Mitigating Eclipse Attacks in Peer-to-Peer Networks

Daniel Germanus*, Stefanie Roos†, Thorsten Strufe†, Neeraj Suri*
*DEEDS Group, CS Department, TU Darmstadt, Germany
{germanus, suri}@cs.tu-darmstadt.de
†Privacy and Data Security Group, CS Department, TU Dresden, Germany
{stefanie.roos, thorsten.strufe}@tu-dresden.de

***Abstract*—Peer-to-Peer (P2P) protocols usage is proliferating for a variety of applications including time- and safety-critical ones. While the distributed design of P2P provides inherent fault tolerance to certain failures, the large-scale decentralized coordination exhibits various exploitable security threats. One of these key threats are Eclipse attacks, where a large fraction of malicious peers can surround, i.e., *eclipse* benign peers. Topology-aware localized Eclipse attacks (taLEAs) are a new class of such attacks that allows for highly efficient denial of service attacks with a small amount of malicious resources. Our contribution is twofold: First, we show the generic susceptibility of structured P2P protocols to taLEAs. Second, we propose a new lookup mechanism for the proactive and reactive detection and mitigation of such attacks. Our novel lookup mechanism complements the common deterministic lookup with randomized decisions in order to reduce the predictability of the lookup. We validate our proposed technique via extensive simulations, increasing the lookup success to 100% in many scenarios.**

***Index Terms*—Peer-to-Peer Protocol, Distributed Hash Table, Security, Localized Eclipse Attack, Mitigation**

I. Introduction

The distributed and dynamic nature of Peer-to-Peer (P2P) systems offers inherent resilience to peer failures or parts of the underlying communication network. Several protocols have been proposed over the past decade, e.g., [1]–[4]. Due to their high robustness, P2P systems represent an attractive networking substrate for critical applications such as supervisory control and data acquisition [5] (SCADA) or wide area monitoring systems [6] (WAMS). In particular, structured systems offer highly efficient content discovery combined with a resilience to failures. They deterministically map content to peers using a distributed hash table (DHT), and achieve short routes by creating a routable small-world topology connecting peers based on a distance function of the hashes they are responsible for. Despite their inherent resilience to failures, structured designs are susceptible to attacks such as Sybil, Routing, Poisoning, Join/leave, or Eclipse attacks (EA) [7]. Our focus is on EAs that constitute a significant vulnerability of P2P systems [8]–[15] and their applications. Besides having attracted attention in the research community, measurement studies of the topology of the KAD network indicate the existence of Eclipse attacks in the real-world [16], [17].

Research partially funded by DFG GRK 1362 (TUD GKmM), DFG Research Grant 'Resilient Network Embeddings for Friend-to-Friend Networks', and LOEWE TUD-CASED.

The EA and its variants often constitute precursors for high level attacks to be launched, e.g., eavesdropping or denial of service. During an EA, an attacker typically places or corrupts peers in the overlay, conducts routing table poisoning (RTP) to subvert benign peers, and intercepts messages of arbitrary benign peers. RTP techniques increase the number of messages between the benign peers that a malicious can intercept.

Varying from the conventional EA, a localized Eclipse attack (LEA) intercepts messages that are addressed to only a subset of benign peers, e.g., a replica group in a distributed hash table (DHT) or a specific peer in a service overlay network. For an EA or LEA to be effective, the malicious peer fraction must be in the range of 10% to 20% of the overlay size [12]. For large systems and a moderate attack duration, this could imply considerable effort for an adversary. In this context, we identify a new class of attacks that we term as *topology-aware LEA* (taLEA) attacks and highlight the systematic weakness of most structured P2P protocols to it. taLEAs are highly efficient attacks that require only a small number of malicious peers that need to be located in the proximity of the targeted peers. In contrast to the EA and LEA approach where the sheer mass of malicious resources determines the attacks' effectiveness, taLEAs exploit an inherent weakness of most DHTs [15].

taLEAs exploit a systematic weakness of the *converging* lookup mechanisms that are applied in DHTs, and work as follows: When a requesting peer has no contact information of the recipient in its routing table, a lookup is initiated, which repeatedly queries peers closer to the destination peer until the contact information has been resolved. Consecutive lookup queries hence *converge* towards the destination's proximity. During a taLEA, the respective proximity consists of malicious peers, which can act adversarially, e.g., by returning bogus contact information or dropping requests. However, not all lookups are routed through the destination's proximity, but some of them resolve the destination from a higher distance (called *divergent* lookups). Assuming a malicious proximity, our mitigation technique thus aims to resolve the destination via those distant contacts.

A. Contributions

Our paper makes the following contributions. We show the construction of a generalized class of taLEAs and its applicability to the widely used P2P protocols Chord [1], Kademlia [3], R/Kademlia [18], and S/Kademlia [19]. Fur-

ther, we propose and analyze *diverging* lookups and their relationship to overlay topologies to make the case for this mitigation technique. Therefore, we simulated the divergent lookup mechanisms for various Kademlia variants using the OverSim [20] simulation environment in order to assess their reliability and efficiency. We conduct a comprehensive simulation study. Results show that diverging lookups are able to mitigate taLEAs with up to 100% success rate in overlay networks which are subject to churn. Moreover, we propose an n-version-lookup (NVL) framework as an architectural pattern to integrate our mitigation into existing systems.

B. Paper Structure

We present related work in Section II. Section III provides a highlevel overview of the considered attack and our mitigation approach. The system model outlined in Section IV describes the detailed functions of the P2P protocol class. The new taLEA attack class and its attack analysis is treated in Section V. The proposed attack mitigation approach is detailed in Section VI. Details on our simulator implementation and evaluation can be found in Section VII.

II. Related Work

We first overview existing work that discusses LEAs, highlighting their application and protocol contexts as well as the characteristics of possible mitigation, if provided in these works. We also summarize selected works that discuss modified overlay lookup variants.

The first mention of LEAs appears as a subsidiary remark in [12]; the article focusses on generic EAs and proposes mitigation through degree bounding, i.e., limiting the amount of routing table entries for each peer. Adherence to this limitation needs to be checked periodically using an auditing scheme. The authors conclude that degree bounding is probably not a suitable countermeasure for LEAs.

In [8] and [10], Sybil attacks are launched as a preparatory step for LEAs against KAD, which is a Kademlia-based filesharing system. The LEA in [8] uses 8 malicious resources to eclipse an address space range with common prefix length 8 in order to intercept all KAD search requests destinated to the victim peer. The proposed mitigation requires a certificate authority in order to enable strong encryption for admission and authentication of peers. The proposed countermeasure in [10] aims at IP address restriction and a flooding protection scheme.

In [9], LEAs using the backpointer hijacking method are discussed in the context of the KAD network. The authors propose to mitigate this by routing table policies which disallow peers from reclaiming the same overlay address using a different IP address.

In [13], two LEA variants against the KAD protocol are presented, where the first variant resembles the approach in [8], [10]. The second variant prevents legitimate peers from publishing new content in the KAD filesharing system by placing malicious peers close to the legitimate one. Countermeasures in this work refer to structural routing table constraints according to the assumption that attacker resources might originate from a single or few underlay network domains. No mitigation is proposed in this work.

In [14], the authors show the LEA susceptibility of the KAD filesharing network which is based on a Kademlia protocol modification. In their LEA discussions, KAD's tolerance zone, which can be regarded as an application level proximity concept, is populated with malicious peers. The evaluation focusses on the attack's impact over time, no mitigation is proposed.

In [15], the authors discuss the systematic LEA susceptibility from an application independent view and refer to accountable design choices of many structured P2P protocols. The authors conducted an analyical and experimental study to underline the potentially high impact of the attack for the Chord [1], Pastry [2] and Kademlia [3] protocols.

In [11], the authors focus on the KAD network and discuss localized attacks such as the LEA in a publicly accessible filesharing network. They propose a reactive countermeasure which crawls the overlay address space and determines deviations in the distribution of peers' IDs in the overlay address space and thereby deduce malicious peers. The approach has been tested for the KAD filesharing system and shows a small false-negative rate.

In [21], the authors propose secure routing for P2P overlays which is an adequate mitigation technique for EA variants. Disadvantageously, this requires crypto primitives and a certificate authority. Consequently, this approach neglects anonymity or openness requirements of some applications.

In [19] and [22], the authors propose overlay routing based on multiple and independent paths in order to increase the robustness. Due to the fact that these approaches rely on convergence across various paths or dimensions, the susceptibility to taLEAs remains, although with potentially higher cost.

We note that the authors of [10] and [11], in the context of a filesharing system, present an instance of a EA termed as a "localized attack" that has some similarities to a taLEA. However, this elaboration is for a specific protocol.

Modified lookup strategies for DHTs have been considered for achieving sender and receiver anonymity in structured overlays as well. Similar to our approach, these strategies introduce randomness into the algorithm.

SALSA [23] establishes circuits in the manner of TOR [24] to obfuscate the sender of an request. It furthermore protects against EA by performing redundant lookups. However, all lookups are in general convergent, so that no protection against LEAs is provided.

Shadowwalker [25] enhances the above approach by a verification scheme that demands that a certain number of peers confirm the identity of a peer before any data exchange. However, the approach has been shown to be vulnerable to selected denial of service attacks [26].

The R^5N routing protocol [27] applied in GnuNet also initially forwards multiple requests randomly before executing a deterministic lookup. A high failure and attack resilience is archived by the price of $\mathcal{O}(\sqrt{n})$ replicas, which limits the

proposed algorithms to filesharing and storage applications.

III. Highlevel Threat and Mitigation Description

This section presents the reader a highlevel overview of the taLEA and our proposed mitigation technique.

A. Threat

The attack aims to block queries addressed to a certain set of victim peers. Hence, the victim peers can only partially provide the requested service, which can range from simple file-sharing to critical services. The temporal scope of the partial unavailability covers the timespan of the malicious peers' presence in the overlay, whereas the spatial scope includes all benign peers that try to send messages towards the victim(s) and have no contact information about the victim(s) in their routing tables.

B. Key Vulnerability of P2P Lookups

The key concept of a taLEA is to place malicious peers close to the victim peers, thus exploiting the convergence of the DHT lookup towards its destination. The malicious peers can be either placed by inserting new peers into the system (if it is open) or by compromising suitable existing peers. Due to the convergence of the lookup, closer peers are more likely to be contacted when routing for a victim, so that the probability of blocking a request to a victim is increased by placing malicious peers as close as possible. However, this placement is not always optimal. In Section V, we clearly define necessary conditions on the P2P systems for the attack to be optimal.

C. Mitigation

Our mitigation technique provides an alternate approach for the lookup mechanism. It does not converge, but is a *divergent lookup* which does not query peers close by the potential victim peer(s), as these are assumed to act malicious. Rather, search strategies are integrated into the lookup in order to undermine the predictability of the convergent routing. The technique is presented in detail in Section VI.

IV. System Model

We now outline the terminology, system assumptions and concepts behind our proposed detection and mitigation schemes. First, we describe P2P overlay networks from a graph theoretic perspective. Next, we outline the fundamental functionality of structured P2P overlays as a basis for the taLEA model in Section V.

A. P2P Overlay Model

The P2P overlay network is modeled as a directed graph $D = (P, E)$. Each peer is an element of the peer set P and maintains a routing table that contains contact information about its adjacent peers to allow for direct communication among them. These routing tables define the edge set E. For example, when peer a's routing table includes peer b, then a directed edge $e = (a, b)$ in the edge set E exists. The set $E^-(a)$ denotes incoming edges to peer a and $E^+(a)$ denotes outgoing edges. Overall, $E(a) = E^-(a) \cup E^+(a)$. We subdivide P into the following disjoint subsets: benign peer set $B \subset P$, malicious peer set $M \subset P$ and victim peer set $V \subset P$. Formally, we have $P = B \cup V \cup M$ with $M \cap V = \emptyset$, $M \cap B = \emptyset$, and $B \cap V = \emptyset$. The total number of peers in the overlay is denoted by $N = |P|$.

B. P2P Protocol Model

Our P2P protocol model describes fundamental concepts which are commonly found in structured P2P protocols.

1) Address space & distance function: Objects, such as peers or data items, are assigned IDs in an overlay address space. Typically, the address space is an integer scale in the range of $[0, \ldots, 2^w - 1]$ with w being 128 or 160 in general. The ID of an object is computed based on a unique feature, e.g., an IP address, MAC address, a logical application identifier, or a random number, which is then processed by a cryptographic hash function in order to map the feature onto the address space. Data item IDs are commonly derived as the hash of their content or some other identifying feature. A distance function $d(a, b)$ allows distance computations between any two IDs a, b in the address space. So an object with ID a can be mapped to peers with IDs $p_1, \ldots, p_r$ such that $d(a, p_1), \ldots, d(a, p_r)$ are small. Peers store objects or pointers to objects that are mapped to them. The distance function differs among various P2P protocol implementations. In the following, all operations on IDs are assumed to be modulo 2^w.

2) Routing table: Peers store pointers to other peers in a data structure called routing table. These pointers usually consist of a tuple that relates a peer's overlay address space ID to a contact information from the underlay network, e.g., another tuple consisting of IP address and port number. Routing tables in average store $c\ log(N)$ pointers with c being a protocol specific constant. The routing tables are structured in such a way that at most c pointers with distance range of 2^i to 2^{i+1} are stored for $i = 0 \ldots w - 1$.

3) Proximity: A peer's proximity includes close-by peers and differs across protocol implementations. We define overlay edges among proximate peers as short distance edges (SDE) $E^+_{SDE} \cup E^-_{SDE} = E_{SDE}$. The remaining overlay edges of peers not located in the same proximity are termed as long distance edges (LDE) $E^+_{LDE} \cup E^-_{LDE} = E_{LDE}$, $E = E_{SDE} \cup E_{LDE}$. We propose two proximity types, namely symmetric and asymmetric ones. We define λ as the average distance between two neighbored peers in the overlay address space. Moreover, we introduce a protocol-specific constant κ which denotes the number of proximate peers. We define the proximity range as $B_{\lambda\kappa}(v) = \{a \in \mathbb{Z}_{2^w} : d(a, v) \leq \lambda\kappa\}$. We differentiate between symmetric and asymmetric proximity placements. In a symmetric proximity, v is the centre of the range, whereas in an asymmetric proximity, all proximate peers have either larger or smaller IDs than v.

4) Lookup mechanism: If a peer p_r needs to communicate with a peer v which is not listed in p_r's routing table, then p_r initiates a lookup call in order to resolve v's contact

information. Lookup calls consist of at most k concurrent queries, each sent to a different peer. Lookups can be either explicit or implicit calls. Explicit calls are dedicated for resolving the contact information only. Implicit lookups on the other hand contain the application's payload to be sent to v as well.

5) Iterative Overlay Routing: Using the iterative approach allows a requesting peer p_r routing process supervision to a large extent. Assuming that p_r tries to resolve contact information of v, p_r is sending k concurrent requests to known peers with the lowest possible distance to v. These peers send their lookup results back to the originator p_r who is then able to either contact v (if it has been resolved) or to query further peers. Iterative routing increases the resilience to several attacks and has potential for optimizations by using concurrent lookup messages and/or multipath routing of application payloads, it is the standard routing proposed for the Kademlia [3] protocol.

6) Recursive Overlay Routing: Recursive routing does not return intermediate messages to the requestor as opposed to the iterative approach. Using the full recursive routing variant, the message payload which should be transferred to the destination peer is being piggybacked on a lookup message and each peer that participates in the specific lookup call sequence forwards the payload until it reaches the destination. The semi recursive variant does not piggyback the payload, instead v's contact information is returned to p_r and then the payload is transferred between p_r and v. Recursive routing tends to have less communication overhead than iterative routing. It is used in the R/Kademlia protocol [18].

7) Maintenance protocol: Peers exchange messages to periodically check for liveness of peers and thus keep/remove contact information of responding/unresponding peers. Additionally, contact information is exchanged among peers during other events, e.g., join, lookup, or leave procedures.

V. taLEA Design and Analysis

In this section, we explicitly define our attacker model, and show that placing peers as close as possible to the victim is optimal for a large class of systems, but not for all of them.

A. Attack Goal and Attacker Capabilities

The goal of the attack is to block as many requests as possible for a specific set of peers V by dropping all requests addressed for v. During the following analysis, we let $V = \{v\}$ be a singleton, the extension to sets of arbitrary size is straightforward. The selection of $v \in V$ is based on external knowledge about the responsibilities of the peers. We assume the conventional model of a local attacker who can drop, read, modify, and replay messages addressed to it, but cannot observe any other communication. The network is assumed to be open, so that the attacker can introduce $|M|$ peers into the network. The IDs of these malicious peers can be chosen from a set ID_A of IDs. In general, the choice of IDs is limited by the number of identifying features, e.g., the number of IP addresses an attacker controls. In principle, an attacker can check if an ID corresponds to an existing peer in the network. That can be easily done by querying for these IDs. Hence we assume that the topology and the ID assignment in the network is known to the attacker. Note that the assumption is not necessary to prove the optimality of our attack for systems such that i) the routing table entries are non-uniquely defined by their identifiers, and ii) a prefix-based distance function is used. If the topology is uniquely determined by the ID assignment, placement decisions could potentially be improved by checking if IDs of potentially neighbors of the victim actually correspond to peers and have to be blocked. Formally, taLEAs solve the following optimization problem: *Given a victim v with ID ID_v and a set of potential IDs ID_A, find the set $ID_M \subset ID_A$, such that the expected fraction $\mathbb{E}(X_v)$ of blocked queries to v is maximized.*

B. Attack Design and Analysis

We solve the optimization problem under the assumption that all peers within a certain ID range can be chosen as neighbors, rather than one uniquely determined peer. We focus on the non-uniquely determined systems because most deployed systems, especially Kademlia and its derivates, follow that principle.

Lemma 5.1: Let D be a structured P2P overlay such that each peer u divides the known peers into sets $a_1(u), \ldots, a_s(u)$ according to their identifier for some system-dependent integer s. Furthermore, for any two peers $w_1, w_2 \in a_i(u)$ and any peer $z \in a_j(u)$ with $i \neq j$ we have that $d(w_1, w_2) < d(w_1, z)$. From each of these sets, u selects at most k contacts for its routing table. Apart from the above restriction, the neighbor selection is independent of the ID, i.e. if two peers are both contained in $a_i(u)$ for some i and no further information is given, they are equally likely to be in the routing table of u. During the routing, at most the $\alpha \leq k$ closest peers to v in a routing table are contacted. The fraction $\mathbb{E}(X_v)$ of blocked queries is maximized if ID_M consists of $|M|$ closest IDs in ID_A to v.

Proof: A query is blocked if α malicious peers are contacted before v is contacted. Let u be any peer contacted during routing for v. Consider any malicious peer m. We have $v \in a_i(u)$ for some i, and $m \in a_j(v)$ for some j. For $i = j$, the probability of v to be in u's routing table is reduced and the probability to be successful in this hop. It remains to show that the probability to choose m for the next hop is maximized if m is as close as possible to v. For $i \neq j$, m is only contacted if there are less than $\alpha \leq k$ peers in $a_i(u)$. However, then m can also be added to $a_i(u)$ if possible and increase the probability to be contacted. The number of peers u for which v and m belong to the same set $a_i(u)$ is maximized if m is as close to v as possible. In summary, for every peer in the network, the probability to block the query is maximized if the malicious peers are given the IDs in ID_A that are closest to v. ■

The above proof holds for prefix-based distance functions as used in all Kademlia derivates such as KAD, but as well in Pastry and Tapestry-like systems. The result does not hold for Chord, as can be seen from the following example. Assume

there are two malicious peers m_1 and m_2, which are both placed closer to v than any benign peer b. Then in the absence of m_2 all requests for v otherwise forwarded to m_1 would be forwarded to m_1, so that placing m_2 does not increase the attack success. However, placing m_2 using the ID in ID_A closest to a benign neighbor b_2 adjacent to v increases the probability to block a query. Hence, the attack strategy cannot be applied to systems in which neighbors are uniquely defined by the IDs. However, all actually deployed systems satisfy the constraints in Lemma 5.1, so that we restrict our analysis to a general study of these systems.

VI. taLEA Mitigation

We propose divergent lookups for taLEA mitigation. In this section, we discuss the objectives, parameters, and search strategies for divergent lookups. Moreover, we come up with brief examples. Finally, we propose a framework for divergent lookup integration in existing systems.

A. Objectives

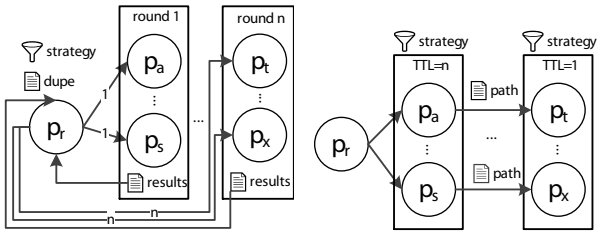
We now specify *divergent lookups* as proposed in Section III-C. In contrast to their convergent counterpart, divergent lookups do *not* converge towards the destination peer's proximity. This is a key requirement for our mitigation, as the majority of convergent lookups query peers in the destination's proximity (see Section V and [15]) and we assume that the taLEA attacker populates the proximity.

B. Divergent Lookup Mechanism

We describe the divergent lookup mechanism in the following subsections in detail. We start with the needed parameters, provide examples and present two different search strategies.

1) Address Space Restriction: In order to avoid convergence towards the destination's malicious proximity, the address space has to be restricted for the divergent lookup. We assume that the destination is peer $v \in V$ and for simplicity we assume $|V| = 1$. Moreover, we assume that the determination of V is trivial and can be performed locally. Once V is determined, the combination of proximity address space ranges for all $v \in V$ is an input parameter for the divergent lookup. The structure of a proximity depends on various protocol design choices and parameters. The proximity's range depends on the amount and distribution of peers in the overlay network and the address space size. Peers may accurately estimate the proximity range by using an overlay size estimation [28] and we further assume that relevant protocol parameters are known.

2) Concurrency & Depth Limitation: Divergent lookups may be executed concurrently as a means of potential speed up. Therefore, we refer to k as the concurrency parameter. Furthermore, the divergent approach requires parameters for the maximum amount of rounds (rds) in iterative overlay routing or a time-to-live (TTL) in recursive overlay routing. Also, a threshold for the maximum amount of returned peers (res) per lookup message is specified.



(a) Iterative (b) Recursive

Fig. 1. Iterative and recursive divergent lookup examples.

3) Random Walk Search Strategy: For the iterative divergent lookup, the random search strategy randomly selects k peers from the routing table in round 1 and subsequently k peers from the deduplicated result set for rounds > 1. The recursive variant differs, as it emits initially k lookup messages, and subsequently only one further lookup message on each intermediate peer in order to prevent a lookup message excess in the overlay.

The advantage of random walks is that zero knowledge is required about the protocol, workload or churn parameters. Consequently, it can be easily applied for various protocols and applications. The strategy does not search exhaustively to prevent message excess. Clearly, the efficiency is moderate and depends on present LDEs in the overlay. Simulation experiments in Section VII show that this is the case for various P2P and application parameter choices.

4) Striped Search Strategy: The striped search strategy is expected to increase the divergent lookup's efficiency compared to the random walk approach. Using the striped search strategy, the restricted address space is further partitioned into stripes to conform with the expected LDE distribution. Assuming that $|V| = 1$, then peers located in the two address space stripes adjacent to v's proximity have a higher probability of having an LDE towards v than further distant stripes. Consequently, a divergent lookup approach may define a stripe search preference order to increase the lookup efficiency especially in very dense populated overlays.

5) Divergent Lookup Examples: We now provide brief examples for the iterative and recursive divergent lookups.

Figure 1a provides an example for the iterative divergent lookup with n rounds and k concurrent requests. Results of each round are returned to the requesting peer p_r, and compared to a duplicate list such that no peer receives more than one lookup message per lookup call. In case the destination has not been resolved and the maximum number of rounds has not been reached, the next lookup round is initiated by passing the most recent result set to the requestor peer's search strategy.

In contrast, the recursive divergent lookup (cf. Figure 1b) cannot provide deduplication, because intermediate results are not collected by the requestor. Instead, each intermediately queried peer initiates the next lookup message. Using the recursive variant, the lookup is not conducted using rounds but each of the k messages has a TTL value that is decreased each time an intermediate peer forwards the lookup message.

Lookup messages are forwarded unless $TTL = 0$. Each lookup message keeps track of its hop sequence in order to avoid lookup cycles. In case the destination was resolved, its contact information is returned to p_r.

C. N-Version-Lookup (NVL) Framework

We propose to integrate divergent lookups utilizing the diversity concepts advocated from the classical framework [29]. The approach improves fault tolerance due to design diversity, i.e., parts of a software system are implemented independently for n times based on the same specification. In this case, we address attacks, which can be regarded as deliberate or malicious faults. Two independent variants of the lookup exist, i.e., a convergent and divergent variant. Both require the destination ID as an input parameter. The rationale for applying a diversity scheme is threefold: (i) Timeliness – divergent lookup calls require more hops than convergent ones and therefore induce higher latencies. In a time-critical context, thresholds could be defined to consider only those lookups that fulfill given timeliness thresholds. (ii) Susceptibility – convergent lookups are highly susceptible to taLEAs, whereas divergent lookups show little risk to taLEAs with proper parameter selection. (iii) Intrusion detection – in case both lookup mechanisms returned different contact information for the same destination ID, the decision mechanism detects this and either returns the divergent lookup result to the requesting peer or declares the current result set invalid and restarts the lookup. Downstream mechanisms could blacklist or further probe the suspected malicious peer. NVL can be applied proactively, i.e., during ad hoc lookup calls, and reactively as part of recurrent routing table maintenance events.

VII. Evaluation of taLEA & Divergent Lookups

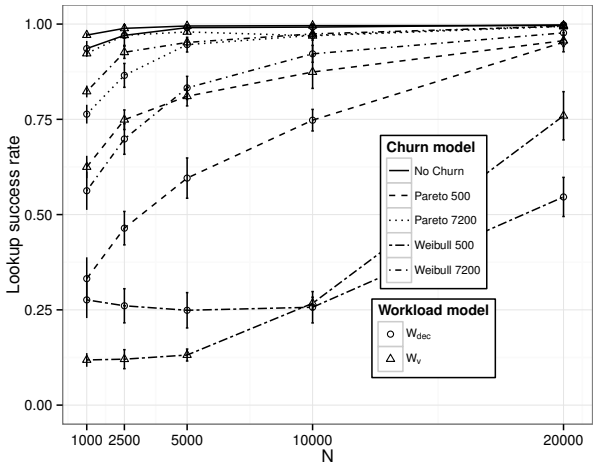
Our simulation based divergent lookup mechanism assessment is motivated by the following set of research questions:

1) *taLEA Baseline* – Firstly, what is the taLEA impact for the traditional convergent lookup mechanism? This is an experimental in-depth follow up of [15].
2) *Divergent Lookup Reliability* – Which success rates of divergent lookups can be achieved for iterative and recursive Kademlia variants, churn variants, workload models, and overlay network sizes?
3) *Divergent Lookup Efficiency* – What is the messaging overhead to resolve contact information using the divergent lookup approach?

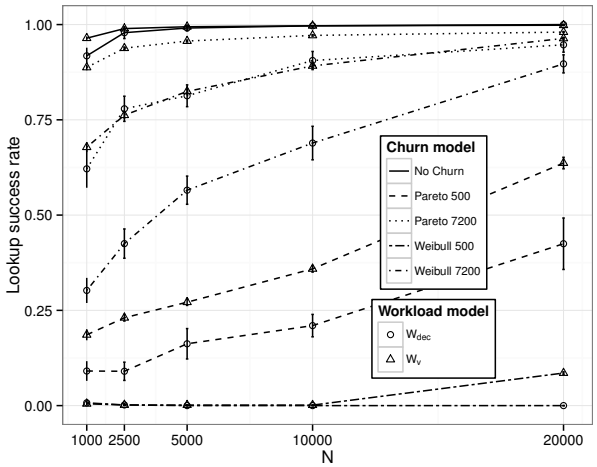
To address our research questions, we simulate overlay networks under taLEA conditions with convergent and divergent lookups. We start by describing the simulation set-up, all relevant model definitions, and parameters. Afterwards we define our metrics, based on which we discuss the results for the the above research questions in detail.

A. Simulation Framework

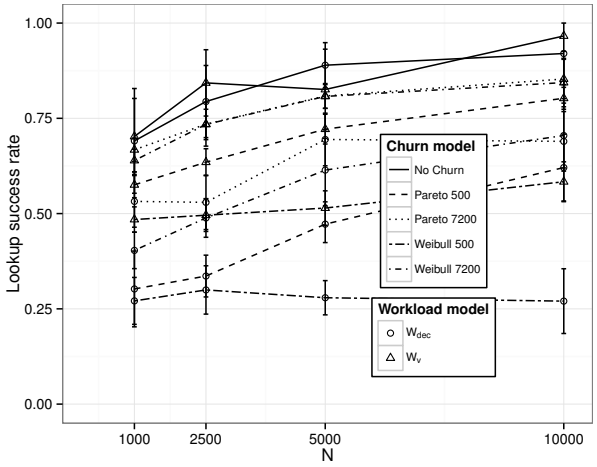
We use the OMNeT++ [30] discrete event simulator together with the OverSim [20] framework. We focus on three different variants of Kademlia, i.e., Kademlia [3] and S/Kademlia [19],



(a) Kademlia (iterative)



(b) R/Kademlia (recursive)



(c) S/Kademlia (iterative) with $s = 16$ siblings and $d = 4$ disjoint paths.

Fig. 2. Success rates of convergent lookups during a taLEA.

which use iterative overlay routing, and R/Kademlia [18], which uses recursive routing. Our analysis is focused on variants of the Kademlia network, since all large-scale DHTs are based on Kademlia.

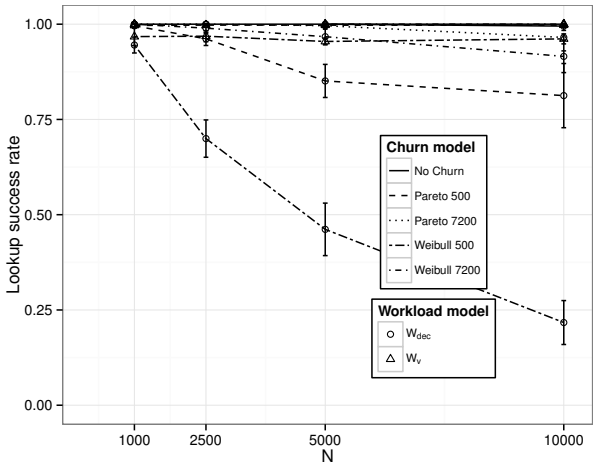
Each simulation experiment consists of 20 runs. Individual runs in OMNeT++ are subject to differing random number generator initializations. This results in different churn and workload initializations and caters for the desired statistical variance of the measurements. Next, we describe the workload and churn models which have been used for the simulations.

1) Workload Simulation Models: By workload models, we refer to the interaction patterns between peers, in particular the request distribution over peers. We distinguish two workloads. For the first workload, W_{dec}, requests are uniformly distributed over $B \cup V$, in particular the victim peer v is as likely to be addressed as any other benign peer. Possible applications are instant messaging or video telephony. For the second workload, W_v, v provides a popular service, such that it is requested with a probability of 90%, whereas all benign peers are equally likely to be requested. Possible applications are safety-critical SCADA or WAMS applications. The message sending interval in both models is subject to a normal distribution with 100s mean and a standard deviation of 10s.

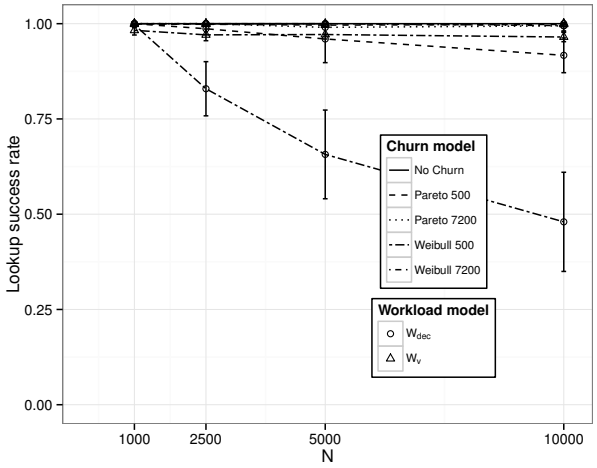
2) Churn Simulation Models: Churn denotes the effect of peers entering and leaving the overlay. Manifold reasons for churn exists, e.g., deliberate user decisions, software failures, mobility (i.e., moving peers), changing the network (e.g., from 4G to wireless or back), communication network perturbations, or blackouts due to battery depletion or electricity grid failures. In the context of our evaluation, churn has noticeable effects on the LDE availability in benign peers' routing tables that point towards the victim peers. Therefore, we employ a *NoChurn* model in which peers do not leave the network which reflects for example an isolated corporate network environment. Moreover, we consider four established churn models [31] which differ in terms of the distribution function (Pareto and Weibull), the mean lifetime, and the mean deadtime. Mean lifetimes are either 500s or 7200s. The Pareto churn models also make use of a deadtime, which refers to the duration between leaving and entering the overlay. Mean deadtimes are set to the same values like mean lifetimes. In view of our experiment duration of about 14 hours, the mean lifetimes of 500s (7200s) imply that the benign peer population B worked its way through 100 (7) full replacement cycles. We label the different churn models as follows: Pareto500, Pareto7200, Weibull500, Weibull7200, and NoChurn.

3) taLEA Simulation Model: We conduct our experiments with $|M| = 24$ malicious peers per victim peer using a symmetric peer placement (cf. Section IV-B3 and [15]).

4) Lookup Simulation Parameters: Research question 1 is addressed by setting up a taLEA in the simulator and using the convergent lookup mechanism. Analogously, research question 2 addresses the divergent lookup mechanism in combination with the random walk search strategy and defer the advanced search strategies such as the striped one for future work. The simulated time for our lookup experiments



(a) $rds = 10$, $k = 10$



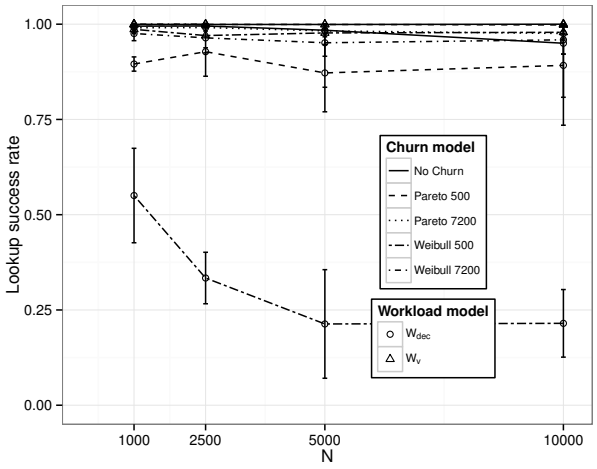
(b) $rds = 30$, $k = 10$

Fig. 3. Success rates of iterative divergent lookups during a taLEA.

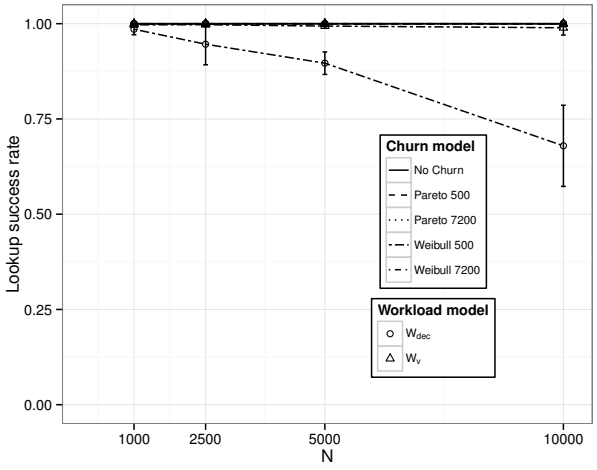
is 14 hours. We provide a taLEA baseline evaluation using the convergent lookup (cf. Figures 2a through 2c) as well as an evaluation of our mitigation approach using the divergent lookup (cf. Figures 3a and 4a). We simulate overlays with $N = \{1000, 2500, 5000, 10000\}$ peers, selected simulations are provided up to $N = 20000$. The iterative divergent lookup is evaluated for $rds = \{10, 30\}$ rounds and $k = 10$ concurrent messages. The recursive divergent lookup's TTL is set to 10 and $k = \{1, 5\}$.

B. Evaluation Metrics

The *lookup success rate* specifies the average percentage of successful lookup calls of benign peers that request contact information of v. Furthermore, the average *number of messages* per lookup is elevated. Moreover, we specify the 95% confidence intervals resulting from individual simulation runs



(a) $TTL = 10$, $k = 1$



(b) $TTL = 10$, $k = 5$

Fig. 4. Success rates of recursive divergent lookups during a taLEA.

for each experiment. The measurement starts 4000s before each simulation run's end.

C. Result Discussion – Baseline

Figures 2a through 2c show the baseline evaluation of the convergent lookup mechanism. Results for the standard Kademlia protocol are depicted in Figure 2a. The taLEA shows the highest impact for the Weibull500 churn model, i.e., only 25% of the lookup messages are successful for an overlay of size $N = 10000$. The Pareto500 churn model shows a higher resilience, for $N = 10000$ peers 75% of the lookup messages are successful. The other churn/workload model combinations result in lookup success rates of 80% to 100% for $N = 10000$ peers. R/Kademlia's baseline is depicted in Figure 2b and shows a higher taLEA susceptibility, especially for the Weibull500 churn model with about 1% successful lookups in case of $N = 10000$ peers. Also, the Pareto500 and Weibull7200 models show an increased susceptibility compared to standard Kademlia. The S/Kademlia baseline evaluation results are shown in Figure 2c using a sibling set size $s = 16$ and $d = 4$ disjoint paths. While being better at the lower end, S/Kademlia shows worse results at the upper end of our measurements, i.e., up to 85% lookup success rate with $N = 10000$, compared to standard Kademlia or R/Kademlia.

The baseline evaluation shows that lookup success rates increase with the size of N and that the average lifetime of peers in the overlay is a crucial factor for the taLEA severity. Basically, the longer a peer $b \in B$ remains in the overlay and the more lookup messages for victim peers $v \in V$ are emitted, the lower the taLEA severity, we see two reasons for that: Firstly, in case b looks up v and the request is satisfied by a benign peer, then b creates an LDE (b, v) in its routing table. Secondly, if peer b is being queried by another benign peer b_2 to resolve v's contact information, it is more likely that b has an LDE towards v which can be returned to b_2. The low lookup success rate for R/Kademlia can be explained by the absence of deduplication for R/Kademlia's recursive routing. Hence, a malicious peer could be queried multiple times in parallel for a lookup. The slight increase of the success rate when increasing the network size is to be expected since the number of peers with LDEs to v increases.

D. Result Discussion – Divergent Lookup Reliability

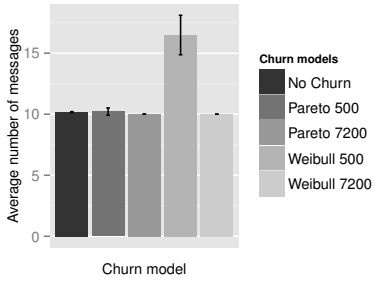
The iterative divergent lookup results are shown in Figures 3a ($rds = 10$) and 3b ($rds = 30$). Increasing the amount of rounds rds from 10 to 30 led to a lookup success rate improvement. Average measurements are except for the Weibull500 W_{dec} experiments in the range of 90% to 100% lookup success rate for $N = 10000$. Compared to the convergent iterative lookups in Figures 2a and 2c this is a considerable reliability gain.

Results of the recursive divergent lookup are presented in Figures 4a ($k = 1$) and 4b ($k = 5$). While for the non-concurrent ($k = 1$) Weibull500 W_{dec} experiment series results are noticeable poor, we performed another experiment series using $k = 5$ concurrent recursive divergent lookups which shows results in the range of 94% to 100% and 65% for Weibull500 W_{dec}. These results outrank most of the recursive convergent lookup shown in Figure 2b.

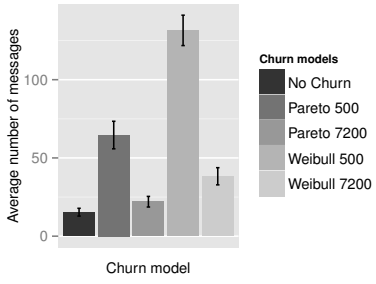
Experiments show a correlation between the average lifetime and better lookup success rates due to a higher average LDE fraction among the benign peer population. Also, the W_v workload model results in better success rates than W_{dec}. In contrast to the baseline experiments, the lookup success rates slightly decrease with increasing N, as a consequence of the divergent lookup limitation by the parameters rds and TTL which may cause divergent lookups to fail. Consequently, the parameters rds and TTL are dependent on N.

E. Result Discussion – Divergent Lookup Efficiency

We present the number of messages in Figures 5a and 5b for the iterative divergent lookup with $rds = 10$, $k = 10$, and $N = 5000$. Clearly, the overhead strongly correlates with the chosen



(a) W_v



(b) W_{dec}

Fig. 5. Iterative divergent lookup number of messages for $N = 5000$.

workload and churn models. Like in the previous evaluation discussion, the Weibull500 model appears as an outlier. We deem the additional divergent lookup overhead acceptable and point out that we expect more advanced search strategies to reduce the overhead.

VIII. Conclusion

We have presented taLEA, a new class of Eclipse attacks that exploit a weakness in common P2P protocols. The crucial point of the taLEA variant is that only a low amount of resources is required to significantly harm selected peers in large overlay networks. In order to increase the resilience to taLEAs, we proposed a taLEA mitigation technique that can be used proactively and reactively, does not require centralized cooperating services such as a certificate authority, and works fully decentralized in heterogeneous and large-scale overlays which are subject to churn. We have validated the approach for different Kademlia protocol variants in a simulation case study and showed for a naive random walk approach mitigation rates of up to 100%. We propose an architectural framework for the integration of divergent lookups together with their convergent counterpart in favor of safety- and time-critical application requirements. For future work, we are planning to implement optimized search strategies and expect keeping high success rates while lowering the network overhead for even larger overlay network sizes, preliminary simulation experiment results support our expectations.

References

[1] I. Stoica et al., "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *In Proc. SIGCOMM*, pp. 149 – 160, 2001.

[2] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," *In Proc. Middleware*, pp. 329–350, 2001.

[3] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," *In Proc. IPTPS*, pp. 53 – 65, 2002.

[4] B. e. a. Zhao, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," *IEEE Journal on Selected Areas in Communications*, pp. 41 – 53, 2004.

[5] J. Lopez, R. Setola, and S. D. Wolthusen, "Critical infrastructure protection," J. Lopez, R. Setola, and S. D. Wolthusen, Eds. Springer-Verlag, 2012, ch. Overview of Critical Information Infrastructure Protection, pp. 1–14.

[6] D. E. Bakken et al., "Smart Generation and Transmission With Coherent, Real-Time Data," *In Proc. IEEE*, vol. 99, no. 6, pp. 928–951, 2011.

[7] G. Urdaneta et al., "A Survey of DHT Security Techniques," *ACM Computing Surveys*, pp. 8:1–8:49, 2011.

[8] M. Steiner et al., "Exploiting KAD : Possible Uses and Misuses," *Computer Communication Review*, vol. 37, no. 5, pp. 65–69, 2007.

[9] P. Wang et al., "Attacking the Kad Network," *SecureComm*, pp. 1–10, 2008.

[10] T. Cholez et al., "Evaluation of Sybil Attacks Protection Schemes in Kad," *Scalability of Networks and Services*, vol. 5637, pp. 70–82, 2009.

[11] ——, "Detection and Mitigation of Localized Attacks in a widely Deployed P2P Network," *Peer-to-Peer Networking and Applications*, vol. 6, no. 2, pp. 155–174, 2013.

[12] A. Singh et al., "Eclipse Attacks on Overlay Networks: Threats and Defenses," *In Proc. INFOCOM*, pp. 1–12, 2006.

[13] M. Kohnen et al., "Conducting and Optimizing Eclipse Attacks in the Kad Peer-to-Peer Network," *LNCS*, vol. 5550, pp. 104–116, 2009.

[14] T. Locher et al., "Poisoning the Kad network," *LNCS*, vol. 5935, pp. 195–206, 2010.

[15] D. Germanus et al., "Susceptibility Analysis of Structured P2P Systems to Localized Eclipse Attacks," *In Proc. SRDS*, pp. 11–20, 2012.

[16] M. Steiner, T. En-Najjary, and E. W. Biersack, "A Global View of KAD," *In Proc. SIGCOMM Conference on Internet Measurement*, pp. 117–122, 2007.

[17] J. Yu, C. Fang, J. Xu, E.-C. Chang, and Z. Li, "ID Repetition in KAD," *In Proc. Peer-to-Peer Computing*, pp. 111–120, 2009.

[18] B. Heep, "R/Kademlia: Recursive and Topology-aware Overlay Routing," *In Proc. ATNAC*, pp. 102–107, 2010.

[19] I. Baumgart and S. Mies, "S/Kademlia: A practicable Approach towards Secure Key-based Routing," *In Proc. ICPADS*, pp. 1–8, 2007.

[20] I. Baumgart et al., "OverSim: A Flexible Overlay Network Simulation Framework," *In Proc. INFOCOM*, pp. 79 – 84, 2007.

[21] M. Castro et al., "Secure Routing for Structured Peer-to-Peer Overlay Networks," *SIGOPS Oper. Syst. Rev.*, pp. 299–314, 2002.

[22] E. Oh and J. Chen, "Parallel Routing in Hypercube Networks with Faulty Nodes," *In Proc. ICPADS*, pp. 338–345, 2001.

[23] A. Nambiar and M. Wright, "Salsa: A Structured Approach to Large-scale Anonymity," *In Proc. CCS*, pp. 17–26, 2006.

[24] R. Dingledine et al., "Tor : The Second-Generation Onion Router," *In Proc. USENIX Security Symposium*, 2004.

[25] P. Mittal and N. Borisov, "ShadowWalker: Peer-to-peer Anonymous Communication using Redundant Structured Topologies," *In Proc. CCS*, pp. 161–172, 2009.

[26] M. Schuchard et al., "Balancing the Shadows," *In Proc. Workshop on Privacy in the Electronic Society*, pp. 1–10, 2010.

[27] N. S. Evans and C. Grothoff, "R5n: Randomized Recursive Routing for Restricted-Route Networks," *In Proc. NSS*, pp. 316–321, 2011.

[28] M. Steiner and E. W. Biersack, "Crawling Azureus," *Institut Eurecom, France, Tech. Rep. EURECOM*, vol. 2495, no. 06, 2008.

[29] A. Avizienis, "The N-Version Approach to Fault-Tolerant Software," *IEEE Transactions on Software Engineering*, vol. SE-11, no. 12, pp. 1491–1501, 1985.

[30] G. Pongor, "OMNeT: Objective Modular Network Testbed," in *In Proc. MASCOTS*, 1993.

[31] Z. Yao, D. Leonard, X. Wang, and D. Loguinov, "Modeling Heterogeneous User Churn and Local Resilience of Unstructured P2P Networks," *In Proc. ICNP*, pp. 32–41, 2006.