Detecting and Mitigating P2P Eclipse Attacks

Hatem Ismail
DEEDS Group, TU Darmstadt
Darmstadt, Germany
hayman@deeds.informatik.tu-darmstadt.de

Daniel Germanus
ENX Association
Frankfurt, Germany
daniel.germanus@enx.com

Neeraj Suri
DEEDS Group, TU Darmstadt
Darmstadt, Germany
suri@cs.tu-darmstadt.de

***Abstract*—Peer-to-Peer (P2P) protocols increasingly constitute the foundations for many large-scale applications as the inherently distributed nature of P2P easily supports both scalability and fault-tolerance. However, the decentralized design of P2P also exposes it to a variety of distributed threats with Eclipse Attacks (EAs) being a prominent type to impact P2P functionality. While the basic technique of divergent lookups has been demonstrated for suitability to mitigate EA, it can only (effectively) address limited variants of EAs.**

This paper investigates both the detection and mitigation potential of enhanced divergent lookups for handling complex EA scenarios. In addition, we propose an approach that can identify malicious peers with a high degree of accuracy. Our simulations have shown EA mitigation rates of up to 96% in case 25% of the peers are malicious. Also, our approach allows for anonymity-fostering, fully decentralized usage, and facilitating downstream mechanisms such as malicious peer removal.

***Index Terms*—P2P, Eclipse Attacks**

I. Introduction

Peer-to-Peer (P2P) computing is an established paradigm used across a variety of data dissemination and data discovery applications. Originally applied for file sharing applications, it is increasingly utilized for diverse large-scale networked applications such as social networks, multimedia streaming, sensing and control, vehicular networking and IoT applications, where the underlying P2P mechanisms also need to be resilient to encountered perturbations.

P2P networks inherently provide good fault-tolerance due to their design approach of redundant message exchange and replicated data storage. Moreover, the decentralized protocol design requires only partial views of the network and thereby facilitates their scalability. Yet, the partial view of each peer on the P2P overlay network also introduces susceptibilities to various attacks [1], [2], [3], [4], [5].

This work focuses on Eclipse Attacks (EA) and especially the class of Localized EA's (LEA) that are known to have a significant impact on P2P functionality [6], [7] of availability, integrity, and confidentiality. Moreover, no generic LEA mitigation technique has been found that also preserves the properties of scalability, decentralization, openness, and timeliness.

Recently, divergent lookups have been proposed as an effective mitigation technique for a variant of localized EA, known as **Topology Aware LEA (taLEA)**, [8], [9], where the attack efficiency stems from very selective placement of malicious peers in the victim peers' vicinity. taLEA depends on knowledge of the topology and overlay protocol to place very few but very carefully placed malicious peers to cause damage. Unlike the niche taLEAs, LEAs form the more general attack case with malicious peers scattered all over the overlay network. While more malicious peers are required to achieve an impact comparable to a focused taLEA scenario, the generalized topology agnostic placement of peer nodes in a LEA makes for a very easy-to-conduct high-damage attacks, and hence the need for LEA mitigation.

On this background, the contributions in this work are (i) demonstrating limitations of conventional divergent lookup mechanisms resilience to LEAs under sophisticated attacker behavior scenarios, and (ii) enhancing divergent lookups via the development of a highly accurate mechanism to detect malicious peers which is based on a dynamic voting algorithm. This enables divergent algorithms not only to mitigate selected LEA variants, but also the generic LEA attack cases. We have conducted comprehensive simulation experiments to assess our contributions considering a diverse P2P parameter landscape. Our approach shows divergent lookup mitigation effectiveness of up to 96% in LEA scenarios involving up to 25% malicious peers in the overlay network. Moreover, we are able to detect close to 100% of malicious peers where the detectability varies depending on the exact attacker behavior.

Paper Structure

The paper is organized as follows. Section II presents the technical background and related work. Section III describes the system model and the concepts underlying the technical sections covering the attacker model (Section IV), divergent lookups (PASS) (Section V), and detection mechanism (Section VI). Finally, the attack severity, mitigation efficiency, and detection rates are evaluated in Section VII.

II. Background & Related Work

A variety of EA mitigation techniques have been proposed, yet their effectiveness as a countermeasure for localized attacks is either quite limited or violates P2P aspects of scalability or decentralization. We first provide a high-level overview on relevant EA variants, followed by an overview of our mitigation approach, and a discussion on related work.

A. Eclipse Attacks (EA)

The goal of an EA [4] is to *eclipse* resources (peers or data managed by peers), i.e., prevent benign peers' service

provision or provide nefarious services by using a set of malicious peers. Peers targeted to be eclipsed are referred to as victims $v \in V$. A variety of approaches can be taken to launch EAs, and in many cases the decentralized routing mechanism is attacked. Malicious peers may collude and behave inconsistently which further complicates their detectability. Localized EA (LEA) and topology aware EA (taLEA) are two common variants of EAs which are discussed next.

B. Localized Eclipse Attacks (LEA)

LEAs [6] are a subcategory of EAs that eclipse only a subset of peers in the P2P overlay network where malicious peers are scattered through the overlay. The adversary chooses the subset, for example, based on the resources managed by the peers to be attacked.

C. Topology-Aware Localized Eclipse Attacks (taLEA)

Topology-aware LEAs [7], [8] (taLEA) are a specialized LEA variant which require only a small, fixed amount of malicious peers to launch an efficient attacks against overlay networks of arbitrary sizes. To this end, the adversary places malicious peers at specific locations in the overlay network's topology. In our previous work in [8], [9], we managed to mitigate taLEA using divergent mechanisms which are described in Section V.

D. Contributions: Detection & Mitigation

In this paper, we demonstrate three sophisticated LEA variants in Section IV. To mitigate these attacks, we assess divergent lookups [8] for their suitability in the new attack's context. We highlight how divergent algorithms are unsuitable to mitigate LEAs and how severely does the proposed adversarial LEA behaviors negatively impact the overlay stability in terms of reliability and connectivity between peers. Section V presents the main concepts of (a) divergent lookups, and (b) the P2P Address Space Slicing Technique (PASS) [9], highlighting the threats that exist from a LEA. In Section VI, we introduce the technical foundations for mitigating generic LEAs and detecting malicious peers that conduct the aforementioned LEA variants. Finally, we evaluate our approach in regard of lookup reliability and performance in Section VII.

E. Related Work

In [3], Sybil attacks were introduced where the attacker can launch an attack with a small set of malicious peers and can consequently garner multiple addresses which allows malicious peers to fake being a larger set of peers. Using Sybil attacks, authors in [10] launched a LEA via a chain of Sybil/malicious nodes. However, the attack relies on the strong assumption about the existence of a single path towards the victim. In [11], a LEA is launched using Sybil peers. Although the authors proposed a mitigation scheme, the scheme is based on a *centralized* encryption authority. Using the same concept, authors in [12] proposed adding Certificate Authorities to peers' network IDs while joining the network. Although authors in [13] proposed a mitigation scheme based on preventing malicious entities from selecting their own network IDs, the mitigation scheme is based on a signing entity that uses public key cryptography.

In [14], a mitigation mechanism is proposed based on assigning multiple paths for each lookup using disjoint paths. Nonetheless, a cryptographic scheme is used that can only be substituted by a centralized authority. In addition, in [15] a similar approach is proposed. However, messages overhead due to using multiple paths is not addressed.

Similarly, the authors in [16] highlight how publish attacks could be used to attack the KAD network which is a Kademlia based network through flooding peers' index tables close to the victim with false information which is a simplistic taLEA variant. However, they didn't provide a mitigation scheme.

In [5], a KAD network crawler is introduced to monitor the network status and detect malicious peers during a LEA. However, in a distributed P2P system, a high overhead arises if each peer uses such a mechanism to detect malicious entities. This becomes impractical as the overlay size increases.

III. System Model

This section presents the system model used in this work. It consists of an overlay model and a P2P protocol abstraction that also includes descriptions of the lookup mechanisms.

A. Overlay Network Model

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

We further detail P as benign peers B, malicious peers M and victim peers V. Moreover, $P = B \cup M \cup V$ and $B \cap M = \emptyset, M \cap V = \emptyset, B \cap V = \emptyset$, and $N = |P|$, where N is the overlay size. Peers $b \in B$ show benign behavior in the network, i.e., according to the P2P model specification and no adverse intentions. Malicious peers $m \in M$ refer to peers being controlled by an attacker and may behave adversarial. Peers targeted by the attacker are victims $v \in V$.

B. P2P Protocol Model

Our abstraction for structured P2P protocols consists of six different aspects as detailed below.

1) Address Space: Peers have a unique assigned identifier referred to as the peer's *key*. 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.

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 [17] makes use of the XOR operation to calculate the Common Prefix Length (CPL) using the bitstring representation of two peers' keys.

3) Routing Table: Each peer maintains a routing table 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 $i = 0 \dots w-1$, and k constant. To resolve new contact information a lookup call is initiated.

4) Lookup Mechanism: In case the destination peer b for a specific message to be sent by peer a is not stored in a's routing table, a lookup call is initiated to *resolve* b's contact information. A commonly applied design best practice is *convergent lookups*, i.e., peer a selects a set of known peers with closest possible distance to b, and asks each of them to either return the contact information or to repeatedly *forward* a's lookup request to even closer peers until b can either be resolved or the lookup is dropped due to a timeout.

5) Proximity: Depending on the overlay size N and the key length w, each peer defines a proximity area, typically as a close by and sparsely populated address space region. We define proximity as the set of peers that is stored in the lowest index i list of the routing table, i.e., all lists with indexes less than i are empty.

6) Short and Long Distance Edges: Two basic notions will be used throughout the paper, *long distance edges* (LDE) and *short distance edges* (SDE), as introduced in [8]. SDEs refer to edges $e \in E$ of proximate peers. Contrary, LDEs refer to edges of peers that are not located in the same proximity.

IV. LEA Attacker Models

We propose a new attack model based on behavioral patterns of malicious peers. The attack model builds upon and extends LEA (cf. Section II-B), thus, all the attack behaviors discussed at next represent the generic LEA.

To launch the attack, malicious peers $m \in M$ join the overlay and we assume they are uniformly distributed across the address space. Once a peer m receives a lookup request for the victim peer, different *attacker behaviors* can be activated. Moreover, the proposed LEA based behaviors are chosen based on security goals (availability, integrity, confidentiality) that exploit the lookup mechanism. Next, we introduce three new complex attacker behaviors that collectively represent the generic LEA behaviors:

A. Fake Destination Attacker Behavior (FD-LEA)

In FD-LEA malicious peers fake the victim's identity, which threaten the availability, confidentiality and exploit the inherent partial view of each peer about the overlay.

Technical Description: During a lookup, once a malicious peer receives a lookup request for a victim peer, it replies to the lookup initiator p_r with contact information that points to a malicious peer that fakes owning the key p_r is looking for.

Behavior Discussion: The overlay's reliability is severely affected since the lookup call terminates once a malicious peer returns a fake destination and p_r believes that the reply was sent from a benign peer that holds v's contact information. Consequently, the availability of the victim peer's service provision is negatively affected. Moreover, in case of unencrypted message payloads, the confidentiality would also be affected, as p_r sends its message to the colluding malicious peer that may subsequently inspect it.

B. Pollution Selection Attacker Behavior (PS-LEA)

In a PS-LEA behavior, malicious peers reply only with malicious contact information which threaten the availability and exploit the candidate selection mechanisms for the lookup initiator peer p_r. The main aim of the attacker during a PS-LEA is to pollute p_r's candidates selection queue which is maintained over the different lookup iterations to store contact information of peers that may be queried. Lookup iterations refers to the number of rounds where p_r sends parallel lookup requests to different peers requesting v's contact information.

Technical Description: Initially, p_r stores a list that contains all the possible candidates that could be queried in the next iterations. This list is updated after each iteration from other queried peers that have no knowledge about v. The selected candidates set sent to p_r are selected according to the lookup algorithm used within the overlay.

Behavior Discussion: Once peer m receives a lookup request for a victim peer, only colluding malicious peers located all over the address space are returned. Hence p_r contacts malicious peers in the next iterations until the lookup request times out after i_{max} iterations. Similarly, the availability of the victim peer's service provision is negatively affected.

C. Mixed Attacker Behavior (FD-PS)

The third attacker behavior is a combination of the previous two, whereas a probability parameter is the basis for a switching decision between the proposed adversarial behaviors. Such a sophisticated attacker behavior has not been considered in previous work [7], [8] so far.

Technical Description: For mixed FD-PS LEA behavior, the attacker chooses weights for the probability of either behavior to be active, and behaviors may be subject to a switch in-between different lookup iterations.

Behavior Discussion: The impact of this behavior on the victim peers is the same as discussed before for the individual attack behaviors. However, activating both attacks with different weights helps vary the degree of perturbation that can be caused by each individual attack.

V. Divergent Lookups using P2P Address Space Slicing

Divergent lookups have been proposed as a suitable taLEA mitigation technique in our previous work [8]. In a nutshell, divergent lookups avoid searching the destination peer's proximity to skip out on querying malicious peers under taLEA assumptions. Also, divergent lookups match the mitigation requirements described beforehand. In this work, we assess the mitigation potential of divergent lookups for the more generic LEA variant. We briefly describe divergent lookups [8], [9].

A. PASS Preliminaries

Divergent lookups segregate the address space into *CPL slices*, i.e., creating equivalence classes according to the CPL peers share with the destination. The technique is called *P2P address space slicing* (PASS) and requires two more threshold parameters, namely upper t_u and lower threshold t_l. We define $0 \leq t_l \leq t_u \leq t_p \leq w$ with t_p being the proximity threshold. Divergent lookups that make use of PASS, detailed in our previous work in [9], try to resolve the destination's contact information from peers in the CPL slice interval $[t_l, t_u]$ because other intervals, as discussed at next, are suboptimal:

- $[0, t_l)$: This range in the address space contains a large amount of peers, divergent lookups in that range tend to yield a bad performance or even timeout.
- (t_u, t_p): This range contains so called *dead ends* which represent peers that cannot reach the destination, i.e., no path towards the destination based on contact information of neighbor peers can be found. Running divergent lookups in that range yields a low reliability.
- $[t_p, w)$: This range is populated with malicious peers under taLEA, therefore to be avoided by the lookup. Otherwise, reliability would significantly decrease.

B. divPASS susceptibility to LEA

Nevertheless, in that context, launching FD-LEA, PS-LEA or mixed FD-PS LEA on the selected CPL range (t_u, t_p), can severely degrade divPASS performance and reliability. PS-LEA behavior can simply (i) send the set of malicious peers within the CPL as possible candidates to p_r, (ii) divert the set of possible candidates outside of the suitable CPL range selected by divPASS, or even (iii) divert the request towards *dead end* peers. Similarly, malicious peers launching FD-LEA block the request from reaching to benign peers within the selected CPL that might have an LDE towards the destination.

Although lookups may be executed in parallel to improve on fault-tolerance and timeliness, divergent lookups are still susceptible to LEA with its variants as evaluated in Section VII. Obviously, the set of results can differ due to several malicious and benign causes. Two detection mechanisms presented in the next section have been designed to deal with such inconsistencies and allow to identify malicious peers that conduct LEAs (with FD and PS attacker behaviors).

VI. Detection Mechanisms

We propose two differing detection mechanisms as (i) lookup result voter, and (ii) lookup reply investigation. The first mechanism analyzes the result set after the lookup completion and detects LEAs with the FD attacker behavior. The second detection mechanism assesses, after each iteration, the lookup's candidate list to detect LEAs with PS attacker behavior. Both variants have been integrated into divPASS and will be evaluated in the subsequent evaluation section.

A. Lookup Result Voter Mechanism

The obvious reasons for lookup result inconsistencies are overlay perturbations such as ongoing attacks, outdated routing table entries, or perturbations in the underlay network. In order to detect inconsistencies, we use a *dynamic majority voter* (DMV) [18]. DMVs are used to assess a set of inputs and thus, decide whether a valid output exists or not, where valid denotes a non empty majority of matching inputs. Basically, using DMV allows to (i) ensure reliable lookup operation in perturbed overlays, and (ii) identify maliciously behaving peers. The DMV can process up to α different inputs, which we group in three classes: *correct*, *fake*, or *no LDE towards the victim*.

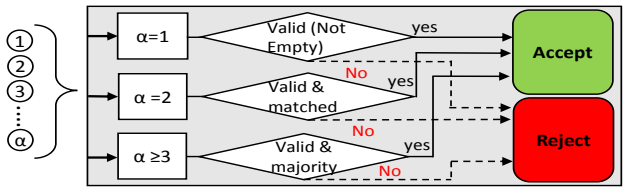


Fig. 1. Acceptance cases for the DMV.

In the following subsection, we discuss the DMV's operation with a focus on the reliable selection process from an inconsistent result set.

B. DMV Operation

Initially, p_r initiates a divergent lookup with α parallel requests. Once a peer replies with an LDE towards the victim p_d to p_r, the lookup terminates and the result is evaluated. Nevertheless, to allow the DMV to process a set of results, we modified the divPASS algorithm such that lookups wait for maximum i_{max} iterations until up to α replies are available.

Due to the probability that no different α benign peers have LDEs towards p_v, a maximum of c correct replies are returned to p_r. Furthermore, a fraction f_m of malicious peers within the specified CPL might intercept the lookup, which in turn will return f_m fake replies to p_r. In addition, n_e empty replies can be returned back to p_r due to (i) dropped replies and (ii) the lookup request can neither be intercepted by a malicious peer nor a benign peer have an LDE to p_d until the maximum number of iterations i_{max} is reached. To that end, the maximum number of replies that can be passed as an input to the DMV is $\alpha = c + f_m + n_e$. The next step is that the α received replies are passed as inputs to the DMV, which in turn decides whether to accept or reject the results, as shown in Figure 1, based on the following cases:

1) $\alpha \geq 3$: the DMV checks if a majority of a valid results is available, i.e., either $c > f_m$ or $c < f_m$.
2) $\alpha = 2$: the voter returns the contact information of index α as a valid reply in case: (i) both replies are identical and (ii) reply $\alpha \notin R_e$, where R_e is the set of empty replies. Otherwise, the DMV rejects the results.
3) If $\alpha = 1$ and reply $\alpha \notin R_e$: the DMV returns the only available reply and assumes it to be valid.

C. Lookup Reply Investigation

As mentioned in Section IV, the lookup replies of malicious peers that conduct a LEA with PS attack behavior contain contact information about other colluding malicious peers.

Accordingly, in order to mitigate such attack while providing a detection feature to such malicious adversarial behavior, we propose an additional detection mechanism to assess further lookup replies before inserting them into the candidate selection list for subsequent iterations.

Technically, investigating received lookup replies, before inserting possible candidates in p_r's selection list, is based on (i) detecting peers whose replies point to peers outside of the CPL range $[t_l, t_u]$, and (ii) assuring that no candidates outside of the specified CPL can be inserted in the possible candidates list. Moreover, each peer is allowed only once in the candidate selection list. As a consequence, malicious peers that keep on sending malicious replies within the specified CPL range cannot excessively load the candidate selection list. At next, we assess our mitigation and detection schemes for the proposed attacks in a comprehensive simulation case studies.

VII. Evaluation

In this section, we assess the performance and reliability of divPASS. To do so, we integrated it with our detection mechanism. For evaluation, we present four study cases:

1) **FD-LEA impact on divPASS:** evaluation of the impact of LEA using FD behavior on divPASS.
2) **Voting mechanism against LEAs:** divPASS resilience assessment after integrating the DMV as a mean to mitigate LEA launched with FD adversarial behavior.
3) **LEA impact on divPASS using FD-PS behavior:** evaluation of LEA impact launched with FD-PS attacker behavior on divPASS.
4) **Detection and mitigation mechanisms for FD-PS:** assessment on how our detection and mitigation techniques perform in a FD-PS behavior scenario during LEA.

Firstly, we detail the simulation environment, parameters, different models and the metrics used throughout the experiments. Secondly, we present each of the above mentioned case studies, results and an interpretation of our observations.

A. Simulation Environment

Simulations were carried out using the OMNeT++ simulator [19] and OverSim [20] that provides various P2P protocol implementations for the simulator environment. In order to validate our results, each simulation was scheduled for 4 hours runtime and 12 repetitions were conducted to allow for confidence interval computation. The simulation parameters used in conducting experiments are presented in Table I.

Parameter	Value	Parameter	Value
i_{max}	10	α	5
w	128	t_p	80
t_l	4	t_u	6
MP	$5\%, 15\%, 25\%$	Q_{max}	50

TABLE I
Simulation parameters

B. Simulation Workload Model - Fully Distributed Application

In our simulation workload model, peers send lookup messages looking for random peers, on average every 10 seconds with a standard deviation of 5 seconds.

C. Simulation Churn Models

In our experiments, different churn models are used which are described below. Churn refers to the rate peers join and leave the overlay. **NoChurn:** refers to a static overlay where peers never leave the overlay once they have joined. **Pareto (P-7200):** Using the Pareto churn model, peers acquire an average lifetime and a dead time of 7200 seconds according to a Pareto distribution which gives a more realistic overview to real life scenarios [21].

D. Simulation LEA Model

A central LEA parameter that we will refer to in the experiments' result discussion is **Malicious Peers per CPL (MP)**. It reflects the average number of malicious peers for a given divPASS CPL region. This metric provides insights about the severity of LEA attacks for an increasing amount of malicious resources. Data collection occurs at periodic intervals for each simulation run to assure the representativeness of the metrics measurements. To address the severity of the proposed adversarial behaviors according to the attacker's available resources, each scenario is simulated where different amounts of malicious peers per CPL, MP, are inserted.

E. Evaluation Metrics

*1) **Lookup Success Ratio (LSR):*** measures the average ratio of successful lookups over all lookups destined to victim peers which provides insights about the accuracy of the voting mechanism's decision.

*2) **Message Complexity (MC):*** is the average number of messages exchanged per lookup process until either α replies or i_{max} is reached. This metric is used to provide message overhead calculations for a given lookup.

*3) **Number of Iterations (NoI):*** provides the average number of iterations a given lookup requires to reach α replies which gives an approximation about the average latency of a given lookup request.

*4) **Malicious Detection Rate (MDR):*** provides the average number of detected malicious peers per lookup. MDR evaluates the accuracy and scalability of the detection mechanism.

F. Case Study 1: LEA impact on divPASS using FD behavior

This case study evaluates the impact of LEA using FD-LEA adversarial behavior to highlight the unsuitability of divPASS to mitigate generic LEA in terms of performance and resiliency without the mitigation and the detection mechanisms. Results are evaluated based on LSR, MC, and NoI. As we are evaluating the performance of divPASS under LEA, data is collected only for lookups destined to the victim. We start by describing the experimental results depicted in Figures 2 through 5, and we close each case study with a detailed interpretation of the results.

Discussion of the results: Figure 2a shows LSR of lookups compared to different overlay sizes $N = 5000, 10000, 20000$ and different malicious peers ratios per CPL, i.e., $MP = 5\%, 15\%, 25\%$. As shown, LSR degrades when increasing MP since the probability of intercepting the lookup request by a malicious peer increases. For $MP = 5\%, 15\%, 25\%$, LSR

values average between 63% and 91%. This is a significant LSR decrease compared to the divPASS performance in a benign overlay (i.e., $MP = 0$) which results in a LSR between 91% and 100%.

In Figure 2b, MC for divPASS average between 7.5 and 11 for different sizes of N and regardless of MP ratio.

Figure 2c shows NoI results in the range from 1.38 to 1.74 regardless of different choices for N and MP. This means that for a successful lookup, less than two iterations are required to find a peer with an LDE to the victim. Compared to other convergent and divergent lookup algorithms [8], divPASS tends to provide low latencies as a consequence of the low NoI required for successful lookups.

Interpretation of the results: LSR decreases as a consequence of fake destination replies. The reason is that a lookup terminates once a peer replies with an LDE to p_r or when i_{max} is reached; LSR decreases for larger choices of MP.

A major advantage of divPASS is PASS's CPL region choice, such that it tends to resolve peers with LDEs to the destination with high probability. Accordingly, NoI shows low values due to the high probability in contacting a peer that replies with an LDE to the victim. In turn, the number of messages exchanged decreases as only few iterations and peers are contacted until an LDE is found. In addition, terminating the lookup once a peer has sent an LDE reply is a major reason for the low lookup MC.

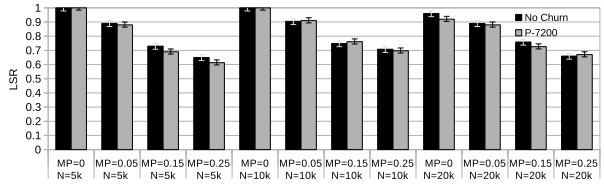
Nevertheless, the results clearly show how divPASS with no additional mitigation and detection mechanisms is susceptible to generic LEA. LSR is severely degraded since lookup results depend only on the first reply. So, we conclude here that although keeping the MC and NoI to minimum is favorable, it imposes a reliability issue for the divPASS algorithm, as shown in Figure 2a. To that end, in the next case study we assess our mitigation technique by deviating from the FD-LEA behavior while maintaining a high divPASS LSR and low MC/NoI.

G. Case Study 2: Voting mechanism DMV against LEA attacks

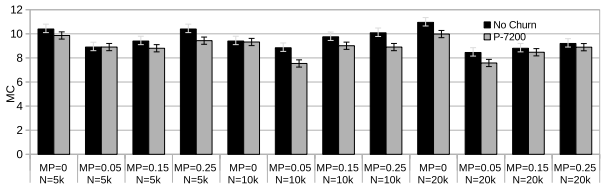
In this case study, we evaluate divPASS's performance after integrating the DMV to evaluate the enhancements of divPASS performance. For better comparison with case study 1, we make use of the same parameter choices for N and MP.

Discussion of the results: LSR results are shown in Figure 3a with LSR average values ranging from 83% to 98% which is a remarkable LSR increase compared to case study 1. Figure 3b presents MC for divPASS in combination with the integrated DMV. MC during a lookup average between 17.5 and 26 which is relatively higher than MC values in case study 1 without DMV. As shown in Figure 3c, a successful lookup demands NoI between 2.14 and 2.83. NoI results without mitigation were lower in the first case study, where values averaged between 1.38 to 1.74.

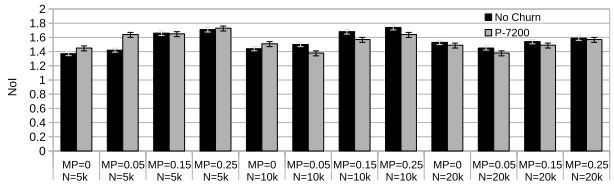
Interpretation of the results: The remarkable enhancement in the LSR values is due to integrating the mitigation mechanism through the DMV. Basically, due to the assessing criteria of the voter, chances of picking a correct reply is considerably high even in case where $MP = 25\%$. We note that, increasing



(a) Lookup success rate (LSR)



(b) Message complexity (MC)



(c) Number of iterations (NoI)

Fig. 2. FD-LEA impact on divPASS using different MP.

MP escalates the probability of malicious peers to get picked and thus, the number of fake replies increases which in turn degrades the system's reliability.

The noteworthy MC increase compared to case study 1 is due to the fact that the voter maintains the lookup process until α replies are received, which is not the case in case study 1 where the lookup terminates once the first resolving reply is received. Due to divPASS's approach to query only a certain CPL range which expectedly contains a high percentage of peers with LDEs to the destination, the NoI required to receive α replies are very small, i.e., between 2 to 3 as observed from Figure 3c. As a result, due to low NoI needed to reach α replies, MC ranges provide an acceptable increase compared to case study 1 where only a single reply is required.

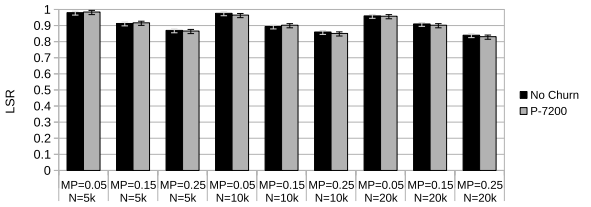
From the results, we assert that divPASS algorithm combined with the DMV provides a very good performance in mitigating LEAs as it provides high LSR values while keeping MC and NoI minimized compared to case study 1.

Our mitigation model for high LEAs shows that divPASS, in conjunction with the proposed mitigation mechanism, is scalable to maintain overlays with thousands-millions of peers.

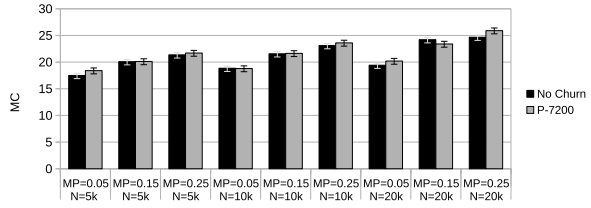
H. Case Study 3: LEA impact using weighted FD-PS impact

In this case study, we assess the impact of launching a LEA using weighted FD-PS to show the effect of the proposed attack behavior combinations on divPASS based lookups.

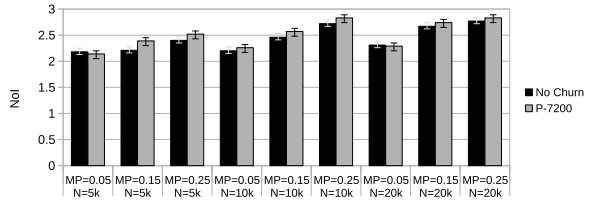
Here FD and PS are weighted equally, i.e., the probability that a malicious peer will choose an FD-LEA or a PS-LEA behavior is 0.5. Same overlay sizes $N = 5000, 10000, 20000$ and same ratios of malicious peers per CPL $MP = 5\%, 15\%, 25\%$ are used in these experiments. We note that scales for compa-



(a) Lookup success rate (LSR)

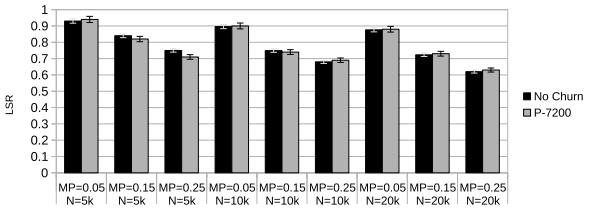


(b) Message complexity (MC)

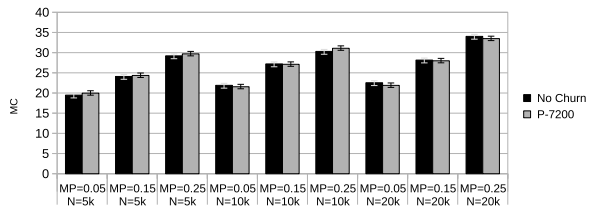


(c) Number of iterations (NoI)

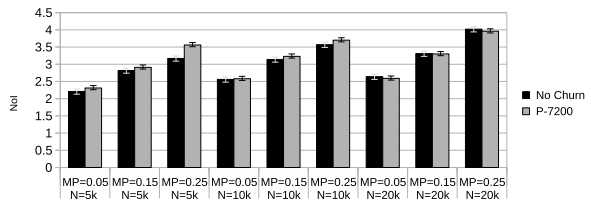
Fig. 3. Combined divPASS/DMV performance with different values for MP.



(a) Lookup success rate (LSR)



(b) Message complexity (MC)



(c) Number of iterations (NoI)

Fig. 4. Baseline results for FD-PS without detection.

rable figures may vary due to distant ranges of values as can be seen in NoI values between Figure 4c and Figure 5c.

Discussion of the results: Figure 4a shows the LSR for different overlay sizes and MP ratios. Values range between 63% and 94%. Obviously, when MP increases, the LSR value decreases accordingly as more malicious peers are able to intercept the parallel lookup requests sent by p_r. In Figure 4b, we notice that a remarkable MC increase stems from the message exchange until α replies are received; MC values range between 19 and 34 messages. Figure 4c shows an increase for NoI, i.e., values range from 2.2 to 4. Compared to case study 2 where only FD-LEA are launched, the NoI increased 30% in the weighted FD-PS LEA.

1) Interpretation of the results: The noticed LSR decrease occurs due to the combined effect of both FD-LEA and PS-LEA behaviors which can be summarized as follows: (i) the impact of fake replies that are sent to the voter and (ii) the increment of malicious peers' ratio due to PS-LEA effect where malicious peers intentionally insert more malicious entities into p_r's candidate list. Moreover, due to (ii), NoI required for a successful lookup increases as the number of malicious peers inside the candidate selection list increases. As a result, the probability of picking more malicious peers for the next rounds increases, which in turn forces the divPASS algorithm to run more iterations until α replies are received.

Increasing the NoI has an impact on the average number of messages exchanged during a lookup as it requires contacting more peers. Accordingly, MC increases per lookup. To that end, we conclude the unsuitability of divPASS with no detection mechanism to mitigate generic LEAs.

We note that according to our observations from running experiments for different weights, Increasing the weight of PS behavior have a direct impact on the average NoI and MC which is the target of PS attacks. Meanwhile, increasing the weight of FD-LEA behavior impacts negatively on the LSR values. For instance, running the same experiment with an FD-LEA weight of 0.25 and a PS-LEA weight of 0.75, we achieve results with MC= 47 and NoI= 5.37.

I. Case Study 4: Detection and mitigation mechanisms response against weighted FD-PS behaviors

Now, we assess the performance of divPASS when integrating the proposed mitigation and detection techniques against FD-PS LEAs. We evaluate the results compared to the previous case study where no detection mechanism were integrated.

Discussion of the results: In Figure 5a, LSR values average between 90% and 99% which is a noticeable increase compared to case study 3, even in scenarios where $MP = 25\%$, where LSR averaged between 63% and 94%. Figure 5b shows the average MC where values range between 18.5 and 23. In this scenario, average MC are relatively similar to case study 2, where only FD-LEA based attack is running. Moreover, in Figure 5c, average NoI average between 2.1 and 2.4 which is again relatively similar to case study 2 where no malicious peers launched a PS-LEA. A noticeable decrease is noted, comparing NoI and MC values to case study 3. Since we aim to evaluate the efficiency of our detection and mitigation mechanism, we evaluate the MDR per lookup. In Figure 5d,

MDR averaged between 0.55 and 6 depending on N and MP which provide insights about the average number of malicious peers that launch a PS-LEA contacted during a lookup.

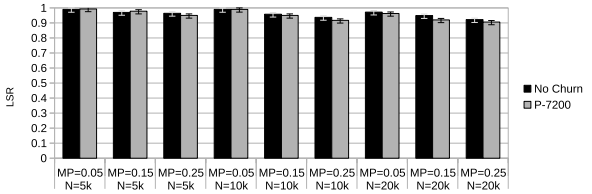
1) Interpretation of the results: After DMV integration with divPASS, a remarkable LSR enhancement can be noticed. This is due to the fact that detecting malicious peers before inserting malicious entities decreases the probability of polluting the candidate list which in turn reduces the probability of launching FD-LEA by other malicious peers. Moreover, once a peer is detected, the MP value decreases which in turn enhances the chance of p_r to contact benign peers. For the same reason, NoI decreases since α replies can be collected in less number of iterations. Accordingly, the number of messages that needs to be exchanged during a lookup decreases.

For high values for MP, the detector shows that all malicious peers that manifest a PS-LEA behavior contacted during a lookup call are detected. In fact, MDR values underline the detector's positive impact on LSR, NoI, and MC.

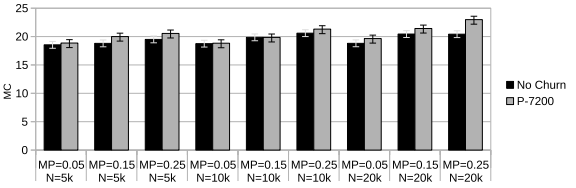
We conclude that the combination of our detection and mitigation mechanisms yields excellent reliability and performance in the presence of FD-PS LEAs. Also, it is a preparatory step to achieve reliable and decentralized malicious peer removal from overlays.

VIII. Conclusion & Future Work

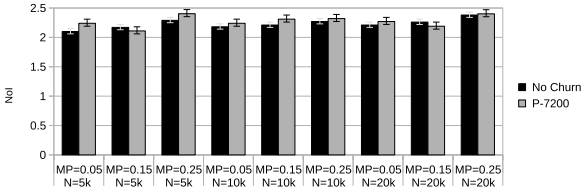
Localized Eclipse attacks (LEA) pose a significant threat to P2P-based applications. We extend the divergent lookup mechanism, which was originally developed to mitigate the specialized topology-aware Eclipse attack (taLEA), to mitigate the more generic LEA. Moreover, we have defined a sophisticated attacker model which causes significant decreases in reliability and performance in divergent lookups. Consequently, we integrated a new detection mechanism which also helps identify attacking peers with high accuracy. As ongoing work, we are developing a decentralized routing table sanitizing mechanism and assessing its performance in real networks.



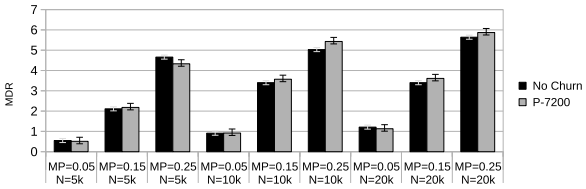
(a) Lookup success rate (LSR)



(b) Message complexity (MC)



(c) Number of iterations (NoI)



(d) Malicious detection rate (MDP)

Fig. 5. Detection performance of FD-PS based LEAs.

References

[1] H. Lin et al., "Conducting Routing Table Poisoning Attack in DHT Networks," *In Proc. ICCCAS*, pp. 254–258, 2010.

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

[3] John R. Douceur, "The Sybil Attack," *In Proc. Peer-to-Peer Systems*, pp. 251–260, 2002.

[4] A. Singh, M. Castro, P. Druschel and A. Rowstron, "Defending Against Eclipse Attacks on Overlay Networks," *In Proc. SIGOPS*, pp. 115–120, 2004.

[5] T. Cholez et al., "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.

[6] A. Singh, T. Ngan, P. Druschel and D. Wallach, "Eclipse Attacks on Overlay Networks: Threats and Defenses," *In Proc. INFOCOM*, pp. 1–12, 2006.

[7] D. Germanus, R. Langenberg and A. Khelil and N. Suri, "Susceptibility Analysis of Structured P2P Systems to Localized Eclipse Attacks," *In Proc. SRDS*, pp. 11–20, 2012.

[8] D. Germanus S. Roos, T. Strufe and N. Suri, "Mitigating Eclipse Attacks in Peer-to-Peer Networks," *In Proc. CNS*, pp. 400–408, 2014.

[9] D. Germanus, H. Ismail and N. Suri, "PASS: An Address Space Slicing Framework for P2P Eclipse Attack Mitigation," *In Proc. SRDS*, 2015. [Online]. Available: www1.deeds.informatik.tu-darmstadt.de/External/PublicationData/1/srds-2015-germanus.pdf

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

[11] M. Steiner et al., "Exploiting kad: Possible uses and misuses," *SIGCOMM Computer Communication Review*, pp. 65–70, 2007.

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

[13] M. Leonardo et al., "Avoiding Eclipse Attacks on Kad/Kademlia: an Identity Based Approach," *In Proc. ICC*, 2009.

[14] I. Baumgart and S. Mies, "S/Kademlia: A Practicable Approach Towards Secure Key Based Routing," *In Proc. ICPADS*, pp. 1–8, 2007.

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

[16] T. Locher, D. Mysicka, S. Schmid and R. Wattenhofer, "Poisoning the Kad Network," *LNCS*, vol. 5935, pp. 195–206, 2010.

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

[18] R.E. Gantenbein, S. Sung Yun and J.R. Cowles, , "Evaluation of Combined Approaches to Distributed Software Based Fault Tolerance," *In Proc. PRDC*, pp. 70–75, 1991.

[19] Pongor, György, "OMNeT: Objective Modular Network Testbed," *In Proc. MASCOTS*, 1993.

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

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