P2P Routing Table Poisoning: A Quorum-based Sanitizing Approach

Hatem Ismail[a], Daniel Germanus[b], Neeraj Suri[a]

[a]*DEEDS Group, TU Darmstadt, Germany*
[b]*ENX Association, Frankfurt, Germany*

Abstract

Peer-to-Peer (P2P) protocols underlie multiple networked applications given that the P2P decentralized design inherently fosters scalability and robustness. While distributiveness and scalability are attractive features, these facets also increase exposure to malicious peers which can propagate malicious routing information. Accordingly, a diverse set of continuously evolving attacks can be mounted that can cause severe service impairments over the entire overlay network. Most proposed countermeasures focus on providing diversity or redundancy to overcome malicious routing information with their emphasis on periodic detection/removal mechanisms done locally within a peer as continuous monitoring or global sharing of peer status entails high costs. However, a local approach naturally also limits the global effectiveness prompting the need for distributed solutions.

In this work, we build upon contemporary distributed solutions (that developed specific attack detection and mitigation techniques for specific overlay types and specific attacks), to propose a generalized attack detection and mitigation approach applicable to varied overlay and attack models.

Consequently, we propose a novel and efficient routing table sanitizing approach that (a) is independent of a specific attack variant, lookup approach or a specific victim set, (b) continuously detects and subsequently removes malicious routing information based on distributed quorum decisions, and (c) efficiently forwards malicious information findings to other peers which allows for progressive global sanitizing. The generalized mechanism shows a high sanitizing accuracy of up to 90% when evaluated against a generalized attack scenario with various adversarial behaviors.

Keywords: P2P, Sanitizing, Attacks, Detection, Security

1. Introduction

The P2P paradigm utilizes decentralized coordination to provide scalability and fault tolerance, which naturally leads to its wide applicability in diverse data dissemination and data discovery applications such as file sharing, multimedia streaming, machine-to-machine communication, IoT and many others [1, 2]. In order to support scalability and low overheads in P2P networks, the design practices typically result in partitioned groups where a peer has only a partial view of the network as obtained from its neighboring peers.

However, the aforementioned design practices render P2P networks susceptible to various attacks, e.g., routing table poisoning, which is an inherent part of composite attacks such as Eclipse (EA), Sybil, flooding and publishing attacks [3, 4, 5, 6, 7, 8]. While the fault tolerance aspect ensures correct operation even for high rates of random peer failures, the disruptions inserted into the peers routing tables (RT) as a form of Routing Table Poisoning (RTP) result in significant degradation of the network services. Notably, using a detailed simulation study, we demonstrate the significant RTP impact of up to 65% message loss. Moreover, we illustrate how the propagation of malicious RT information about the victim peers of RTP attacks further facilitates launching Eclipse, Sybil and other aforementioned attacks.

The existence of RTP attacks and the resulting degradation have received attention [9, 10, 11]. A considerable variety of proposed countermeasures [12, 13, 14, 15, 16, 17] exists, yet these techniques entail one or more of the following inefficiency drawbacks: (i) They are only applicable for a specific P2P protocol, i.e., the countermeasure mechanisms are specifically tailored according to a single P2P protocol specifications. (ii) They are effective against a single form of

Email addresses:
hayman@deeds.informatik.tu-darmstadt.de (Hatem Ismail),
daniel.germanus@gmail.com (Daniel Germanus),
suri@cs.tu-darmstadt.de (Neeraj Suri)

RTP attack. Hence, countermeasures show no resiliency once the attack is modified. (iii) They typically require a central entity that coordinates the detection, monitoring, and decisions about malicious peers. However, in practice, the system's services are degraded as the overlay's fully distributed architecture is compromised. (iv) They often rely on cryptographic schemes, which can then constrain communication between lightweight peers to necessitate enhanced computing.

Aiming towards finding a general solution to overcome the aforementioned deficiencies, we explored a detection and sanitizing scheme in [18] as a countermeasure against a single attack variant of Localized Attacks (LAs). We build upon the basic notions of providing anonymous detection from our proposed mechanism in [18] to develop a generalized attack handling approach applicable to multiple attack models and overlays.

Contributions: In the course of our previous work, we develop an adaptable RTP attack mitigation approach that overcomes the aforementioned deficiencies. We propose a protocol-independent, fully distributed, simple and effective detection and overlay-sanitizing mechanism.

As a mean of an adaptable mitigation, we make use of a majority voting based detection in order to detect inconsistencies in RTs. The detection mechanism shows high accuracy with detection rates up of to 90% even for 20% malicious peers attacking. The sanitizing mechanism is triggered by initiating a quorum of peers in order to unveil the inconsistencies stemming from RTP attacks. Once the quorum investigates and accordingly declares finding malicious RT entries, the sanitizing mechanism informs other peers in order to let them reliably remove the RT information inserted by the suspected malicious peer.

Overall, our contributions span (i) demonstrating the high impact of RTP attacks on benign peers RTs and the overall network's service provision, and (ii) proposing a novel quorum based sanitizing mechanism that efficiently removes malicious peers and propagates information about their identity while providing anonymity and scalability.

Paper Structure

The rest of the paper is organized as follows: Section 2 presents the technical background along with related work. Section 3 provides the system model and defines the concepts underlying the attacker model (Section 4), the detection mechanism (Section 5) and the proposed sanitizing mechanism (Section 6). The attack severity, mitigation efficiency, and detection rates are evaluated in Section 7.

2. Related Work: Typical Attacks & Mitigation Approaches

Given the diverse set of applications that utilize the P2P functionality, a corresponding variety of attack types exists threatening the operations and reliability of P2P services. However, as routing constitutes a core P2P functionality, naturally most threats stem from deliberate attempts to compromise the peers routing tables with malicious information. Consequently, the launching of RT attacks on P2P networks has attracted considerable research interest.

While a variety of countermeasures are proposed, most existing techniques either address a specific P2P protocol or a specific adversarial behavior arising from the malicious side of the network. To that end, we discuss (i) the existing work that addresses the impact of related attacks and the feasibility of inserting malicious peers in peers RT, (ii) the existing mitigation, detection and sanitizing techniques and their respective pros and cons, and (iii) the main aspects and challenges for the development of the generalized attack handling mechanisms. We highlight the factors that lead to providing a generalized sanitizing mechanism (as in our proposal) for malicious information removal.

2.1. Contemporary Approaches

We categorize each of the following presented related work according to their objective and developed techniques. For each of the presented papers, we summarize the discussion listing the pros and cons for each of them, i.e., whether such mechanisms allow for malicious peers detection, sanitizing and propagating information about malicious peers or not.

2.1.1. Specific mitigation techniques

Here we discuss all the relevant mitigation and detection mechanisms. This category contains mechanisms which are: (i) only effective against a certain attack, (ii) only applicable in a specific topology, (iii) dependent on the lookup approach used or (iv) only consider secure routing mechanisms as a solution.

The authors in [19] present a technique termed SALSA to increase lookups successful rate in the presence of malicious peers. SALSA organizes the address space into groups, thus, each peer has a limited view of the overlay. Subsequently, an anonymous forwarding scheme is used to reliably deliver lookup requests while lowering the probability of malicious peers intercepting the lookup requests. Although the proposed technique shows a successful lookup delivery rate of up to 89%, a remarkable false negative rate is noticed where malicious peers can actually bias the lookup replies. In fact, experimental studies in [20] show that SALSA technique can be greatly compromised when the number of malicious peers averages around 20%. Moreover, the proposed technique is only applicable in structured overlays and no sanitizing is proposed. In fact, this work is relatively comparable to our mitigation scheme proposed in [21], where the mitigation approach is based on modifying the lookup forwarding protocol to prevent malicious peers from intercepting lookup requests.

Another countermeasure for RT pollution attacks was introduced in [16] that addresses P2P attacks in Smart Grids using auxiliary RT in Chord protocol [17]. However, the mitigation technique is not generalized as it is only applicable for P2P Chord based networks.

A random walk technique for structured overlays is proposed in [22], where peers periodically sign and certify their neighbors. Eventually, a secure route can be established for forwarding lookups. In [6], we proposed a similar random walk strategy to mitigate topology aware attacks, which yields reliable lookup delivery in trade of imposing lookup forwarding overhead on the overlay. Although these mitigation techniques effectively increase the overlay's reliability, they do not provide detection or removal schemes and are specific to either a specific overlay topology or a specific attack variant.

In [23], the authors propose a labeling mechanism named SybilInfer that identifies honest and malicious peers. SybilInfer relies on a probabilistic model of social networks. Analytically, SybilInfer efficiently provides high accuracy. However, the proposed mechanism is not applicable for traditional P2P networks as the mechanism assumes that the network is aware of social connections between users, which is true for a specific subset of P2P topologies.

Usphere [24] is a countermeasure mechanism against Sybil attacks that is based on a location-independent routing protocol. Usphere relies on the trust edge created by each peer towards its 1-hop neighbors. Although the results show high resiliency against Sybil with a path stretch of O(1), the proposed mechanism has the following drawbacks: (i) relies on elliptic curve cryptography [25], (ii) is only applicable to social-P2P based networks and (iii) considers neither any sanitizing mechanism nor any propagation of information about detected malicious peers.

2.1.2. Centralized authorities-based mitigation

A secure routing approach was introduced in [15] via encapsulating certificate authorities to peers' IDs during joining the network. Nevertheless, the proposed scheme relies on a centralized encryption authority.

In [26], a detection mechanism against Sybil attacks is proposed based on calculating trust values for each peer joining the overlay. However, the mechanism is effective against a single attack variant, relies on central authorities and no evaluation is provided.

2.1.3. Anonymity-based mitigation

In [18], we proposed an anonymous detection and eviction scheme as a countermeasure to Localized Attacks where LAs refer to attacks that only target a certain set of victims. Through the detection mechanism, peers are able to make a collaborative decision along with the other peers who received the lookup request. Hence, malicious peers are removed from the benign peers RT's. Given that our approach was focusing on a specific type of attacks, the detection and the sanitizing criteria can be characterized as being (i) attack specific, and (ii) with an absence of a rapid information propagation scheme to accelerate sanitizing the overlay from malicious peers that would, in turn, increase the cost of conducting an attack. Nevertheless, using the insights developed over [18], we build our generalized sanitizing mechanism utilizing the basic techniques of anonymous detection and attack sanitizing. The proposed sanitizing mechanism is evaluated against general attack scenarios, where general attack denotes common adversarial behaviors that constitute most of the well established attack forms in P2P networks.

As anonymity plays a major role in maintaining security in P2P networks and have a direct impact on the distribution of information and scalability, in [27], an anonymous low-latency networking protocol called Torsk is presented. Through the efficient relay selection and root verification schemes between peers, Torsk manages to mitigate various common P2P attacks in the Tor structured network [28] while allowing the network to scale. Similarly in [29], the authors present NISAN, an anonymous approach that assures high scalability while anonymously distributing the network information. In fact, NISAN shows high resiliency to known P2P attacks. Nevertheless, the absence of detection and sanitizing schemes might put the overlay at risk if the attacker manages to insert more malicious resources in the overlay or gain newly joined peers' trust during verification. In addition, the proposed protocol is only suitable for structured overlays.

In [7], the authors proposed an anonymous auditing scheme to mitigate eclipse attacks. The auditing scheme focuses on monitoring the ingoing and outgoing bounds of each peer and thus, detects peers that exceed a given threshold. Although the proposed technique allows for detecting maliciously behaving peers and locally removes those peers from the RT, no propagation or collaboration between peers to advertise such information about malicious peers exists.

In [30], the authors propose a partitioning scheme for large-scale overlays called Commensal cuckoo. Thus, such small groups cooperate to keep the group's decision correct despite of the launched join-leave attacks. Depending on several mechanisms such as secure routing, group authentication and bootstrapping, the proposed technique shows high resiliency even when higher fraction of malicious peers than the average state-of-the-art values exists. However, the proposed technique does not allow for propagating information about malicious peers and only addresses a single form of attack (join-leave).

2.1.4. *Reducing complexity and overhead-based techniques*

The authors in [31] discuss the overhead imposed by different mitigation and detection techniques in Distributed Hash Tables (DHTs) and how impractical these techniques are when applied to real world applications. Moreover, they present a technique to bound the message complexity when distributed quorums are required. Although the proposed technique remarkably lowers the overhead compared to already existing quorum-based techniques, the proposed technique: (i) uses a complicated cryptographic scheme and (ii) assumes that malicious peers in each quorum can maximally be $< 1/3$ of the quorum size.

A mitigation approach is proposed in [32]. The authors present a recursive algorithm that can reliably locate resources in the presence of malicious peers. Although the evaluation of the algorithm's performance yields very high accuracy in locating resources, the algorithm is protocol dependent. In addition, no removal or propagation of information about malicious peers from benign peers RT is provided.

2.1.5. *Attack assessment*

In [9], the authors launch an RT poisoning attack on DHTs by attacking nodes close to the victim. During the search process, malicious nodes were able to intercept and thus, manipulate the replies. Similarly, in [12, 13], the authors implement an RT pollution attack in P2P kAD networks, a Kademlia-based network [33], via allowing malicious peers to manually select keys that match the key of the victim. Consequently, malicious peers receive lookup requests directly and in turn falsely convince the lookup initiator to trust their replies. Nevertheless, both attack mechanisms address only convergent approaches and no practical removal or mitigation techniques were proposed.

In [14] the authors propose an RT poisoning technique based on altering the "Hello" request messages in Kademlia-based P2P networks such as KAD. However, neither a detection nor a mitigation scheme is provided. The impact of launching RT attacks on Pastry based P2P networks is provided in [34], but no further countermeasures were proposed.

Similarly, the authors in [35] evaluate the impact of several well-known attacks on the KAD network. In addition, they propose a new attack that exploits the main features of index poisoning and Sybil attacks. However, no detection or sanitizing schemes were proposed.

The authors in [36] point out the severity of attacking peers RT in DHT systems through proposing a DDoS attack to overload the key resources at the victim. Mainly, malicious peers manipulate benign peers to insert multiple entries in their RT with the same IP address of the victim which in turn flood the victim with messages. Similarly, in [37, 38], the authors highlight the severity of RT poisoning. However, in both works, no sanitizing mechanism for malicious entries were introduced.

Given the above discussion, we infer that: (i) RT attacks evolve in various contexts and are capable of severely degrading the network services causing significant impairments in the network functionalities, (ii) the absence of a generalized sanitizing mechanism that does not require central coordination. Accordingly, this highlights the importance of designing a generalized sanitizing mechanism that relies neither on a protocol specific parameter nor on a central coordinating entity.

3. System Model

This section presents the system model used for the evaluation of our approach. Utilizing the established models from [18, 39], it consists of an overlay model along with a P2P protocol abstraction that includes descriptions of the lookup mechanism.

3.1. *Overlay Network Model*

The network is modeled as a directed graph $D = (P, E)$. P is the set of peers $p \in P$ in the overlay network. Distinct peers $p, q \in P$ that maintain a neighbor relationship are represented by $e = (p, q) \in E$.

We further partition P as follows: benign peers B, malicious peers M and victim peers V, so that $P = B \cup M$, where $B \cap M = \emptyset$, $V \subseteq B$ and $N = |P|$, where N is the overlay size. Malicious peers $m \in M$ refer to peers being controlled by an attacker and may behave maliciously. Peers targeted by the attacker are victims $v \in V$. Furthermore, malicious and victim peers do not churn, which in fact gives the attacker more control over the available resources.

Peers $b \in B$ show benign behavior in the network, i.e., according to the P2P model specification and no adverse intentions. Poisoned peers $o \in O$ refer to benign peers that store or propagate malicious information as a consequence of contacting malicious peers, where $O \subseteq B$. Churning peers $c \in C$ refer to peers that leave the network either randomly or according to a certain distribution. As only benign peers, except victim peers, experience churning behavior, $C \subseteq B$ and $V \cap (O \cup C) = \emptyset$.

3.2. P2P Protocol Model

Our abstraction for structured P2P protocols consists of five salient aspects as detailed below.

3.2.1. Address Space

Peers have a unique assigned identifier referred to as the peers' *keys*. Typically, keys are generated from an external feature such as the IP address, MAC address, a serial number, or a random number. Keys usually have a length of $w \in \{128, 160, 192\}$ bits and are mapped onto the overlay's *address space* which is used to address resources such as peers and addressable data tuples.

3.2.2. Distance Function

A distance function is defined for peers on the address space. The distance notion is an important feature for many peer operations and the choice of the distance function differs among P2P protocol implementations. For example, Kademlia [33] makes use of the XOR operation to calculate the common prefix length (CPL) using the bit-string representation of the keys from two peers.

3.2.3. Routing Table (RT)

Each peer maintains an RT that contains contact information about neighboring peers. Contact information is a tuple that relates keys of peers with their underlay network information (e.g., IP address and port number). Routing tables vary among protocols and usually store k contact information tuples of peers in w lists for distance ranges $[2^i, 2^{i+1})$ with $i = 0 \ldots w-1$, and k constant. In order to resolve new contact information a lookup call is initiated.

3.2.4. Lookup Mechanism

In case the destination peer p_v for a specific message to be sent by peer p_i is not stored in p_i's routing table, a lookup call is initiated to *resolve* p_v's contact information. To initiate a lookup, p_i selects α peers from its RT to query them about p_v. We now describe the two main lookup mechanisms used in structured P2P overlays.

1. A commonly applied design best practice are *convergent lookups*, i.e., peer p_i selects a set of known peers with closest possible distance to p_v, and iteratively queries each of them to either return the contact information or to repeatedly *forward* p_i's lookup request to even closer peers until p_v can either be resolved or the lookup is dropped due to a timeout. Due to the structured nature of the overlay, convergent mechanism guarantees low message overhead with minimum number of hops for resolving a certain lookup. Nevertheless, selective placement of malicious peers in a very close distance to the victim eclipses the victim's existence as evaluated in our previous work in [40].
2. We proposed in [39, 21, 6], *divergent lookups* to mitigate attacks that make use of convergent mechanisms. Divergent lookups restrict the ability to contact peers close to the victim, where the notion of closeness is referred to as the peer's proximity. In [21], the PASS algorithm efficiently defines the address space range that contains peers with high probability of resolving the contact information of p_v. However, contacting peers during lookups from different address space ranges naturally results in suboptimal performance and reliability degradation. Unlike convergent mechanism which is highly susceptible to certain localized attacks, divergent mechanisms show high resiliency to such attacks while providing a comparable performance to convergent schemes.

3.2.5. Proximity

Each peer defines a proximity area, typically a proximate and sparsely populated region of the address space that is selected based on the overlay size N and the key length w. We define the proximity of a peer as the set of peers with the closest distance to this peer, and subsequently stored in its RT.

4. Routing Table Poisoning (RTP)

In this section, we present the fundamentals of launching an RTP attack that targets inserting and propagating malicious entries in benign peers RT. The proposed attack model constitutes the basis for evaluating the proposed sanitizing mechanism.

In order to validate the effectiveness and applicability of the sanitizing mechanism in various RTP attack scenarios, we consider a sophisticated general attack model that (i) is not only applicable for a specific P2P protocol and topology, (ii) does not target a specific victim or (iii) considers various attacker capabilities and adversarial behaviors that represent severe attack scenarios.

First, we state the attacker's target. Second, the attacker's capabilities in terms of the available malicious resources and the placement criteria are described. Finally, the adversarial behaviors of inserted malicious peers that allow for poisoning peers RT are discussed.

4.1. RTP Attacks Types and Targets

RTP attacks are launched by allowing malicious peers to intercept lookup requests. The RTP attack's target is defined based on the intercepted lookup request's destination, i.e., malicious peers behave adversarial or not when intercepting a given lookup. RTP attacks are launched either to generally cause perturbations to the overlay (undirected RTP) or to hide the existence (directed RTP) of a specific data or peer, referred to as directed RTP. Both attack targets are described below.

Directed attacks

In case of directed RTP, malicious peers target only a certain victim set V such that $|V| < |B|$. Mainly, the selected victim set are those peers with critical data or popular content. The main target of malicious peers, inserted in the overlay, is to poison entries that point to a targeted victim set [41, 35, 18].

As the focus of this work is to assess the efficiency of the sanitizing mechanism under severe attack conditions, we do not focus on directed attacks for the following reasons: (i) a major fraction of RT entries is not poisoned. Hence, the validity of the evaluation of the sanitizing mechanism is affected as the target of the proposed sanitizing mechanism is to remove malicious peers from RTs regardless of their targeted entries to poison and (ii) directed attacks coerce specific adversarial behaviors and thus, are not suitable to assess the generality of the sanitizing mechanism. For these reasons, we do not consider directed attacks in our work.

In [18], we highlight how the proposed sanitizing mechanism against directed attacks stems from the general form of the attacks proposed in this work. We study and highlight the modifications needed to launch such attack and the relative modifications in the sanitizing mechanism. Furthermore, we evaluate the impact of such attack along with the effectiveness of the proposed eviction mechanism.

Now we discuss the challenges that arise from continuously attacking all possible peers and the specific adversarial behavior executed by malicious peers.

Undirected attacks

The target of undirected RTP attacks is to poison benign peers RT where no specific victim is targeted. In this case, $B = V$.

Unlike directed RTP, a major fraction of the benign peers RT is poisoned as malicious peers do not only target specific entries to poison. As this attack target shows more severity and thus, is suitable for evaluating the performance of the sanitizing mechanism, we focus on undirected RTP as the attack's target. The target of undirected RTP attacks is to intensively pollute benign peers RT ($b \in B$) via blocking, altering or diverting lookup requests.

4.2. Attacker Capabilities

Now we detail the attacker's capabilities to launch an RTP attack. Capabilities refer to the amount of available malicious resources and the placement of malicious peers according to the selected undirected RTP attack.

Malicious resources

As the amount of malicious resources inserted in the overlay increases, the perturbations that can be imposed on the overlay also increase. In order to validate the performance of the sanitizing mechanism in severe attack scenarios, we assume the attacker is capable of inserting various amounts of malicious peers up to 20% of the whole overlay size. The malicious insertions are done by inserting new peers into the network or hijacking existing peers.

Malicious placement

The placement of malicious peers mainly depends on the lookup forwarding approach specified in the P2P protocol. Hence, we start by defining the lookup approach and the matching placement criteria.

In order to validate the applicability of the sanitizing mechanism for various P2P protocols, we make use of a lookup approach that does not impose any specific criteria or route for forwarding lookup requests, i.e., to increase the attack's impact via allowing maximum interception of lookups when malicious peers are inserted. Therefore, the divergent PASS lookup approach from our previous work in [21] is usable here. The reason for selecting divergent PASS is that the peers get randomly selected within a specific address range for forwarding lookup requests.

Accordingly, the malicious peers are randomly placed within the specified range. In turn, lookup requests are equally probable to be intercepted by a benign or a malicious peer depending on the amount of inserted malicious peers.

4.3. RTP Adversarial Behaviors

To emphasize the impact of the attack, we propose a variety of adversarial behaviors, where malicious peers are capable of dynamically altering the actions taken according to the attacker's resources and target. We note here that since a lot of the existing research addresses the problem of join-leave attacks such as [30], we consider this attack behavior out of scope of this paper.

Malicious peers attack the lookup mechanism by intercepting lookup requests and hence, replying with malicious information which affects the lookup reliability, integrity and confidentiality. Consequently, malicious information propagates to benign peers RTs causing an RTP. Once a malicious peer $p_m \in M$ successfully intercepts a request, p_m replies with an *Fake Reply (FR)* as a resolving address to the lookup request.

In an FR, p_m inserts the contact information of another colluding malicious peer which claims to hold the key of the lookup destination that p_i is requesting and falsely convinces p_i that the request was successfully resolved. As a result, $p_i \in B$ updates its RT with the newly received entry which allows malicious information to propagate through the overlay and thus, poison benign peers RT. We now define the content and the frequency of generating an FR reply.

Generating False Replies (FR)

Malicious peers are assumed to always reply with colluding malicious information to the lookup initiator. This adversarial behavior is chosen when the attacker's main target is to propagate malicious routing information regardless of the detection likelihood. This denotes that, in case of malicious peers generating false replies based on a certain probability, the detection and thus, the sanitizing of the overlay would require longer time.

Nonetheless, the perturbations effect on the overlay will be relatively less compared to the perturbations caused by malicious peers continuously lying. We refer to our previous work in [39] where the detection accuracy in case of randomly lying malicious peers is evaluated.

Note that another reason for choosing this behavior is that the attacker may target fast spreading of perturbations in the overlay. Such case can occur when the attacker owns enough resources and the attack is time dependent, i.e., the attacker's aim is to launch the attack in a specific time period or during a time triggered event in the overlay. Hence, as our focus is to evaluate the effectiveness and the applicability of the sanitizing mechanism in drastic attack scenarios, we assume that malicious peers always generate fake replies.

False Reply (FR) Content

Malicious peers control the content that should be sent in a fake reply. The possible FR content can potentially cover:

1. Replying to all intercepted lookups with a **single** malicious lookup reply. Such behavior is executed when a specific information needs to propagate through the overlay. Nevertheless, such adversarial behavior makes p_m more susceptible to detection.
2. Replying with **different** malicious replies. As malicious peers collude, p_m can reply to the lookup request with selecting one of the malicious peers that p_m is aware of. Such behavior is deployed by p_m when seeking general perturbations. In addition, replying with different malicious fake destinations further complicates the detection process.

Further details about the detection procedure of the consistent malicious replies are provided in Section 5.

5. Detection Mechanism

We now introduce the detection mechanisms used locally by each peer to suspect other peers based on the received lookup replies. In order to detect lookup inconsistencies, we propose a modified lookup mechanism in [39] where peers are able to gather more than a single reply.

The lookup initiator can detect inconsistencies through comparing the set of received replies according to (i) the consent of the replying peers' location with the lookup protocol specifications, (ii) the average number of hops experienced by the lookup reply compared to the recorded average from previous lookups and (iii) the returned contact information in the lookup reply. Consequently, peers are suspected when violating these criteria. The most important feature in our detection mechanism is comparing replies, i.e., peers are also suspected when replies are not identical which are detected through a certain feature in the detector discussed through this section.

Originally in any given lookup mechanism, the lookup is terminated when receiving the first reply that contains the requested information about a given peer p_v. This coerces the lookup initiator p_i to accept the lookup result without being able to validate the results since only a single reply is considered. As a result, whenever a malicious peer receives the lookup request and replies with a fake reply, p_i accepts the reply which results in poisoning p_i's RT with a malicious entry. The modified lookup mechanism is discussed below which provides the operations that allows p_i to gather a set of lookup replies from different peers.

5.1. Modified lookup Approach

We first outline the drawbacks of the contemporary lookup mechanisms, which highlights the motive of using a modified lookup mechanism. Subsequently, the operations of the modified lookup approach are presented.

The drawback of existing lookup approaches

In prior lookup implementations, p_i picks α candidate peers from its RT to start forwarding a lookup request for p_v's contact information, where α is a lookup specific parameter for the maximum number of parallel requests that can be sent. Once the lookup request is received by peer p_r, it replies with p_v's contact information if p_r has an entry for p_v in its RT. Otherwise, p_r inserts a list of potential candidates that, according to the lookup specification, have a high chance of owning p_v' contact information in their RT. Iteratively, p_i initiates α new requests from this list. Finally, the look up process terminates immediately once p_v's address is resolved or i_{max} iterations are reached.

Such approach coerces p_i to accept the single received reply. This means that p_i has no comparing base to validate the received reply, i.e., no multiple replies to enable p_i to detect inconsistencies. Hence, malicious peers can misuse this approach to falsely terminate the lookup without being detected while simultaneously resulting in a very low message overhead for the lookup approach.

Modified lookup operations

To obviate the above mentioned drawback, we proposed a lookup modification in [39]. As depicted in Figure 1, the major extension in the lookup modification is that the lookup process is not terminated till: (i) R replies containing p_v's contact information are received or (ii) i_{max} iterations are reached, where in this case, $R \leq \alpha$. Consequently, p_i is able to compare multiple replies from different peers and thus, detects inconsistencies to suspect certain peer(s). Note that, in this context, a suspected reply refers to a suspected peer as replies are only accepted from distinguished peers, i.e., no peer can provide more than a single reply.

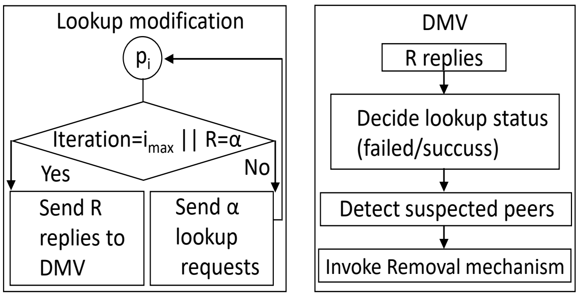


Figure 1: Detection procedures.

For this purpose, we make use of a Dynamic Majority Voter [42] (DMV) to process the R received replies. Now we discuss the technical aspects of the DMV, for more details we refer to our work in [39].

5.2. Dynamic Majority Voter (DMV)

Firstly, the suspected peers reported by the lookup modification are added to the set of suspicious peers S. Second, the remaining unsuspected peers are processed as inputs to the DMV. Afterwards, the DMV decides whether a valid (non-empty) majority of identical replies exists or not.

The DMV declares the lookup status as successful if such majority is found. Consequently, p_i stores the corresponding contact information. In case the lookup is declared to be unsuccessful, p_i initiates a new lookup to resolve p_v. To sum up, we provide the acceptance cases for the DMV:

1. $R \geq 3$ and a valid identical majority of the replies exists.
2. $R = 2$ and both replies are valid and identical.
3. $R = 1$ and the reply is valid.

Notably, the unmatched minority is added to the suspicious list, i.e., the remaining set of replies that did not constitute the majority. Finally, once the list of suspected peers is announced by the detector, the sanitizing mechanism is triggered to further investigate and thus, sanitize the overlay through removing malicious peers as discussed in the next section.

6. Sanitizing Mechanism (SM)

In this section we present the sanitizing mechanism. Prior to the operation of SM, the detector proposes a set of suspected peers according to their lookup replies as discussed in Section 5. Afterwards, the SM is invoked to investigate and thus, reach a decision about the suspected peers. Consequently, the SM executes a removal procedure for suspected peers identified malicious to sanitize the benign peers RT.

Unlike the detector which is operated locally by each peer, the SM is executed as a distributed quorum to reliably investigate and propagate information about malicious peers. Such propagation further accelerates the sanitizing rate for the whole overlay. Hence, the SM results in a stable and reliable P2P service provision.

We note that the quorum needs to be obtained in the decentralized P2P environment and in the presence of malicious peers. Accordingly, Byzantine resilient SM procedures are developed.

The sanitizing mechanism is constructed from four main procedures where each procedure is illustrated in the following subsections. For consistency, Table 1 provides a list of the variables and annotations used through the rest of this section.

The first procedure (Forming a quorum) defines how p_i can create a quorum, quorum size dependabilities and quorum members constraints. The second procedure (Quorum investigation) describes how the quorum investigates p_s's behavior. Afterwards, the third procedure (Reaching an agreement) handles messages exchanging and reaching a coordinated decision between p_i and the quorum peers. Finally, the removal procedure (Malicious removal) is invoked to sanitize benign peers RT from peers identified to be malicious as illustrated in Figure 2.

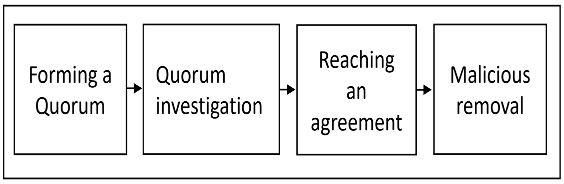


Figure 2: SM procedures blocks.

6.1. Forming a Quorum

As shown in Figure 3a, once the detector suspects certain peers, p_i starts the sanitizing mechanism by forming a quorum. In the following, we describe the criteria for choosing a quorum size along with the constraints regarding joining peers' types raised by the existence of malicious peers in the overlay. Afterwards, the formation mechanism which describes the messages exchanged while forming a quorum and the acceptance-rejection criteria for joining a quorum is described.

Table 1: Acronyms description

Variable	Description
p_s	Suspected malicious peer
p_i	Quorum initiator
p_v	Denotes the victim peer
Q	Quorum
p_q	Peer p in Q
n	Number of peers in Q
Q_B	Set of benign peers in Q
Q_O	Set of poisoned peers in Q
Q_C	Set of churning peers in Q
Q_M	Set of malicious peers in Q
MR	Monitoring Request
RT	Routing Table

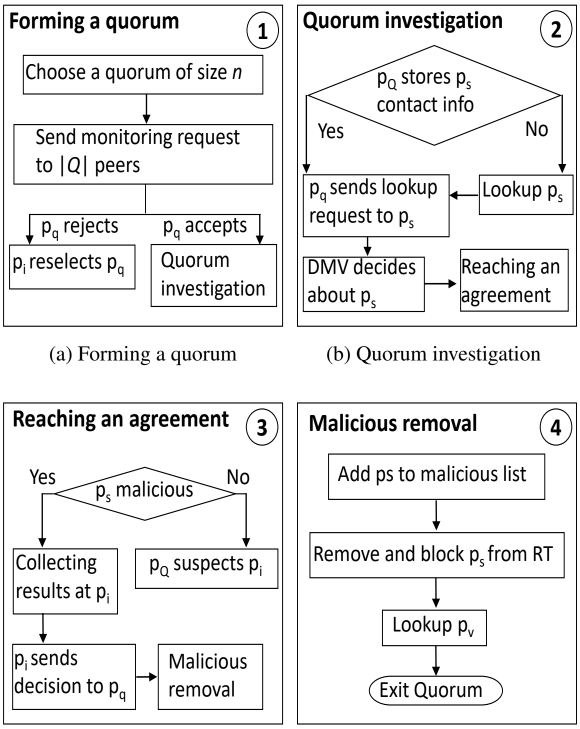
6.1.1. Quorum size and members selection

Initially, p_i selects n peers from its RT where selected peers are uniformly distributed according to their distance in p_i's RT. This selection approach guarantees peers from all possible distances participate in investigating p_s which accelerates propagating information about peers identified malicious in the complete address space.

The generality of the mechanism allows to choose any distribution according to the already used lookup approach. In our experiments we use PASS, characterized by an address space divided into regions according to the common prefix in the peers keys. Within the lookup forwarding mechanism, peers forward the lookup request to other peers located in the same address space region.

In order to propagate such information to all regions, we make use of the uniform distribution selection of quorums, so that all peers in different regions are updated with the correct information and can propagate it when requested from peers within the same region. This demonstrates that, the sanitizing mechanism is adjustable according to the basic information about the used lookup approach in the overlay.

As p_q is a member of Q, p_q belongs to one of the four



(a) Forming a quorum

(b) Quorum investigation

(c) Reaching an agreement

(d) Malicious removal

Figure 3: Technical aspects of SM procedures.

sets listed in Table 1. Hence, n can be defined as:

$$n = |Q_{B\setminus O\cup C}| + |Q_{O\setminus C}| + |Q_C| + |Q_M| \tag{1}$$

where $|Q_{B\setminus O\cup C}|$ denotes benign peers that are not poisoned and do not churn. $|Q_{O\setminus C}|$ refers to the number of poisoned peers that do not churn.

In order to assure reaching a reliable decision in Q, the size of the quorum n must follow certain constraints regarding the maximum number of malicious peers that can exist in the quorum. Based on Practical Byzantine Fault Tolerance (PBFT) and Byzantine Agreement [43, 44, 45, 46, 47], the number of peers in a decentralized system that is capable of maintaining a correct state of the system is bounded by $n \geq 3f + 1$, where f is the amount of faulty peers in the system. As f peers can be malicious and another f can be poisoned or churning, this adds up to $2f$. This means that at least $f + 1$ should exhibit: (i) no malicious behavior (neither malicious nor poisoned and (ii) alive (not churning) in order to maintain a correct system state.

Similarly, the selected quorum is capable of providing reliable results in case the number of non-poisoned non-churning benign peers $|Q_{B\setminus O\cup C}|$ outnumbers the rest of other selected peers, therefore:

$$|Q_{B\setminus O\cup C}| > |Q_M| + |Q_C| + |Q_{O\setminus C}| \tag{2}$$

As these peers can deviate the quorum's decision according to the following possible actions:

1. $|Q_{O\setminus C}|$ peers such as $p_Q \in O \setminus C$ may provide malicious replies due to acquiring a poisoned entry from other malicious peer.
2. $|Q_M|$ where $p_Q \in M$. These peers behave maliciously via sending fake replies (see Section 4) to divert votes majority.
3. $|Q_C|$ peers where $p_Q \in C$ that initially accept to join the quorum. However, although these peers are neither malicious nor poisoned, they are subject to churning out and thus, do not respond to p_i.

Accordingly, the quorum size n that is capable of reliably investigating and thus, correctly reaching an agreement about suspected peers is constrained to:

$$n \geq |Q_M| + |Q_C| + |Q_{O\setminus C}| + |Q_{B\setminus O\cup C}| \tag{3}$$

Such constraint guarantees that the quorum's majority votes about p_s are benign, neither poisoned nor churning as depicted in Figure 4. In Section 7, we validate how such constraints hold conveniently given the existence of a remarkably high fraction of malicious peers of up to 20%. Now we describe the quorum formation mechanism in terms of messages exchanged and joining acceptance/rejection criteria.

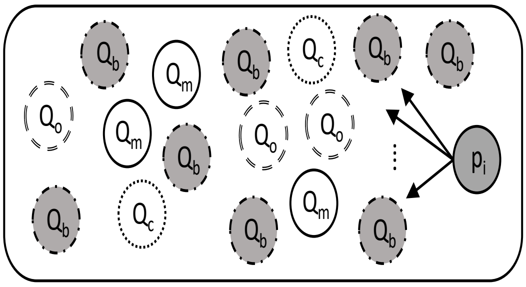


Figure 4: Example of types of peers in Q.

6.1.2. Formation mechanism

Initially, p_i sends a monitoring request message to p_q as depicted in Figure 3a. A monitoring request contains the contact info of each p_s. As the DMV at p_i might suspect multiple peers at the detection phase, p_i encapsulates all suspected peers in a single monitoring request which allows for the following:

1. Avoid excessive quorum formations which remarkably decreases the messages exchanged by the quorum's peers during executing the SM.

2. Increase the cost of generating fake quorums by malicious peers since the probability of detecting such adversarial behavior increases. This is due to the ability of p_q to suspect p_i after investigating the suspected peers. Hence, the SM will be invoked against p_i, more details are provided in (Malicious Removal) procedure (see Section 6.4).

Once p_q receives the monitoring request, it can either accept and proceed to the next step, or reply with a rejection to p_i. Rejections could be due to: (i) being involved in another quorum, (ii) loaded traffic or (iii) malicious behavior to limit quorum formations. In case p_q agrees on joining the quorum, it starts executing the quorum investigation procedure which is described below.

6.2. *Quorum Investigation*

As illustrated in Figure 3b, the target of this procedure is to allow each p_q to monitor p_s's behavior and thus, confirm or deny p_i's suspicion about each p_s. After receiving a monitoring request from p_i and accepting joining Q, the investigation procedure is triggered. The idea of this procedure is to check whether p_s is manipulating requests or not. Therefore, p_q monitors p_s's behavior via requesting information from p_s that p_q can validate from p_s's reply.

Nevertheless, the investigation procedure should be conducted seamlessly to obviate p_s from acting benignly once it detects being monitored. The two main aspects that define the procedure's anonymity are: (i) the type of messages p_q sends when monitoring p_s and (ii) the content requested from p_s in such a message. Both factors are detailed below.

6.2.1. *Messages Exchange*

Communication between p_q and p_s must be handled seamlessly as p_s can divert the monitoring process once it detects being monitored by p_q, which can remarkably impact on the sanitizing process.

For this reason, p_q makes use of regular lookup messages defined by the P2P protocol to avoid being detected by p_s. This means that p_q initiates lookup request to α peers, including p_s. Afterwards, p_q processes the replies to the detector to detect inconsistencies in p_s's reply.

Nonetheless, p_s can detect being monitored when it receives multiple lookup requests requesting only p_v's contact information, i.e., assuming p_s collect statistics of how frequently it has been requested about each peer. Hence, the content of the lookup request should not reveal the victim's identity which is discussed next.

6.2.2. *Messages content*

As the RTP attack proposed in Section 4 targets maximizing RT perturbations, malicious peers are assumed to reply with fake replies regardless of the lookup key in the lookup request. Hence, in order not to exceed p_s's expected average of receiving lookup requesting only p_v's contact information, p_i attaches a list of peers in addition to p_v in the monitoring request. Consequently, p_q can assign any peer in the list as the lookup destination key. Using the detection mechanism discussed in Section 5, p_q decides whether to confirm or deny p_i's suspicion about p_s.

To sum up, the investigation procedure assures that:

1. p_i keeps the identity of the victim p_v anonymous to the quorum, which in turn prohibits malicious peers that may exist in the quorum from colluding against p_v.
2. p_s cannot predict or estimate being monitored as it receives normal lookups requesting different contact information.
3. p_q can validate its decision about p_s through the detection mechanism.

Adjusting SM: we note that our approach does not depend on a certain attack or lookup mechanism, i.e., the approach is adjustable according to the basic information such as the used lookup approach and the overlay topology. For example, in case of the BitTorrent protocol where only one NodeID is considered, the detection is applied based on: (i) the replies consistency about the destination ID, (ii) the average number of hops and (iii) the visited nodes compliance with the lookup forwarding criteria towards this single node. Afterwards, the detector applies the DMV to decide about the suspicious replies. Clearly, in case the fraction of malicious peers is larger than the benign ones, the detection can be useless. More details regarding this case are provided in [39].

6.3. *Reaching an Agreement*

As depicted in Figure 3c, once p_q decides locally about each p_s, the decision is forwarded to the quorum initiator p_i along with a time stamp identifying the reply time of p_s. Note that p_q forwards its local decision about p_s directly to p_i instead of mutually exchanging decisions with other peers in Q. Hence, p_q have no information about the other participating peers in Q. Obscuring the identity of Q's members from each other yields:

1. Decreasing the probability of malicious colluding against benign peers in Q.

2. Preventing manipulation of the results aggregation procedure as malicious peers might intercept the aggregated results messages.

Simultaneously, p_i waits for a timeout t_{max} to receive all the monitoring replies. Otherwise, in case the waiting time exceeds t_{max}, p_i proceeds with the set of received monitoring replies. t_{max} is set according to the average time a lookup process consumes plus adding a guard time for considering the quorum formation time which is an application configurable parameter.

Afterwards, p_i inputs the set of received replies to the DMV to decide about the majority of the replies, i.e., whether the majority of the replies suspects p_s or not. Based on the DMV output, p_i reaches a decision about p_s and accordingly, inform each p_q about its decision. Now the possible decisions reached by p_i about p_s along with the consequent actions are discussed.

6.3.1. p_s is poisoned

p_i checks the time stamp encapsulated in each p_q monitoring reply in order to differentiate whether p_s is malicious or was poisoned. This is done through checking the time stamps sequence for each reply. This means in case p_s starts to consistently provide correct information about a given lookup key till the last time stamp for the same lookup key, it's considered to be poisoned. Otherwise, if p_s consistently replying, or alternating, with fake information, p_s is considered malicious.

6.3.2. p_s is malicious

Two cases lead p_i to consider p_s as malicious. First, if all the monitoring requests regardless of their time stamps report suspicious about the replies provided by p_s. Second, in case p_s's replies show alternating behavior. Alternating behavior refers to providing fake replies after providing correct ones which can be inspected from the time stamps of the received replies.

6.4. Malicious Removal Procedure

The agreement reached by the quorum allows p_i to determine whether to proceed with removing p_s from its RT or not. Thus, we now discuss how p_i proceeds according to the DMV decision.

6.4.1. Suspicion confirmed

If p_i decides that p_s is malicious, p_i executes the following **Malicious Removal (Mal-Rem)** steps as depicted in Figure 3d.

1. p_i removes p_s from its RT and blocks any further contact with p_s.
2. p_i initiates a new lookup to search for the correct contact information of p_v.
3. p_i sends the decision to all peers in Q.

Once each p_q is informed that the quorum's decision confirms that p_s is malicious, p_q performs one of these actions depending on its local decision about p_s.

1. In case p_q's local decision also confirms the adversarial behavior of p_s, p_q executes the (Mal-Rem) steps.
2. If p_q's DMV decision about p_s' reply is benign, p_q suspects p_i and initiates a quorum against p_i.

Since p_i is aware of the conflict between both decisions, p_i expects to be monitored by a new quorum. Accordingly, in case p_i is benign, p_i replies consistently to all the monitoring lookups so that p_q detects p_i's benign status. Afterwards, p_q initiate a lookup request to p_s to investigate it's status without launching new quorums.

Without such a scheme, malicious peers will be able to invoke the sanitizing mechanism towards benign peers which can severely affect the network stability through (i) eclipsing p_v by initiating a malicious majority quorum or (ii) exhausting the network bandwidth and overload peers with exchanging messages via joining fake quorums. Such countermeasure prevents malicious peers from starting fake quorums as consequently, such malicious behavior will re-trigger the sanitizing mechanism towards these malicious peers.

Moreover, encapsulating multiple peers in monitoring requests forces malicious peers to behave benignly to deviate being suspected. Otherwise, such malicious behavior will be obviously detected when p_s sends multiple fake replies at once to p_q. Thus, malicious peers are forced to refrain from initiating fake quorums.

In Section 7, we investigate the impact of turning poisoned entries in the quorum to benign and how propagating the correct information about the victim(s) helps in sanitizing the overlay and increase the isolation of malicious peers.

6.4.2. Suspicion declined

In case p_i decides that p_s was poisoned, after processing the monitoring replies through the DMV, p_i initiates lookups to p_s requesting the contact information of the same monitoring set that was initially sent to the quorum members. Note that p_i selects distinctive time slots to initiate such set of lookups to p_s. Therefore, p_s cannot detect any abnormal behavior from p_s due to receiving excessive lookups or multiple lookups from p_i requesting p_v's contact information. Afterwards, according to p_s replies, p_i executes one of the following steps:

1. If p_s replies correctly, p_i trusts p_s and thus, inserts p_s and any reply provided by p_s in the future into its RT.
2. In case the DMV reconfirms suspicion about p_s, p_i suspects that the majority of peers in Q is malicious. To that end, p_i creates a new quorum to re-monitor and subsequently unveil p_s, after restricting peers in Q from joining the new quorum.

In Section 7, we evaluate the likelihood of forming a quorum with malicious majority and validate the constraints regarding forming a quorum along with evaluating the effectiveness of the proposed sanitizing mechanism.

7. Evaluation

This section assesses the effectiveness of SM as a countermeasure for RTP attacks launched with the set of the proposed adversarial behaviors. The target is to evaluate the severity of RTP attacks on the overlay's reliability and the imposed perturbations resulting from poisoning RT entries. Consequently, SM is evaluated in terms of reliability enhancements, imposed overhead on the network and malicious removal ratio from benign peers RT.

In order to do so, two experiments are conducted. The first experiment **(RTP attack severity)** evaluates the impact of launching RTP attacks on structured P2P overlays while focusing on how these attacks severely degrade the overlay performance. The second experiment **(Sanitizing mechanism influence)** is conducted to assess the effectiveness of SM on sanitizing overlays during launched (undirected) RTP attacks while analyzing the correlation between different SM parameters.

First we start by introducing the simulation environment, metrics and parameters used for evaluation. After that, we introduce each case study with related discussion and results interpretation. Finally, we provide a detailed summary about the results that highlight the effectiveness of SM.

7.1. Simulation environment

Experiments were conducted using the OMNeT++ simulator [48] and OverSim [49]. OverSim provides the OMNeT++ simulator with various P2P protocol implementations. For Average-Min-Max calculations, each simulation is scheduled for 10 repetitions. Moreover, for results validation, each simulation duration was scheduled to 4 hours runtime.

In Table 2, the simulation parameters used in the experiments are provided.

Table 2: Simulation parameters

Parameter	**Value**
Maximum iterations i_{max}	10
Maximum replies number α	7
Key length w	128
Lookup approach	Iterative
Malicious insertion ratio `MI`	$10\%, 15\%, 20\%$
Overlay size N	$2.5k, 5k, 10k, 20k, 30k$

7.2. Simulation Model

In order to validate the reliability and the scalability of the sanitizing mechanism, the experiments were conducted for different overlay sizes $N = 2500, 5000, 10000, 20000, 30000$. Moreover, high values of malicious peers $|MI| = 10\%, 15\%, 20\%$ were injected in the overlay. MI is used to measure the impairments caused in the network given a certain amount of inserted malicious resources.

We now provide a description of the system workload and churn distributions used in our simulation model.

7.2.1. System Workload

In order to evaluate the experimental results for undirected RTP attacks where every peer is a potential victim, we use a "Fully Distributed Application (FDA)" workload where every peer initiates lookups requesting the contact information of randomly selected peers. Lookups are sent on average every 10 seconds with a standard deviation of 5 seconds.

7.2.2. Simulation Churn Models

We use a **Pareto (P-500)** churn model to simulate the churning rate of benign peers. Using the Pareto (P-500) distribution, peers acquire average life-time and dead-time of 500 seconds. Pareto distribution provides realistic experimental results for real P2P overlays scenarios [50].

7.3. Evaluation Metrics

For both experiments, the following performance metrics are used.

- Lookup Success Ratio (`LSR`) measures the average ratio of successful lookups over all lookups destined to random peers. `LSR` provides insights about the reliability of the P2P overlay in both experiments.

- Message Complexity (`MC`) evaluates the average message overhead on the network per lookup. `MC` also assesses the message overhead imposed by the sanitizing mechanism.

- Malicious ratio per RT (MRT) provides the average ratio of malicious entries poisoned in benign peers RT as a result of the existing malicious resources in the overlay.

7.4. Experiment 1: RTP attack severity

The target of this experiment is to assess the severity of undirected RTP attacks with the proposed adversarial behaviors and to highlight the reliability degradation of the system's service provision. Results are evaluated based on LSR, MRT, MI and MC. As in undirected RTP attack, where all peers are potential victims, data is collected whenever a lookup is initiated regardless of the lookup key. We continue with describing the experimental results depicted in Figures 5 through 8, and we conclude each case study with a detailed interpretation of the results.

7.4.1. Discussion of the results

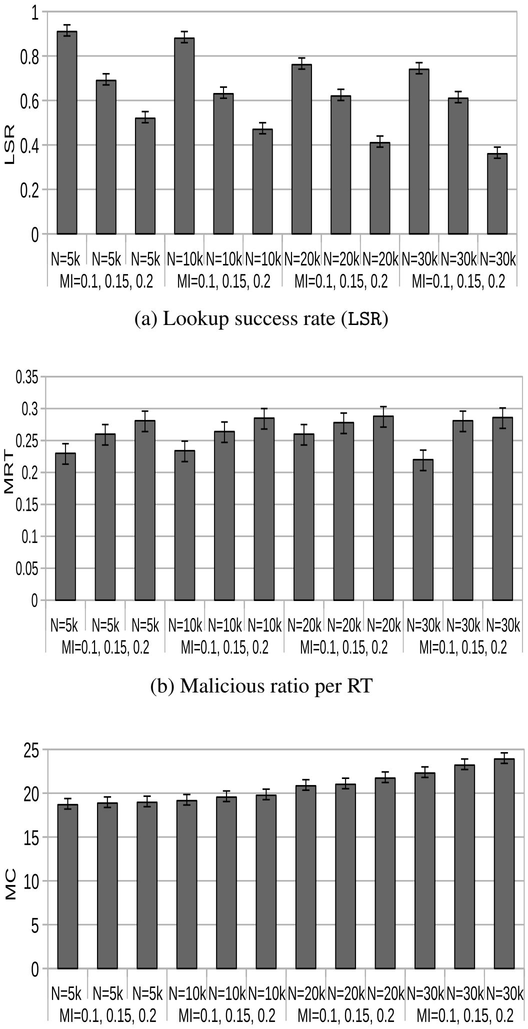
Figure 5a shows the LSR values across different overlay sizes $N = 5000, 10000, 20000, 30000$ with $MI = 10\%, 15\%, 20\%$ in order to assess the correlation between inserting different malicious peers, the perturbations level in peers RT entries and the hosting overlay size. As depicted in Figure 5a, LSR remarkably decreases when MI increases as the number of malicious peers that intercept lookup requests increases. LSR values range between 36% and 91% for $N = 5000, MI = 10\%$ and $N = 30000, MI = 20\%$, respectively.

In Figure 5b, the average number of malicious peers inserted into benign peers RT is calculated through measuring the percentage of malicious entries to the total number of entries per RT. Note that each entry represents a distinct peer, i.e., the same peer can not exist in multiple entries. As shown, MRT increases when the percentage of malicious peers inserted into the overlay increases from 10% to 20% where the average MRT ratios range between 24% to 31% regardless of the overlay size.

The average message overhead complexity (MC) using divergent lookups is depicted in Figure 5c. MC measurements show comparable values across different overlay sizes and MI ratios, where MC ranges between 18 and 24 message per lookup.

7.4.2. Interpretation of the results

The significant decrease in LSR occurs due to malicious interception of the lookups. Due to the effect of RTP adversarial behavior, malicious peers reply with fake replies to p_i. Consequently, such behavior results in a non successful lookup according to the DMV decision due to the depicted majority resulted from α malicious replies.



(a) Lookup success rate (LSR)

(b) Malicious ratio per RT

(c) Message Complexity (MC)

Figure 5: RTP attack impact.

The trend depicted in Figure 5a is that LSR decreases when MI increases as seen in different values of MI within the same N. Hence, at the same malicious insertions MI values and different N such as (combinations of $\{MI, N\}$): $\{MI = 0.15 \wedge N = \{10k, 20k, 30k\}\}$, $\{MI = 0.1 \wedge N = \{5k, 10k\}\}$ and $\{MI = 0.1 \wedge N = \{20k, 30k\}\}$ the values are closely comparable to each other due to the impact of the same fraction of malicious peers on the overlay size N.

Further degradation of the success ratio occurs when poisoned peers advertise for malicious entries whenever they are queried about a lookup destination that points to a poisoned entry in their RT. In addition, as benign peers RT is polluted with malicious entries, the probability of querying malicious peers during lookup iterations increases. Hence, the corresponding ratio of benign peers queried decreases, which in turn significantly impacts the LSR.

In Figure 5b, perturbations in benign peers RTs due to malicious peers insertion can be inferred as $MI = 10\%$ results in more than 24% poisoned malicious entries in benign peers RT. Moreover, slight increases of MRT causes significant perturbations as for $MRT = 31\%$, 64% loss rate at $N = 30$ is depicted. Although at $MRT = 22\%$ the loss rate is 26% at $N = 30$. Hence, with a few resources, RTP attacks are able to cause substantial impairments in the overlay.

Figure 5c, MC shows moderate messages overhead per lookup despite of the degraded LSR and high values of MRT. Note that the average MC overhead due to divergent PASS averages between 10-12 messages as evaluated in our previous work in [21]. Nevertheless, due to the modified lookup approach, where the lookup is not terminated till α replies are returned, MC overhead increases to 18-24 messages per lookup. Although α replies are required instead of one reply, PASS shows a reasonable increase in MC due to forwarding lookup requests to a specific range in the overlay where peers are most likely to store the destination's lookup address.

Furthermore, RTP attacks do not introduce more messaging overhead on the overlay since malicious peers directly provide fake replies which minimizes the average messages exchanged per lookup. However, maintaining reasonable MC does not indicate high reliability for the overlay when RTP attack is launched. Although divergent lookups show high resiliency against localized attacks, such protocol specific approaches are inefficient against RTP attacks.

7.5. Experiment 2: SM Influence

In this experimental study, we evaluate the performance of the proposed SM in terms of: (i) restored reliability, (ii) malicious removal effectiveness, (iii) imposed overhead on the overlay during the sanitizing process. Moreover, we investigate the correctness and accuracy of the quorum's decision via analyzing the ratio of selected peers according to their types (benign, poisoned, churning or malicious) during quorum formation discussed in Section 6.1. For consistency, same overlay sizes and evaluation metrics are used for evaluation and comparison of the results with the previous experimental study.

7.5.1. Discussion of the results

Figure 6a shows the average LSR when SM is on. A remarkable increase in LSR is noticed compared to the first experimental study, where LSR averaged between 36% and 91%. When SM is operating, average LSR ranges from 97% to 100%.

The average MRT is illustrated in Figure 6b. MRT values average from 0.02 to 0.04 which is a significant decrease in the average amount of malicious entries per RT compared to the first case study where values range from 0.24 to 0.31.

Figure 6c depicts the average MC overhead induced on the overlay as a result of messages exchanged during executing the sanitizing procedures. As messages overhead due to SM varies with time according to the rate at which the mechanism is invoked, MC varies from 22 to 710 messages.

Although the MC values with SM running are remarkably high at the starting phase of the SM, MC decreases to show similar behavior as in experiment 1 starting at time t=2000s where messages average between 18 and 24. For better interpretation of the results, a time line of MC imposed on the overlay is provided. In addition, For comparing the variations of MC with the first experiment, both cases where SM is on and off are shown in the same figure. MC is measured at each data collection point which is scheduled every 200 seconds.

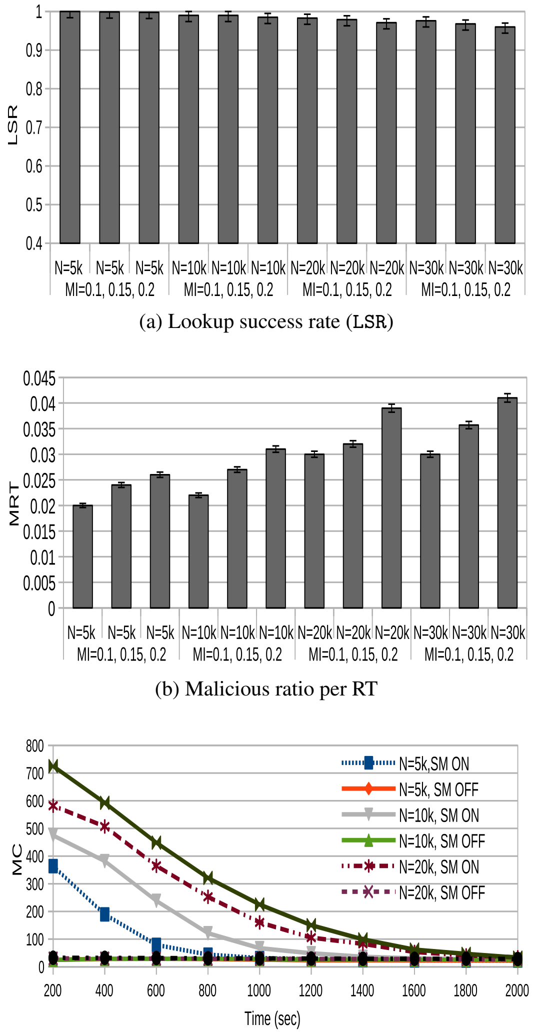
In Figure 7, the average MRT in benign peers RT is measured during SM runtime, which gives an overview about the required sanitizing time given different MI values.

Figure 8 evaluates the correctness of decisions taken by the initiated quorums as discussed in Section 6.1 which depends mainly on the types of selected peers, i.e., the ratio of benign, malicious, poisoned and churning peers selected by p_i during the quorum formation procedure.

7.5.2. Interpretation of the results

As malicious peers are sanitized from benign peers RT, the number of malicious peers contacted during different lookup iterations decreases. Hence, more correct replies are passed to the detector resulting in a remarkable increase in the ratio of successful lookups, as shown in Figure 6a.

Moreover, once p_Q receives a lookup request for p_v's contact information, p_Q replies with the correct information that do not contain a malicious entry. Consequently, the propagation of malicious entries decays, which in turn increases the availability of the correct contact information of peers. This results in increasing



(a) Lookup success rate (LSR)

(b) Malicious ratio per RT

(c) Message Complexity (MC)

Figure 6: SM performance measurements.

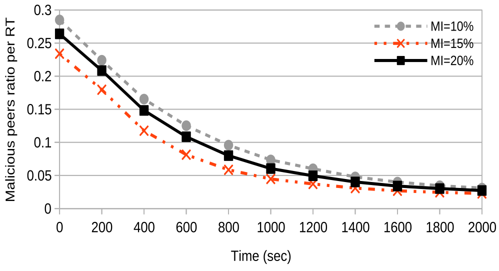


Figure 7: MRT decay due to SM

LSR which indicates full restoration of the overlay's reliability through SM.

As shown in Figure 6b, the ratio of malicious peers that exist in benign peers remarkably decreases below 5%. This is due to the effect of removing malicious entries from benign peers RTs which consequently blocks propagating malicious entries and provide correct lookup replies to other peers.

In addition, once p_s is announced to be malicious, all peers in Q block p_s which restrains any further contact with p_s. Hence, the attacker is not capable of restoring the ratio of malicious entries in benign peers RT using the same malicious resources. For large overlay sizes as in $N = 30000$, there is a lower probability that all malicious peers are contacted during lookup iterations. This is the reason MRT shows higher values (4%) than in smaller network sizes as for N=5000 the average MRT ratio is 2%.

Figure 7 provides more insights about the effectiveness of SM in decreasing MRT. Measurements are shown for $N = 10000$ and $MI = 10\%, 15\%, 20\%$. Malicious insertions are applied at the initialization phase of the network in order to evaluate the decaying rate of MRT where peers RTs are highly poisoned with malicious entries due to RTP attack. Hence, the sanitizing rate can be assessed when the attack's impact is maximized.

MRT significantly decreases below 5% even in cases when RTs are initially poisoned up to 30%. Such sanitizing rate is due to the selection of an average quorum size of $n = size(RT)/3$. Accordingly, a relatively high fraction of peers simultaneously remove p_s from their RT which accelerates the sanitizing process. Moreover, as the RTP attack is undirected, p_Q decides about p_s based on sending a lookup request destined to any random peer which allows for faster monitoring procedure as p_i assigns closer time stamps to the quorum's peers.

Note that choosing large quorum size entails a high message overhead. As seen in Figure 6c, the average MC generated due to quorum formations exceeds 700 messages at $N = 30000$. This is due to around 30% malicious insertions in benign peers RT which in turn increases the triggering rate of the sanitizing mechanism.

Nevertheless, MC shows a decreasing trend as MRT decreases. This denotes that the number of quorums initiated decreases as peers RTs get sanitized over time which leads MC to restore the average number of messages exchanged per lookup. In fact, a key factor of our approach is that the sanitizing mechanism do not impose permanent message overhead as the mechanism is invoked only through the detector while no permanent periodic monitoring is required.

Notably, MC depends on the quorum size. This means that selecting smaller n results in lower MC due to decreasing: (i) the number of lookups initiated to mon-

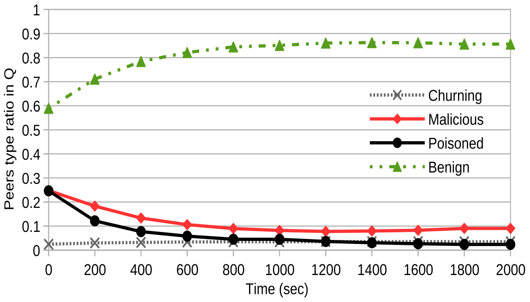


Figure 8: Ratio of types of peers in quorum Q.

itor p_s and (ii) the number of messages exchanged to reach an agreement. However, minimizing MC comes at the cost of a slow sanitizing rate as the number of peers joining Q decreases, which in turn decreases the number of peers that removes p_s per quorum. Subsequently, the propagation rate of peers' correct contact information is affected.

Figure 8 depicts peers participating in quorum formation according to their types. Such measurements are conducted for network size $N = 30000$ and $MRT = 0.28$ to evaluate the reliability of the quorum's decision in drastic RTP attack impact. As peers RT highest poisoning level is at time $t = 0$, the measured values show that the average recorded amount for benign peers $|Q_{B\setminus O\cup C}| = 0.59$ which is greater than $|Q_M| + |Q_C| + |Q_{O\setminus C}| = 0.492$.

This denotes that the majority of the replies in the quorum is benign and thus, the quorum is capable of tolerating the existence of malicious (Byzantine) replies, even at remarkably high MRT values. Hence, the quorum is able to successfully reach a reliable decision. In addition, as SM is triggered, the average number of benign peers $|Q_{B\setminus O\cup C}|$ per quorum continuously increases due to removing malicious peers from benign peers RT. Consequently, the ratio of malicious peers picked during quorum formation that affects the sanitizing mechanism decreases, which provides more reliability for Q to reach an accurate agreement about p_s.

7.6. Summary

Two groups of experiments were conducted in order to (i) assess the impact of launching RTP attacks on P2P networks and provide measures for the impairments imposed on the overlay and (ii) evaluate the performance of our proposed sanitizing mechanism as an approach to remove malicious peers from benign peers RT and accordingly restore the overlay's reliability.

The first experiment evaluates the impact of RTP attacks that target poisoning benign peers RT. Results show that RTP attacks severely degrade the reliability and cause significant perturbations in the overlay as peers can experience more than 64% lookup failures. Further evaluation indicates that acquiring 10% of network resources is enough for the attacker to poison more than 22% of peers RT entries.

The second experiment assesses the performance of the proposed SM. SM provides a remarkable increase in the overlay's reliability as LSR increases up to 100% at a reasonable, non permanent messaging overhead. Moreover, due to SM, malicious peers are removed from the overlay as malicious entries are eliminated from benign peers RT. Consequently, MRT drops from 28% to 3%.

7.6.1. SM possible challenges

One of the challenges for SM is the excessive message overhead exerted on the network when the network is drastically under attack. Given that some critical applications or lightweight peers with limited computing capability such as WSN cannot tolerate such overhead, SM might need some adjustment.

Although selecting a small quorum size n effectively decreases the message exchanged between the quorums peers, the time delay to sanitize the overlay might be intolerant to specific critical applications. Hence, in order for SM to be applicable for such applications, decreasing the messages exchange between peers is a potential solution, taking into consideration the anonymity and accuracy of SM is not affected. For this, we propose to introduce a set of trusted peers in critical P2P environments that can easily manage the quorum initiation. Accordingly, the decision making procedure would result in much lower overhead.

8. Conclusions & Future Work

RTP attacks pose a significant threat to P2P networks as reliability is severely degraded to cause service impairments. As a countermeasure, we have proposed a distributed sanitizing mechanism based on reaching a consensus once a peer is suspected over the lookup process by the DMV based detector. The proposed sanitizing mechanism eliminates more than 90% of the malicious entries from the peers RTs, and successfully restores the benign state of the overlay as the lookup success rate increases to almost 100%. Significantly, the developed approach has been shown to be independent of the overlay structure and attack types to result in a generalized P2P attack detection and mitigation mechanism.

As on-going work, we are assessing the performance of our sanitizing mechanism on directed RTP attacks where malicious peers manipulate replies destined to specific victims in the overlay. Malicious peers can also collude against the sanitizing mechanism once a malicious peer detects being monitored. The sanitizing mechanism for such attacks is currently being tested on PlanetLab.

References

[1] Steinheimer, M., Trick, U., Fuhrmann, W. and Ghita, B., P2P-based Community Concept for M2M Applications, Proc. of Future Generation Communication Technology (FGCT) (2013) 114–119.

[2] Wu, Y., Sheng, Q. and Ranasinghe, D., P2P Object Tracking in the Internet of Things, Proc. of of International Conference on Parallel Processing (ICPP) (2011) 502–511.

[3] Kohnen, M., Leske, M. and Rathgeb, E., Conducting and Optimizing Eclipse Attacks in the Kad Peer-to-Peer Network, Proc. of International Conference on Research in Networking (2009) 104–116.

[4] Locher, T., Mysicka, D., Schmid, S. and Wattenhofer, R, Poisoning the Kad network, Proc. of International Conference on Distributed Computing and Networking (ICDCN) (2010) 195–206.

[5] Douceur, J., The Sybil attack, Proc. of International Workshop on Peer-to-Peer Systems (2002) 251–260.

[6] Germanus, D., Ismail, H. and Suri, N., Mitigating Eclipse Attacks in Peer-to-Peer Networks, Proc. of Communications and Network Security (CNS) (2014) 400–408.

[7] Singh, A., Ngan, T., Druschel, P. and Wallach, D., Eclipse Attacks on Overlay Networks: Threats and Defenses, Proc. of International Conference on Computer Communications (INFOCOM) (2006) 1–12.

[8] Li, Q., Yu, J. and Li, Z., An Enhanced Kad Protocol Resistant to Eclipse Attacks, Proc. of nternational Conference on Networking, Architecture, and Storage (NAS) (2014) 83–87.

[9] Lin, H., Ma, R., Guo, L., Zhang, P. and Chen, X., Conducting Routing Table Poisoning Attack in DHT Networks, Proc. of International Conference of Communications, Circuits and Systems (ICCCAS) (2010) 254–258.

[10] Cholez, T., Chrisment, I., Festor, O., Doyen, G., Detection and Mitigation of Localized Attacks in a Widely Deployed P2P Network, In Peer-to-Peer Networking and Applications 6 (2) (2013) 155–174.

[11] Urdaneta, G., Pierre, G. and Steen, M., A Survey of DHT Security Techniques, In ACM Computing Surveys (CSUR) 43 (2) (2011) 8.

[12] Lee, Y., Koo, H., Choi, S., Roh, B. and Lee, C., Advanced node insertion attack with availability falsification in Kademlia-based P2P networks, Proc. of International Conference on Advanced Communications Technology (ICACT) (2012) 73–76.

[13] Koo, H., Lee, Y., Kim, K., Roh, B. and Lee, C., A DDoS Attack by Flooding Normal Control Messages in Kad P2P Networks, Proc. of International Conference on Advanced Communications Technology (ICACT) (2012) 213–216.

[14] Li, Z. and Chen, X., Misusing Kademlia Protocol to Perform DDoS Attacks, Proc. of International Symposium on Parallel and Distributed Processing with Applications (ISPA) (2008) 80–86.

[15] Castro, M., Druschel, P., Ganesh, A., Rowstron, A. and Wallach, D., Secure Routing for Structured Peer-to-Peer Overlay Networks, In Operating Systems Review (OSR) in ACM Special Interest Group on Operating Systems (SIGOPS) 36 (SI) (2002) 299–314.

[16] Rottondi, C., Savi, M., Verticale, G. and Krauß, C., Mitigation of Peer-to-peer Overlay Attacks in the Automatic Metering Infrastructure of Smart Grids, Security and Communication Networks 8 (3) (2015) 343–359.

[17] Stoica, I., Morris, R., Karger, D., Kaashoek, M. and Balakrishnan, H., Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, In ACM Computer Communication Review (SIGCOMM) 31 (4) (2001) 149 – 160.

[18] Ismail, H., Germanus, D. and Suri, N., Malicious Peers Eviction for P2P Overlays, Proc. of Communications and Network Security (CNS) (2016) to appear.
URL `http://www1.deeds.informatik.tu-darmstadt.de/External/PublicationData/1/MaliciousPeers.pdf`

[19] Nambiar, A. and Wright, M., Salsa: A structured Approach to Large-scale Anonymity, Proc. of ACM Conference on Computer and Communications Security (CCS) (2006) 17–26.

[20] Mittal, P. and Borisov, N., Information Leaks in Structured Peer-to-Peer Anonymous Communication Systems, In Transactions on Information and System Security (TISSEC) 15 (1) (2012) 5.

[21] Germanus, D., Ismail, H. and Suri, N., PASS: An Address Space Slicing Framework for P2P Eclipse Attack Mitigation, Proc. of Symposium on Reliable Distributed Systems (SRDS) (2015) 74–83.

[22] Mittal, P. and Borisov, N., Shadowwalker: Peer-to-Peer Anonymous Communication Using Redundant Structured Topologies, Proc. of ACM Conference on Computer and Communications Security (CCS) (2009) 161–172.

[23] Danezis, G. and Mittal, P., SybilInfer: Detecting Sybil Nodes using Social Networks, Proc. of The Network and Distributed System Symposium (NDSS).

[24] Kos, J., Aiash, M., Loo, J., and Trček, D., U-Sphere: Strengthening scalable flat-name routing for decentralized networks, Computer Networks 89 (2015) 14–31.

[25] Bernstein J., Curve25519: new Diffie-Hellman speed records, International Workshop on Public Key Cryptography (2006) 207–228.

[26] Sharma, D. and Thakur, S., Performance Analysis of Sybil Decline: Attack Detection and Removal Mechanism in Social Network, In International Journal of Computer Science and Information Security (IJCSIS) (2015) 165.

[27] McLachlan, J., Tran, A., Hopper, N. and Kim, Y., Scalable Onion Routing with Torsk, Proc. of ACM Conference on Computer and Communications Security (CCS) (2009) 590–599.

[28] Loesing, K., Murdoch, S. and Dingledine, R., A Case Study on Measuring Statistical Data in the Tor Anonymity Network, Proc. of International Conference on Financial Cryptography and Data Security (2010) 203–215.

[29] Panchenko, A., Richter, S. and Rache, A., NISAN: Network Information Service for Anonymization Networks, Proc. of ACM Conference on Computer and Communications Security (CCS) (2009) 141–150.

[30] Sen, S. and Freedman, M., Commensal Cuckoo: Secure Group Partitioning for Large-scale Services, In Operating Systems Review (OSR) in ACM Special Interest Group on Operating Systems (SIGOPS) 46 (1) (2012) 33–39.

[31] Young, M., Kate, A., Goldberg, I. and Karsten, M., Practical Robust Communication in DHTs Tolerating a Byzantine Adversary, in: Proc. of International Conference on Distributed Computing Systems (ICDCS), 2010, pp. 2009–31.

[32] Kapadia, Apu. and Triandopoulos, Nikos., Halo: High-Assurance Locate for Distributed Hash Tables, Proc. of The Network and Distributed System Symposium (NDSS) (2008) 142.

[33] Maymounkov, P. and Mazieres, D., Kademlia: A Peer-to-Peer

Information System Based on the XOR Metric, Proc. of International Workshop on Peer-to-Peer Systems (IPTPS) (2002) 53 – 65.

[34] Eichert, F., Monhof, M. and Graffi, K., The Impact of Routing Attacks on Pastry-Based P2P Online Social Networks, Proc. of European Conference on Parallel Processing (Euro-Par) (2014) 347–358.

[35] Wang, P., Tyra, J., Chan-Tin, E., Malchow, T., Kune, D., Hopper, N. and Kim, Y., Attacking the Kad Network, Proc. of International Conference on Security and Privacy in Communication Networks (SecureComm) (2008) 1–10.

[36] Naoumov, N. and Ross, K., Exploiting P2P Systems for DDoS Attacks, Proc. of International Conference on Scalable Information Systems (2006) 47.

[37] Liang, J., Naoumov, N. and Ross, K, The Index Poisoning Attack in P2P File Sharing Systems, Proc. of International Conference on Computer Communications (INFOCOM) (2006) 1–12.

[38] Kamat, P., Gite, S., Kumar, M. and Patil, S., A Critical Analysis of P2P Communication, Security Concerns and Solutions, In International Journal of Applied Engineering Research 9 (24).

[39] Ismail, H., Germanus, D. and Suri, N, Detecting and Mitigating P2P Eclipse Attacks, Proc. of International Conference on Parallel and Distributed Systems (ICPADS) (2015) 224–231.

[40] Germanus, D., Langenberg, R., Khelil, A. and Suri, N., Susceptibility Analysis of Structured P2P Systems to Localized Eclipse Attacks, Proc. of Symposium on Reliable Distributed Systems (SRDS) (2012) 11–20.

[41] Lin, H., Ma, R., Guo, L., Zhang, P. and Chen, X., Conducting Routing Table Poisoning Attack in DHT Networks, Proc. of International Conference of Communications, Circuits and Systems (ICCCAS) (2010) 254–258.

[42] Belli, F. and Jedrzejowicz, P., Comparative Analysis of Concurrent Fault-Tolerance Techniques for Real-Time Applications, Proc. of International Symposium on Software Reliability Engineering (ISSRE) (1991) 202–209.

[43] Castro, M. and Liskov, B., Practical Byzantine Fault Tolerance, Proc. of USENIX Symposium on Operating Systems Design and Implementation (OSDI) (1999) 173–186.

[44] Augustine, J., Pandurangan, G., Robinson, P. and Upfal, E., Distributed Agreement in Dynamic Peer-to-Peer Networks, In Journal of Computer and System Sciences 81 (7) (2015) 1088–1109.

[45] Fischer, M., The Consensus Problem in Unreliable Distributed Systems, Proc. of International Conference on Fundamentals of Computation Theory (1983) 127–140.

[46] Hsieh, H. and Chiang, M., New Approach to Improve the Generalized Byzantine Agreement Problem, In International Journal of Computer Theory and Engineering 7 (2) (2015) 120.

[47] Castro, M., and Liskov, B., Practical Byzantine Fault Tolerance and Proactive Recovery, In ACM Transactions on Computer Systems (TOCS) 20 (4) (2002) 398–461.

[48] Pongor, G., OMNeT: Objective Modular Network Testbed, Proc. of International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems (MASCOTS) (1993) 323–326.

[49] Baumgart, I., Heep, B. and Krause, S., OverSim: A Flexible Overlay Network Simulation Framework, Proc. of Global Internet Symposium (GI) (2007) 79 – 84.

[50] Yao, Z., Leonard, D., Wang, X. and Loguinov, D., Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks, Proc. of International Conference on Network Protocols (ICNP) (2006) 32–41.

Hatem Ismail is currently a PhD student at DEEDS group in the Department of Computer Science at Technische Universitt of Darmstadt, Germany. His research interest includes P2P attacks evalution, mitigation and performance enchancement techniques.

Daniel Germanus obtained his PhD from TU Darmstadt, Germany. His research interests include P2P network resiliency and critical information infrastructures. Currently, he is a researcher at ENX Association.

Neeraj Suri received his PhD from the University of Massachusetts at Amherst and is a Chair Professor at the Technische Universitt of Darmstadt, Germany. His research addresses the design, analysis, and assessment of trustworthy cloud services.