accessed 28 Nov 2019
9. Anderson, T., Roscoe, T., Wetherall, D.: Preventing internet denial-of-service with capabilities. ACM SIGCOMM Comput. Commun. Rev. **34**(1), 39–44 (2004)
10. Andrychowicz, M., Dziembowski, S.: PoW-based distributed cryptography with no trusted setup. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 379–399. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_19
11. Aspnes, J., Jackson, C., Krishnamurthy, A.: Exposing computationally-challenged byzantine impostors. Technical report, Technical Report YALEU/DCS/TR-1332, Yale University (2005). http://www.cs.yale.edu/homes/aspnes/papers/tr1332.pdf
12. Aspnes, J., Shah, G.: Skip graphs. In: Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 384–393 (2003)
13. Ateniese, G., Bonacina, I., Faonio, A., Galesi, N.: Proofs of space: when space is of the essence. In: Abdalla, M., De Prisco, R. (eds.) SCN 2014. LNCS, vol. 8642, pp. 538–557. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10879-7_31
14. Augustine, J., Molla, A.R., Morsy, E., Pandurangan, G., Robinson, P., Upfal, E.: Storage and search in dynamic peer-to-peer networks. In: Proceedings of the Twenty-fifth Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 53–62 (2013)

15. Augustine, J., Pandurangan, G., Robinson, P.: Fast byzantine agreement in dynamic networks. In: Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC), pp. 74–83 (2013)
16. Augustine, J., Pandurangan, G., Robinson, P.: Fast byzantine leader election in dynamic networks. In: Moses, Y. (ed.) DISC 2015. LNCS, vol. 9363, pp. 276–291. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48653-5_19
17. Augustine, J., Pandurangan, G., Robinson, P., Roche, S., Upfal, E.: Enabling robust and efficient distributed computation in dynamic peer-to-peer networks. In: Proceedings of the IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS), pp. 350–369 (2015)
18. Augustine, J., Pandurangan, G., Robinson, P., Upfal, E.: Towards robust and efficient computation in dynamic peer-to-peer networks. In: Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 551–569 (2012)
19. Aura, T., Nikander, P., Leiwo, J.: DOS-resistant authentication with client puzzles. In: Christianson, B., Malcolm, J.A., Crispo, B., Roe, M. (eds.) Security Protocols 2000. LNCS, vol. 2133, pp. 170–177. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44810-1_22
20. Awerbuch, B., Scheideler, C.: The hyperring: a low-congestion deterministic data structure for distributed environments. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 318–327 (2004)
21. Awerbuch, B., Scheideler, C.: Robust random number generation for peer-to-peer systems. In: Shvartsman, M.M.A.A. (ed.) OPODIS 2006. LNCS, vol. 4305, pp. 275–289. Springer, Heidelberg (2006). https://doi.org/10.1007/11945529_20
22. Awerbuch, B., Scheideler, C.: Towards a scalable and robust DHT. In: Proceedings of the 18th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 318–327 (2006)
23. Awerbuch, B., Scheideler, C.: Towards scalable and robust overlay networks. In: Proceedings of the 6th International Workshop on Peer-to-Peer Systems (IPTPS), p. n. pag. (2007)
24. Baird, H.S., Moll, M.A., Wang, S.Y.: ScatterType: a legible but hard-to-segment CAPTCHA. In: Proceedings of the Eighth International Conference on Document Analysis and Recognition (ICDAR), pp. 935–939 (2005)
25. Ball, M., Rosen, A., Sabin, M., Vasudevan, P.N.: Proofs of work from worst-case assumptions. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018. LNCS, vol. 10991, pp. 789–819. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96884-1_26
26. Bazzi, R.A., Konjevod, G.: On the establishment of distinct identities in overlay networks. In: Proceedings 24th Annual ACM Symposium on Principles of Distributed Computing (PODC), pp. 312–320 (2005)
27. Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M.: Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract] y. ACM SIGMETRICS Perform. Eval. Rev. **42**(3), 34–37 (2014)
28. Beutel, A., Xu, W., Guruswami, V., Palow, C., Faloutsos, C.: CopyCatch: stopping group attacks by spotting lockstep behavior in social networks. In: Proceedings of the 22nd International Conference on World Wide Web (WWW), pp. 119–130 (2013)
29. Binmore, K., et al.: Playing for Real: A Text on Game Theory. Oxford University Press, Oxford (2007)
30. BitcoinWiki: Bitcoinwiki network (2019). https://en.bitcoin.it/wiki/Network. Accessed 28 Nov 2019

31. Borisov, N.: Computational puzzles as sybil defenses. In: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing (P2P), pp. 171–176 (2006)
32. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. In: Proceedings of the 5th Usenix Symposium on Operating Systems Design and Implementation (OSDI), pp. 299–314 (2002)
33. CBS News, A.P.: Buyer beware: scourge of fake reviews hitting Amazon, Walmart and other major retailers (2019). https://www.cbsnews.com/news/buyer-beware-a-scourge-of-fake-online-reviews-is-hitting-amazon-walmart-and-other-major-retailers/
34. Chau, D.H., Pandit, S., Faloutsos, C.: Detecting fraudulent personalities in networks of online auctioneers. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 103–114. Springer, Heidelberg (2006). https://doi.org/10.1007/11871637_14
35. CoinDesk: Vulnerable? Ethereum's Casper Tech Takes Criticism at Curacao Event (2018). https://www.coindesk.com/fundamentally-vulnerable-ethereums-casper-tech-takes-criticism-curacao
36. Cracked: I get paid to write fake reviews for amazon (2016). https://www.cracked.com/personal-experiences-2376-i-get-paid-to-write-fake-reviews-amazon.html
37. Danezis, G., Lesniewski-Laas, C., Kaashoek, M.F., Anderson, R.: Sybil-resistant DHT routing. In: di Vimercati, S.C., Syverson, P., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 305–318. Springer, Heidelberg (2005). https://doi.org/10.1007/11555827_18
38. Demirbas, M., Song, Y.: An RSSI-based scheme for sybil attack detection in wireless sensor networks. In: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks (WOWMOM), pp. 564–570 (2006)
39. Digiconomist: Bitcoin energy consumption index (2020). https://digiconomist.net/bitcoin-energy-consumption
40. Dinger, J., Hartenstein, H.: Defending the sybil attack in P2P networks: taxonomy, challenges, and a proposal for self-registration. In: Proceedings of the First International Conference on Availability, Reliability and Security (ARES), pp. 756–763 (2006)
41. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_24
42. Dwork, C., Goldberg, A., Naor, M.: On memory-bound functions for fighting spam. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 426–444. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_25
43. Dwork, C., Naor, M.: Pricing via Processing or combatting junk mail. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 139–147. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-48071-4_10
44. Dziembowski, S., Faust, S., Kolmogorov, V., Pietrzak, K.: Proofs of space. In: Gennaro, R., Robshaw, M. (eds.) CRYPTO 2015. LNCS, vol. 9216, pp. 585–605. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48000-7_29
45. Easley, D., Kleinberg, J., et al.: Networks, Crowds, and Markets, vol. 8. Cambridge University Press, Cambridge (2010)
46. Falkner, J., Piatek, M., John, J.P., Krishnamurthy, A., Anderson, T.: Profiling a million user DHT. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 129–134 (2007)

47. Fiat, A., Saia, J.: Censorship resistant peer-to-peer content addressable networks. In: Proceedings of the Thirteenth ACM Symposium on Discrete Algorithms (SODA), pp. 94–103 (2002)
48. Fiat, A., Saia, J., Young, M.: Making chord robust to byzantine attacks. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 803–814. Springer, Heidelberg (2005). https://doi.org/10.1007/11561071_71
49. FitzGibbon, C.D., Fanshawe, J.H.: Stotting in Thomson's gazelles: an honest signal of condition. Behav. Ecol. Sociobiol. **23**(2), 69–74 (1988)
50. Forbes: Amazon's fake review problem is getting worse (2019). https://www.forbes.com/sites/emmawoollacott/2019/04/16/amazons-fake-review-problem-is-getting-worse/#f6988195f525
51. Fraigniaud, P., Gauron, P.: D2B: a de bruijn based content-addressable network. Theoret. Comput. Sci. **355**(1), 65–79 (2006)
52. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: analysis and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 281–310. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_10
53. Gil, S., Kumar, S., Mazumder, M., Katabi, D., Rus, D.: Guaranteeing spoof-resilient multi-robot networks. In: Proceedings of Robotics: Science and Systems, Rome, Italy, July 2015
54. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles (SOSP), pp. 51–68 (2017)
55. Gilbert, S., Newport, C., Zheng, C.: Who are you? Secure identities in ad hoc networks. In: Kuhn, F. (ed.) DISC 2014. LNCS, vol. 8784, pp. 227–242. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45174-8_16
56. Gilbert, S., Zheng, C.: SybilCast: broadcast on the open airwaves. In: Proceedings of the 25th Annual ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp. 130–139 (2013)
57. Guerraoui, R., Huc, F., Kermarrec, A.M.: Highly dynamic distributed computing with byzantine failures. In: Proceedings of the 2013 ACM Symposium on Principles of Distributed Computing (PODC), pp. 176–183 (2013)
58. Gupta, D., Saia, J., Young, M.: Proof of work without all the work. In: Proceedings of the 19th International Conference on Distributed Computing and Networking (ICDCN) (2018)
59. Gupta, D., Saia, J., Young, M.: Peace through superior puzzling: an asymmetric sybil defense. In: Proceedings of the 33rd IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 1083–1094 (2019)
60. Gupta, D., Saia, J., Young, M.: ToGCom: an asymmetric sybil defense. arXiv preprint arXiv:2006.02893 (2020)
61. Hartline, J.D., Roughgarden, T.: Optimal mechanism design and money burning. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 75–84 (2008)
62. Harvey, N.J.A., Jones, M.B., Saroiu, S., Theimer, M., Wolman, A.: Skipnet: a scalable overlay network with practical locality properties. In: USENIX Symposium on Internet Technologies and Systems (2003)
63. Heilman, E., Kendler, A., Zohar, A., Goldberg, S.: Eclipse attacks on bitcoin's peer-to-peer network. In: Proceedings of the 24th USENIX Conference on Security Symposium, pp. 129–144 (2015)

64. Heydari, A., AliTavakoli, M., Salim, N., Heydari, Z.: Detection of review spam: a survey. Expert Syst. Appl. **42**(7), 3634–3642 (2015). https://doi.org/10.1016/j.eswa.2014.12.029. http://www.sciencedirect.com/science/article/pii/S0957417414008082
65. Hildrum, K., Kubiatowicz, J.: Asymptotically efficient approaches to fault-tolerance in peer-to-peer networks. In: Fich, F.E. (ed.) DISC 2003. LNCS, vol. 2848, pp. 321–336. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39989-6_23
66. Hooi, B., Song, H.A., Beutel, A., Shah, N., Shin, K., Faloutsos, C.: FRAUDAR: bounding graph fraud in the face of camouflage. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 895–904. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2939672.2939747
67. Hou, R., Jahja, I., Luu, L., Saxena, P., Yu, H.: Randomized view reconciliation in permissionless distributed systems, pp. 2528–2536 (2018)
68. Huck, S., Müller, W.: Burning money and (pseudo) first-mover advantages: an experimental study on forward induction. Games Econ. Behav. **51**(1), 109–127 (2005)
69. Hunt, K.M.: Gaming the system: fake online reviews v. consumer law. Comput. Law Secur. Rev. **31**(1), 3–25 (2015). http://www.sciencedirect.com/science/article/pii/S0267364914001824
70. Hussain, A., Heidemann, J., Papadopoulos, C.: A framework for classifying denial of service attacks. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 99–110 (2003)
71. Jagadish, H., Ooi, B.C., Vu, Q.H.: BATON: a balanced tree structure for peer-to-peer networks. In: Proceedings of the 31st International conference on Very Large Data Bases (VLDB), pp. 661–672 (2005)
72. Jaiyeola, M.O., Patron, K., Saia, J., Young, M., Zhou, Q.M.: Tiny groups tackle byzantine adversaries. In: Proceedings of the IEEE International Parallel and Distributed Processing Symposium, IPDPS, pp. 1030–1039 (2018)
73. Jindal, N., Liu, B.: Opinion spam and analysis. In: Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219–230 (2008)
74. Johansen, H., Allavena, A., van Renesse, R.: Fireflies: scalable support for intrusion-tolerant network overlays. In: ACM SIGOPS Operating Systems Review, pp. 3–13 (2006)
75. John, R., Cherian, J.P., Kizhakkethottam, J.J.: A survey of techniques to prevent sybil attacks. In: Proceedings of the International Conference on Soft-Computing and Networks Security (ICSNS), pp. 1–6 (2015)
76. Juels, A., Brainard, J.: Client puzzles: a cryptographic countermeasure against connection depletion attacks. In: Proceedings of the Network and Distributed System Security Symposium (NDSS), pp. 151–165 (1999)
77. Kaiser, E., Feng, W.C.: Mod_kaPoW: mitigating DoS with transparent proof-of-work. In: Proceedings of the 2007 ACM CoNEXT Conference, pp. 74:1–74:2 (2007)
78. Kapadia, A., Triandopoulos, N.: Halo: High-assurance locate for distributed hash tables. In: Proceedings of the Network and Distributed System Security Symposium (NDSS) (2008)
79. Karami, A., Zhou, B.: Online review spam detection by new linguistic features. In: iConference 2015 Proceedings (2015)

80. Kaashoek, M.F., Karger, D.R.: Koorde: a simple degree-optimal distributed hash table. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, pp. 98–107. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45172-3_9
81. Katz, J., Miller, A., Shi, E.: Pseudonymous secure computation from time-lock puzzles. IACR Cryptol. ePrint Arch. 2014, 857 (2014). http://eprint.iacr.org/2014/857
82. Khan, S.M., Mallesh, N., Nambiar, A., Wright, M.K.: The dynamics of salsa: a robust structured P2P system. Netw. Protocols Algorithms **2**, 40–60 (2010)
83. Kiayias, A., Russell, A., David, B., Oliynykov, R.: Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 357–388. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_12
84. Knockel, J., Saad, G., Saia, J.: Self-healing of byzantine faults. In: Higashino, T., Katayama, Y., Masuzawa, T., Potop-Butucaru, M., Yamashita, M. (eds.) SSS 2013. LNCS, vol. 8255, pp. 98–112. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03089-0_8
85. Laurie, B., Clayton, R.: "Proof-of-work" proves not to work. In: Proceedings of the 3rd Annual Workshop on Economics and Information Security (WEIS) (2004)
86. Lesniewski-Laas, C., Kaashoek, M.F.: Whanau: a sybil-proof distributed hash table. In: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, NSDI 2010 , p. 8 (2010)
87. Li, D., Lu, X., Wu, J.: FISSIONE: a scalable constant degree and low congestion DHT scheme based on Kautz graphs. In: Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1677–1688 (2005)
88. Li, F., Mittal, P., Caesar, M., Borisov, N.: SybilControl: practical sybil defense with computational puzzles. In: Proceedings of the Seventh ACM Workshop on Scalable Trusted Computing, pp. 67–78 (2012)
89. Lin, I.C., Liao, T.C.: A survey of blockchain security issues and challenges. IJ Netw. Secur. **19**(5), 653–659 (2017)
90. Liu, D., Camp, L.J.: Proof of work can work. In: Proceedings of the 5th Workshop on the Economics of Information Security (WEIS) (2006)
91. Liu, Y., Bild, D.R., Dick, R.P., Mao, Z.M., Wallach, D.S.: The Mason test: a defense against sybil attacks in wireless networks without trusted authorities. IEEE Trans. Mob. Comput. **14**(11), 2376–2391 (2015)
92. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 17–30 (2016)
93. Malbon, J.: Taking fake online consumer reviews seriously. J. Consum. Policy **36**(2), 139–157 (2013)
94. Malliga, S., Tamilarasi, A., Janani, M.: Filtering spoofed traffic at source end for defending against DoS/DDoS attacks. In: Proceedings of the International Conference on Computing, Communication and Networking, pp. 1–5. IEEE (2008)
95. Mankins, D., Krishnan, R., Boyd, C., Zao, J., Frentz, M.: Mitigating distributed denial of service attacks with dynamic resource pricing. In: Proceedings of the Seventeenth Annual Computer Security Applications Conference, pp. 411–421. IEEE (2001)
96. Maymounkov, P., Mazières, D.: Kademlia: a peer-to-peer information system based on the XOR metric. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 53–65. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_5

97. Miller, A., et al.: Discovering bitcoin's public topology and influential nodes (2015). http://cs.umd.edu/projects/coinscope/coinscope.pdf
98. Miller, G.: Spent: Sex, Evolution, and Consumer Behavior. Penguin, New York (2009)
99. Mohaisen, A., Hollenbeck, S.: Improving social network-based sybil defenses by rewiring and augmenting social graphs. In: Kim, Y., Lee, H., Perrig, A. (eds.) WISA 2013. LNCS, vol. 8267, pp. 65–80. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-05149-9_5
100. Mohaisen, A., Kim, J.: The sybil attacks and defenses: a survey. Smart Comput. Rev. **3**(6), 480–489 (2013)
101. Mónica, D., Leitao, L., Rodrigues, L., Ribeiro, C.: On the use of radio resource tests in wireless ad-hoc networks. In: Proceedings of the 3rd Workshop on Recent Advances on Intrusion-Tolerant Systems, pp. 21–26 (2009)
102. Moran, T., Orlov, I.: Simple proofs of space-time and rational proofs of storage. In: Boldyreva, A., Micciancio, D. (eds.) CRYPTO 2019. LNCS, vol. 11692, pp. 381–409. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-26948-7_14
103. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system (2008). http://bitcoin.org/bitcoin.pdf
104. Nambiar, A., Wright, M.: Salsa: a structured approach to large-scale anonymity. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 17–26 (2006)
105. Naor, M., Wieder, U.: Novel architectures for P2P applications: the continuous-discrete approach. In: Proceedings of the 15th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA) (2003)
106. Newsome, J., Shi, E., Song, D., Perrig, A.: The sybil attack in sensor networks: analysis & defenses. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks (IPSN), pp. 259–268 (2004)
107. Nolan, C.: The Dark Knight. Quote from the scene where the Joker sets a large pile of money ablaze (2008)
108. Noureddine, M.A., et al.: Revisiting client puzzles for state exhaustion attacks resilience. In: Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), pp. 617–629 (2019)
109. Oikonomou, G., Mirkovic, J.: Modeling human behavior for defense against flash-crowd attacks. In: Proceedings of the IEEE International Conference on Communications, pp. 1–6. IEEE (2009)
110. Ott, M., Choi, Y., Cardie, C., Hancock, J.T.: Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 309–319. Association for Computational Linguistics, USA (2011)
111. Park, S., Kwon, A., Fuchsbauer, G., Gaži, P., Alwen, J., Pietrzak, K.: SpaceMint: a cryptocurrency based on proofs of space. In: Meiklejohn, S., Sako, K. (eds.) FC 2018. LNCS, vol. 10957, pp. 480–499. Springer, Heidelberg (2018). https://doi.org/10.1007/978-3-662-58387-6_26
112. Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Hu, Y.C.: Portcullis: protecting connection setup from denial-of-capability attacks. ACM SIGCOMM Comput. Commun. Rev. **37**(4), 289–300 (2007)
113. Penn, D.J.: The evolutionary roots of our environmental problems: toward a darwinian ecology. Q. Rev. Biol. **78**(3), 275–301 (2003)
114. Pogue, D.: Time to kill off captchas. Sci. Am. **306**(3), 23–23 (2012)

115. Rayana, S., Akoglu, L.: Collective opinion spam detection: bridging review networks and metadata. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015, pp. 985–994. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2783258.2783370
116. Rowaihy, H., Enck, W., McDaniel, P., La Porta, T.: Limiting Sybil attacks in structured P2P networks. In: Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM), pp. 2596–2600 (2007)
117. Rowstron, A.I.T., Druschel, P.: Pastry: scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms, pp. 329–350 (2001)
118. Saad, G.: The Evolutionary Bases of Consumption. Psychology Press (2007)
119. Saad, G., Vongas, J.G.: The effect of conspicuous consumption on men's testosterone levels. Organ. Behav. Hum. Decis. Process. **110**(2), 80–92 (2009)
120. Saad, G., Saia, J.: Self-healing computation. In: Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS), pp. 195–210 (2014)
121. Saad, G., Saia, J.: A theoretical and empirical evaluation of an algorithm for self-healing computation. Distrib. Comput. **30**(6), 391–412 (2017)
122. Saia, J., Young, M.: Reducing communication costs in robust peer-to-peer networks. Inform. Process. Lett. **106**(4), 152–158 (2008)
123. Savage, D., Zhang, X., Yu, X., Chou, P., Wang, Q.: Detection of opinion spam based on anomalous rating deviation. Expert Syst. Appl. **42**(22), 8650–8657 (2015). https://doi.org/10.1016/j.eswa.2015.07.019. http://www.sciencedirect.com/science/article/pii/S0957417415004790
124. Scheideler, C., Schmid, S.: A distributed and oblivious heap. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part II. LNCS, vol. 5556, pp. 571–582. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02930-1_47
125. Sen, S., Freedman, M.J.: Commensal cuckoo: secure group partitioning for large-scale services. ACM SIGOPS Oper. Syst. **46**(1), 33–39 (2012)
126. Shoker, A.: Sustainable blockchain through proof of exercise. In: 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), pp. 1–9. IEEE (2017)
127. Singh, A., Ngan, T.W., Druschel, P., Wallach, D.S.: Eclipse attacks on overlay networks: threats and defenses. In: Proceedings IEEE International Conference on Computer Communications (INFOCOM), pp. 1–12 (2006)
128. Steiner, M., En-Najjary, T., Biersack, E.W.: A global view of KAD. In: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, pp. 117–122 (2007)
129. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: a scalable peer-to-peer lookup service for internet applications. In: Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 149–160 (2001)
130. Stoica, I., et al.: Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Trans. Netw. **11**(1), 17–32 (2003). https://doi.org/10.1109/TNET.2002.808407
131. Sundie, J.M., Kenrick, D.T., Griskevicius, V., Tybur, J.M., Vohs, K.D., Beal, D.J.: Peacocks, porsches, and thorstein veblen: Conspicuous consumption as a sexual signaling system. J. Pers. Soc. Psychol. **100**(4), 664 (2011)

132. Tegeler, F., Fu, X.: SybilConf: computational puzzles for confining sybil attacks. In: Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM), pp. 1–2 (2010)
133. The Guardian, G.M.: The need to protect the internet from ‘astroturfing’ grows ever more urgent (2011). https://www.theguardian.com/environment/georgemonbiot/2011/feb/23/need-to-protect-internet-from-astroturfing
134. Thorstein, V.: The Theory of the Leisure Class: An Economic Study of Institutions. BW Huebsch, New York (1912)
135. Urdaneta, G., Pierre, G., van Steen, M.: A survey of DHT security techniques. ACM Comput. Surv. **43**(2), 1–53 (2011)
136. von Ahn, L., Blum, M., Hopper, N.J., Langford, J.: CAPTCHA: using hard AI problems for security. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 294–311. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_18
137. Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: human-based character recognition via web security measures. Science **321**(5895), 1465–1468 (2008)
138. Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D., Shenker, S.: DDoS defense by offense. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), pp. 303–314 (2006)
139. Walfish, M., Vutukuru, M., Balakrishnan, H., Karger, D., Shenker, S.: DDoS defense by offense. ACM Trans. Comput. Syst. (TOCS) **28**(1), 3 (2010)
140. Wang, H., Zhu, Y., Hu, Y.: An efficient and secure peer-to-peer overlay network. In: Proceedings of the IEEE Conference on Local Computer Networks, pp. 764–771 (2005)
141. Wang, L., Kangasharju, J.: Measuring large-scale distributed systems: case of BitTorrent mainline DHT. In: IEEE 13th International Conference on Peer-to-Peer Computing (P2P), pp. 1–10 (2013)
142. Wang, X., Reiter, M.K.: Defending against denial-of-service attacks with puzzle auctions. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy, p. 78 (2003)
143. Wei, W., Xu, F., Tan, C.C., Li, Q.: SybilDefender: a defense mechanism for sybil attacks in large social networks. IEEE Trans. Parallel Distrib. Syst. **24**(12), 2492–2502 (2013)
144. Wu, Y., Ngai, E.W., Wu, P., Wu, C.: Fake online reviews: literature review, synthesis, and directions for future research. Decis. Support Syst. **132**, 113280 (2020). https://doi.org/10.1016/j.dss.2020.113280. http://www.sciencedirect.com/science/article/pii/S016792362030035X
145. Xie, S., Wang, G., Lin, S., Yu, P.S.: Review spam detection via temporal pattern discovery. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 823–831. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2339530.2339662
146. Xie, S., Wang, G., Lin, S., Yu, P.S.: Review spam detection via time series pattern discovery. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012 Companion, pp. 635–636. Association for Computing Machinery, New York (2012). https://doi.org/10.1145/2187980.2188164
147. Yan, J., El Ahmad, A.S.: Captcha robustness: a security engineering perspective. Computer **44**(2), 54–60 (2011)

148. Yan, J., El Ahmad, A.S.: Usability of CAPTCHAs or usability issues in CAPTCHA design. In: Proceedings of the 4th Symposium on Usable Privacy and Security, SOUPS 2008, pp. 44–52. Association for Computing Machinery, New York (2008). https://doi.org/10.1145/1408664.1408671
149. Yang, X., Wetherall, D., Anderson, T.: TVA: A DoS-limiting network architecture. IEEE/ACM Trans. Netw. **16**(6), 1267–1280 (2008)
150. Ma, Y., Li, F.: Detecting review spam: challenges and opportunities. In: 8th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 651–654 (2012)
151. Young, M., Kate, A., Goldberg, I., Karsten, M.: Towards practical communication in Byzantine-resistant DHTs. IEEE/ACM Trans. Netw. **21**(1), 190–203 (2013)
152. Yu, H.: Sybil defenses via social networks: a tutorial and survey. SIGACT News **42**(3), 80–101 (2011)
153. Yu, H., Gibbons, P.B., Kaminsky, M., Xiao, F.: SybilLimit: a near-optimal social network defense against sybil attacks. IEEE/ACM Trans. Netw. **18**(3), 885–898 (2010)
154. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: SybilGuard: defending against sybil attacks via social networks. In: Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM) vol. 36, pp. 267–278, August 2006
155. Yu, S., Thapngam, T., Liu, J., Wei, S., Zhou, W.: Discriminating DDoS flows from flash crowds using information distance. In: Proceedings of the Third International Conference on Network and System Security, pp. 351–356. IEEE (2009)
156. Zahavi, A.: Mate selection - a selection for a handicap. J. Theor. Biol. **53**(1), 205–214 (1975)
157. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Commun. Surv. Tutor. **15**(4), 2046–2069 (2013)
158. Zatloukal, K.C., Harvey, N.J.A.: Family trees: an ordered dictionary with optimal congestion, locality, degree, and search time. In: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 308–317 (2004)