Attack Resistant Network Embeddings for Darknets

Benjamin Schiller and Stefanie Roos and Andreas Höfer and Thorsten Strufe
P2P Networks, Dept. of Computer Science, TU Darmstadt
[schiller, roos, a.hoefer, strufe][at]cs.tu-darmstadt.de

***Abstract*—Darknets, connecting only devices between participants of mutual trust in the real world, rely on cooperative, precise, and attack resistant embeddings to evolve routing structures on the name space.**

Only precise embeddings allow for performant communication with low overhead on these networks. With Darknets being deployed in generally untrusted, even adverse environments, external or internal attacks have to be assumed commonplace. Their impact hence has to be limited and the embedding must be made resistant against even sophisticated attacks.

Analysing Dark Freenet, the only current approach implementing a full Darknet, we devise simple attacks that render its embedding entirely corrupt. In response we derive a novel embedding that is based on local decisions only, and which not only is resistant to such attacks, but additionally outperforms the Dark Freenet in terms of precision.

***Index Terms*—Networks, Distributed Computing, Distributed Control, Availability, Security**

I. Introduction

Friend-to-Friend networks, often called Darknets, are a type of overlay networks to enable anonymous communication in the Internet. To conceal network membership, connections in these networks are restricted to trusted relationships – i.e. an identity-revealing IP connection between two network nodes is only established when they know and trust each other.

The Dark Freenet as introduced in [1] is an example of a Friend-to-Friend network. It offers a data publication system based on trusted connections. The Dark Freenet applies a greedy routing algorithm based on IDs which are assigned to the nodes by a distributed embedding algorithm. The IDs are at first drawn uniformly at random and are then swapped between nodes to optimize the embedding and improve the routing performance.

In this paper we examine the resilience of network embeddings for Darknets against adversarial nodes. We devise attacks on Dark Freenet's ID swapping procedure and study their effects on the embedding. A new embedding algorithm based on Markov chains is introduced and its performance under attacks is analyzed. It offers two major improvements with respect to security: Users are not allowed to actively influence the IDs of others. Secondly, IDs are not given to anyone but trusted peers. The evaluation of the algorithms and the attacks is done by simulating them in graph theoretical models of the networks. We show that the newly proposed algorithm offers a better routing performance and is less affected by the attacks compared to Dark Freenet's approach, at least in the initial experiments presented here.

This work in parts has been supported by the IT R&D program of MKE/KEIT of South Korea (10035587, Development of Social TV Service Enabler based on Next Generation IPTV Infrastructure).

The remainder of this paper is structured as follows: Section II gives some background on embedding and routing in Dark Freenet. Section III discusses the quality criteria of an embedding and the model we use to evaluate the algorithms and attacks. Sections IV and V present Dark Freenet's embedding algorithm and the newly proposed embedding algorithm, respectively. In Section VI we describe three attacks targeting different properties of the embedding. The evaluation setup, the tested hypotheses, and the obtained results are presented in section VII. Section VIII concludes and states future work.

II. Embedding and Routing in Freenet

Mapping resources and nodes in the same name space and constructing a routing structure by careful neighbor selection at all nodes, a greedy routing minimizing distances in the name space can be implemented in distributed hash tables (DHT), and resources published or found at nodes corresponding to their key in the namespace (cf., e.g., [2]). This approach requires to structure the network in a way that allows mapping data items to nodes and finding the respective nodes afterwards. Structuring the network is done by creating links between nodes in a way that makes greedy routing, i.e. forwarding a request to the neighbor that is closest to the destination, possible and reasonably fast. This strategy is not possible in Darknets, since links are based on trust relationships and cannot be freely created. IDs, however, can be freely assigned to nodes in a Darknet; therefore, IDs are assigned to nodes in a way that optimizes the routing performance. This ID assignment procedure is called network embedding.

Freenet's approach to enable deterministic routing under Darknet conditions was introduced by Sandberg [3] based on the small world model by Kleinberg [4]. In Kleinberg's model nodes are arranged in a d-dimensional lattice. Additionally each node u is given a long-range neighbor. The probability for choosing v as long-range neighbor of u is proportional to $\frac{1}{m(u,v)^d}$ where $m(u,v)$ denotes the Manhattan distance on the lattice. Kleinberg showed that in this model greedy routing is possible in $\log^2 n$ steps where n is the side length of the lattice. The idea of Sandberg is to assign each Darknet node an ID representing a position on the unit ring ($d = 1$), so that the edge lengths in the graph are distributed in the same way as the long-range links in Kleinberg's model. However, it is in general not possible to connect every node to its direct

predecessor and successor on the ring, so instead of actual greedy routing a weighted depth-first search is applied, where the message is always forwarded to the neighbor closest to the destination that did not receive the message before. In case no such neighbor exists, backtracking is used.

An analysis of Freenet's embedding algorithm and its resilience in particular has be done in [5]. The authors study the effects of attacks and churn on the uniformity of the ID distribution and therewith on load balancing. For skewing the ID distribution towards a clustered distribution where all nodes gather around a small number of clusters the attacker uses Freenet's swapping mechanism to spread IDs to its neighbours. By choosing IDs from one cluster the attacker can skew the ID distribution towards the cluster. Due to the non-uniform ID distribution some nodes become responsible for large chunks of the ID space which results in high load. In a simulation study in [5] it is shown that the attack effectively skews the ID distribution and causes data loss due to overwhelmed nodes. Further simulations show that churn has very similar effects on the ID distribution over time. Compared with our work the attack against the uniformity of the ID distribution is similar to our contraction attack (Section VI-C). Beyond uniformity we also target convergence of the algorithm and routing performance with the divergence (Section VI-B) resp. sub-optimal distance distribution (Section VI-D) attacks.

III. Preliminaries

A Darknet is represented by a static undirected graph $G = (V, E)$. Each node $v \in V$ represents a user and $E = \{\{u_1, v_1\}, \dots \{u_m, v_m\} : u_1, \dots u_m, v_1, \dots, v_m \in V\}$ is the set of edges. $N(v) = \{u \in V : \{v, u\} \in E\}$ is the set of neighbors of v, $deg(v) = |N(v)|$ denotes the degree of v. Messages can be exchanged between nodes u and v if and only if there is an edge $e \in E$ which is incident to both u and v. Each node is assigned an ID $id : V \to [0, 1)$, representing a position on the unit circle. An ID selected uniformly at random is denoted by id_R. All computations involving IDs are done $mod\ 1$. The distance between two IDs id_1 and id_2 is defined as their minimal distance in both directions on the unit circle. The distance between two nodes u and v is then given by $d(u, v) = d(id(u), id(v))$. The length $l(e)$ of an edge $e \in E$ is the distance of its incident nodes. The successor $succ(u)$ of u is then defined as the node with the closest ID in clockwise direction.

A. Qualities of an embedding

1) Partition of ID space: We assume that the keys associated with stored data items are uniformly distributed in the whole ID space. Therefore it would be beneficial, if the node IDs are chosen uniformly so that the distances $d(v, succ(v))$ are normally distributed with mean $1/|V|$. Then the content should also be distributed uniformly between nodes, so that no small set of nodes is responsible for an inappropriate number of data items.

2) Distance distribution: As detailed in Section II Kleinberg has shown that a decentralized algorithm can find short paths between nodes using only local knowledge if the distances have a simple rational distribution. It is therefore favorable to achieve a distance distribution similar to the one in Kleinberg's model in order to enable routing using a decentralized algorithm.

3) Topological closeness to successor: As stated in Section II it is usually not possible to connect every node with its successor, but still it is preferable that a node and its successor are topologically close since greedy routing relies on the correlation between distance in the ID space and hop distance (which directly follows if all successors are topologically close). While an embedding cannot guaranteed local links as needed for Kleinberg's proof, it should provide a set of local connections that can be used when routing.

4) Routing performance: The routing performance of a system can be measured by two main characteristics: success rate and average path length. The success rate specifies the percentage of routing attempts that successfully reach the destination. While it is very important that a large percentage of routing attempts reach their destination, the number of hops in the overlay plays a crucial part in the usability and perceived performance of a system. In order to exhibit a good routing performance, a distributed system needs to maximize the success rate and minimize the number of hops required to reach a specified destination.

B. Adversary Model

An adversary attempting to damage the system's embedding can control one node and observe all communications this node is part of. It can also inject messages to its neighbors as well as drop or modify messages it receives from neighboring nodes.

An adversary is restricted in the following form: It can neither change the topology of *G* nor observe or influence any communication that does not pass its node. An adversary does not keep state and bases all its decisions purely on the current IDs of its neighbors and the information received as part of a request. Furthermore, adversaries are unaware of each other and hence do not collaborate. Also, an adversary does not tamper with the routing itself since our focus lies on the resilience of embedding algorithms.

IV. Swapping

In this Section Sandberg's embedding from [3] is introduced. We start with the more general concept of using Markov chains for finding an embedding, before describing the actual Markov chain used by Sandberg. Each state of such a Markov chain is characterized by the current ID assignment. In each step new IDs are suggested for some of the nodes. The new IDs are accepted with a probability depending on how likely the new IDs are given the long-range link distribution in Kleinberg's model ([4]) in comparison with the old IDs: Let $l_C : E \to [0, 0.5)$ be the edge length using current ID assignment and $l_S : E \to [0, 0.5)$ the edge length using

the suggested ID assignment. The probability that a new ID assignment is accepted can be done in a distributed way. Each node calculates:

$$c(u) = \frac{\prod_{e=\{u,v\}\in E} l_C(e)}{\prod_{e=\{u,v\}\in E} l_S(e)}$$

The new state is then accepted with probability $\min\{\prod_{u\in V} c(u), 1\}$.

This means that an edge length distribution that is closer to the ideal distribution is always accepted, while a worse distribution is accepted with a certain probability to escape a local optimum. In case of a symmetric selection kernel this corresponds to the Metropolis-Hastings algorithm [6].

Note that only nodes changing their ID need to be involved in the calculation process (otherwise $c(u) = 1$), ideally only one or two nodes. Sandberg suggested that two nodes u and v consider to swap IDs. As motivated above the swap is accepted with probability $min\{c(u)c(v), 1\}$. These swaps are the basis for the embedding currently done in Freenet: Random walks are performed to find a swapping partner, sending along the IDs of the swapping partners and their neighbors [5].

A node's behavior can be divided into three independent actions: *ASK*, *SWAP* and *TURN*.

- *ASK*: a node is asked for its current ID
- *SWAP*: a node receives a swapping request
- *TURN*: a node initiates a swap request

A *swapping request* is a message containing the ID id_I of the initiator I, the IDs of its neighbors $ID_N(I)$, and a time-to-live counter *TTL* detailing for how many steps the message should be forwarded. Initially *TTL* is set to some TTL_{RW}. The actions of a node u are described in listing 1. Note that in the beginning of every *TURN* a node asks for its neighbors' IDs, as does a partner in a swap. The function $swap$ chooses a random neighbor to execute *SWAP* with the given parameters. The number of iterations of the ID distribution algorithm is defined as the number of *TURN* actions that are executed in total.

```
ASK:   return id(u)
SWAP:  if TTL − 1 > 0
           return swap(id_I, ID_N(I), TTL − 1)
       elseif swap is accepted
           return id(u) and change id(u) to id_I
       else return FAILED
TURN:  id_S = swap(id_I, ID_N(I), TTL_RW)
       if id_S ≠ FAILED
           change id(u) to id_S
```

Listing 1. Honest node behavior in swapping

Figure 1 illustrates the individual steps of a swapping attempt. Considering adversarial behavior, swapping offers a large attack surface and a wide range of vulnerabilities. N_I, a malicious neighbor of the initiator I, can supply a wrong ID (1) when asked by I. I can initiate a swapping request with incorrect information (2), or the information can be changed by forwarding nodes F (3). Neighbors N_R of the swapping partner R can supply incorrect information (4), as can R (5) and again forwarding nodes F can modify the answer (6).

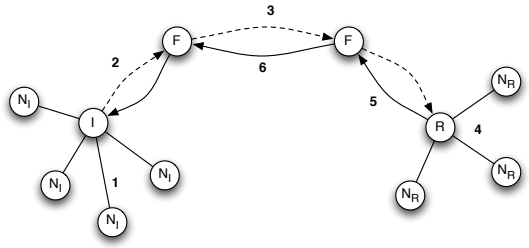

Fig. 1. One iteration in the swapping algorithm

Note that it does not make a difference if R sends back an ID or its neighborhood information to I. Given I's neighborhood information R can always choose its own ID so close to a neighbor of I that the swap will be accepted.

V. Local Markov Chain algorithm

The goal of our newly designed algorithm is to minimize the number of possibilities to attack the algorithm as well as the expected damage of an attack. The most problematic phase of swapping is *SWAP*. For one thing the neighborhood information of nodes is given to untrusted peers, providing an opportunity to gain information about the network's structure. The second and more important issue is that any node u receiving a swapping request can trick the initiator into accepting an arbitrary ID by pretending that the offered ID improves u's distance distribution extremely, so that the swap is accepted. For these reasons our embedding algorithm, Local Markov Chain (LMC), only uses *TURN* and *ASK* which are executed as shown in Listing 2. Again it is implicit that u asks for the IDs of its neighbors in the beginning of a *TURN*. It then computes $c(u)$ as defined in Section IV, where in the suggested next state u's current ID is replaced by id_R, an ID selected uniformly at random by u.

```
ASK:   return id(u)
TURN:  select id_R
       if id_R is accepted
           change id(u) to id_R
```

Listing 2. Honest node behavior in LMC

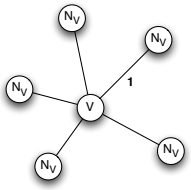

Fig. 2. One iteration of LMC

In contrast to Sandberg's algorithm, new IDs are not offered by peers, but chosen uniformly at random. The own ID is not given to any nodes but the direct neighbors. A node only has to rely on its direct neighbors when asking for their ID. This provides a point of attack, giving an adversary the possibility to supply an incorrect ID. However, the adversary has no information about the neighborhood of that neighbor

or the suggested ID, which complicates choosing an ID that might effectively undermine the quality of the embedding. An adversary receiving a swapping request in comparison has much more information.

VI. Attacks

In this Section we describe three types of attacks which pursue different goals and can be applied to both embedding algorithms. For each attack we describe how an attacker could implement it while still following the respective protocol of the targeted embedding algorithm. As described before, the protocols of both embeddings can be described by a node's behavior for *TURN* and *ASK*. In addition the swapping protocol also implements the *SWAP* messages for finding a potential swapping partner and possibly executing an ID exchange.

In the following we first introduce some additional notations allowing us to describe the attacker's behavior. Then, we describe three possible goals of an attacker, namely divergence, contraction, and non-optimal distance distribution and how they can be implemented by an attacker in both sorting algorithms. Divergence denotes the strategy to make other nodes regularly change their ID, thus keeping the algorithm from converging. Contraction means that an attacker tries to undermine the uniformity of the ID distribution, influencing a large number of nodes to accept an ID in a certain interval. The goal of the third attack - creating a non-optimal distance distribution - is to maximize the number of edges with a length not favored by the distribution in Kleinberg's model.

A. Notation

$id_\delta(id_X)$ denotes an ID chosen uniformly at random from $[id_X, id_X + \delta]$. $ID_{opp}(id_X, k) = \{id_1, id_2, \dots id_k : id_i = id_\delta(id_X + 0.5)$ for $1 \leq i \leq k\}$ denotes a set of IDs of size k on the opposite side of the identifier space of id_X. Let $ID = \{id_1, \dots id_{|ID|}, id_1\}$ be an ordered list of IDs with $id_i \leq id_{i+1}, 1 \leq i < |ID|$. Then $id_{far}(ID) = \max_{d(id_i, id_{i+1})} id_i + d(id_i, id_{i+1})/2$ is the ID at biggest minimal distance to all IDs in *ID*.

Furthermore, let A denote an attacker and TTL_R a number chosen uniformly at random from $\{1, 2, \dots TTL_{RW}\}$ where TTL_{RW} is the standard random walk length applied by non-malicious nodes.

B. Divergence

One goal of an attacker could be to avoid the convergence of the sorting algorithm into a stable state or, of course, to destabilize an already established stable state. Such a stable state cannot be reached as long as a large number of nodes are still changing their IDs. This could be achieved by an attacker by tricking as many nodes as possible to change their IDs regularly to a random ID.

An attacker therefore returns a random ID whenever it receives an *ASK* request, hoping that the neighbor changes its ID to adapt to this arbitrary ID, even if only with a small probability. When receiving a *SWAP* request, the attacker also returns a random ID, forcing the initiator to accept the arbitrary ID. Here the main weakness of swapping becomes obvious, because the attacker can actually dictate the initiator the new ID, while for *TURN* and *ASK* the attacker has to rely on (possibly very small) probabilities. During a *TURN*, the attacker selects a random ID. Afterwards it initiates a *SWAP* request with a random *TTL* and again proposes a random ID. In addition it computes a neighborhood at the furthest possible distance of the randomly chosen ID, which is supposed to cause the recipient to accept the swap request in order to improve the attacker's pretended position (cf. Listing 3).

```
ASK:   return id_R
SWAP:  return id_R
TURN:  select id_R
       swap(id_R, ID_opp(id_R, d(A)), TTL_R)
```

Listing 3. Attacker behavior to achieve divergence

In LMC, an attacker simply executes *ASK* and *TURN*.

C. Contraction

A different goal of an attacker is the contraction of the identifier space to a single point. This should compromise the uniformity of the ID distribution as well as routing performance since there is more or less no difference between long- and short-distance links any more. In order to achieve this, an attacker randomly selects one of its neighbors N_{sel} and attempts to convince others to choose an ID close to this neighbor's current ID. *ASK*, *TURN* and *SWAP* are now implemented analogously to the case of a divergence attack, returning an ID close to $id(N_{sel})$ instead of a random ID (cf. Listing 4).

```
ASK:   return id_δ(id_A(N_sel))
SWAP:  return id_δ(id_A(N_sel))
TURN:  select id_δ(id_A(N_sel))
       swap(id_δ(id_A(N_sel)), ID_opp(id_δ(id_A(N_sel)), d(A)), TTL_R)
```

Listing 4. Attacker behavior to achieve contraction

D. Non-optimal Distance Distribution

The third goal we are discussing in this paper is to achieve non-optimal distances between nodes and their neighbors by ruining the targeted distance distribution. To achieve this, an attacker could either propose a bad ID to a node, depending on its current neighborhood, or keep others from adapting their ID at all. In LMC a node has no information about another node's current neighborhood and therefore can only perform the second action.

Therefore, when an attacker receives an *ASK* request from a neighbor S, it returns an ID close to the requester's ID in order to keep the neighbor from changing its current ID. During each *TURN*, an attacker selects an ID far away from most neighbors in order to ruin the own distance distribution. When receiving a *SWAP* request from another node with initiator I and given neighborhood $ID_N(I)$, the attacker is able to supply an ID that leads to an unfavored edge length distribution (cf. Listing 5).

```
ASK:   return id_δ(id_A(S))
SWAP:  return id_far(ID_N(I))
TURN:  select id_far(ID_N(I))
```

Listing 5. Attacker behavior to achieve non-optimal distances

VII. Evaluation

We evaluated the embedding algorithms and the discussed attacks using GTNA, a framework for graph-theoretic network analysis [7]. The source code of GTNA as well as our implementation of the embedding algorithms, attacks, and routing algorithm are available online[1].

In this Section our simulation setup and results are presented, discussing three hypotheses regarding the performance of the Swapping and LMC embeddings.

A. Simulation Setup

As basis for the connectivity graph of trusted relationships between the participants, we use a snapshot of the friendship relations between members of "Studentenportal Ilmenau"[2] which we obtained in August 2010 directly from the service provider. After removing isolated nodes, the graph consists of 9223 nodes and 48875 undirected edges. The graph has a clustering coefficient of 0.38, a characteristic path length of 4.66, and a diameter of 12.

Simulations are structured as follows: First every node is assigned a random ID from the ID space $[0, 1)$. Then, similar to [3], $|V| * 6000$ iterations of the respective algorithm are performed, where the node executing *TURN* is chosen uniformly at random. In a second phase, a specified number of malicious nodes is chosen randomly from the set of 500 nodes around the median degree. Again $|V| * 6000$ iterations are executed using the already established embeddings from before, while the selected attackers now behave according to the different attack modes described in Section VI.

After the respective number of iterations, the quality measures from Section III-A are computed. In order to measure the routing performance, the routing algorithm is applied five times for every node using randomly selected IDs of different nodes in the network as target. For every routing attempt, the number of hops required to reach the destination node is stored. In case the routing attempt exceeds the given *TTL* of $log_2(|V|)^2 \approx 174$ it is marked as unsuccessful like in [3].

All attackers use the parameter $\delta = 1/1,000,000$ to compute $id_\delta(id_X)$ as described in Section VI. As proposed by Sandberg in [3], we chose the random walk's time-to-live counter as $TTL_{RW} = 6 \approx log_2(|V|)/2$.

The results presented in the following are the averages of 100 runs for every combination of embedding and attack.

B. Swapping meets the quality criteria of an embedding

Our first hypothesis is that Sandberg's swapping algorithm meets the quality criteria of an embedding described in III.

Since the algorithm does not change the randomly chosen identifiers but only reassigns them to different nodes, the uniform ID distribution is preserved as indicated by Figure 3, upper left.

[1]http://www.p2p.tu-darmstadt.de/research/gtna/

[2]http://spi.tu-ilmenau.de/

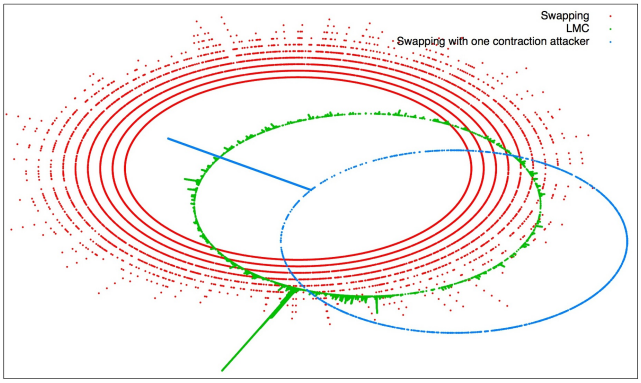

Fig. 3. Radial representation of the binned ID spaces of swapping, LMC, and swapping with one contraction attacker (from upper left to lower right). Each bin contains the identifiers in an interval of size 0.0005 and its height represents the respective number of contained IDs.

As indicated by Figure 4, upper left, the topological distances between nodes is only improved slightly when comparing the result of the swapping algorithm to a random ID assignment.

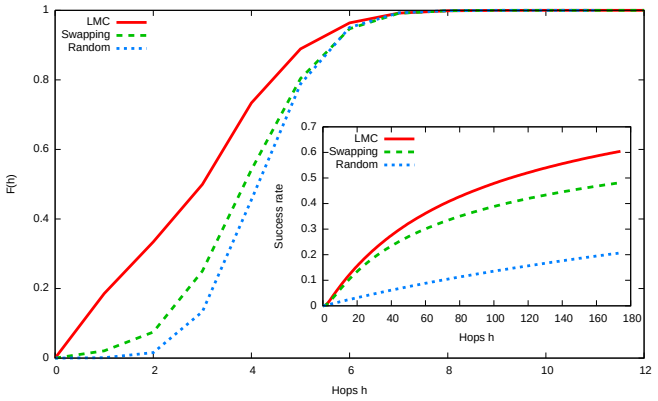

Fig. 4. Cumulative distribution of the number of hops to a node's successor (upper left) and cumulative distribution of the number of hops of successful routing attempts (lower right) for LMC, swapping, and a random ID assignment

Lastly, the routing performance of a network using a swapping embedding can be seen as adequate considering the constraints imposed by the setting and its distributed nature. While a network with randomly chosen IDs only achieves a routing success rate of 21%, the application of the swapping algorithm improves this rate to 48% as depicted by Figure 4, lower right. Obviously, this still low success rate comes at the price of a comparably long path length.

Our measurements lead us to the conclusion, that in the absence of attacks swapping outperforms a random embedding significantly with regard to the quality criteria described in Section III.

C. Swapping is vulnerable to attacks

Our second hypothesis is that the swapping algorithm is vulnerable to attacks leading to a huge degradation of the

embedding's quality. To test this hypothesis, we evaluated the resilience of the swapping algorithm by introducing one, ten, and 100 attackers to the system as described above. In each scenario, one of the three attack strategies described in Section VI were followed by the respective attackers.

One good example of the impact of an attacker is shown in Figure 3, lower right. It shows the IDs distributed in a system using the swapping embedding with one single attacker present. This attacker attempted to contract the identifier space which was clearly successful. While the IDs are uniformly distributed in a system without any attacker (cf. Figure 3, upper left), they center around one spot afterwards.

Table I lists the routing success rates of each scenario. With only one attacker in the system, the routing's success rate drops from 48% to 45% or even 41%, depending on the applied attack strategy. Increasing the number of attackers leads to average rates between 33% and 36% successful routing attempts for 100 attackers, a decline between 31.25% and 25% compared to the already low success rate without attacks. The decline is less drastic since the neighborhoods influenced by the attackers overlap considerably

These observations verify that swapping is not resilient.

TABLE I
ROUTING SUCCESS RATE FOR EMBEDDINGS AFTER ATTACKS

	LMC			**SW**		
Attackers	Contr.	Div.	Dist.	Contr.	Div.	Dist.
0	60%	60%	60%	48%	48%	48%
1	60%	60%	60%	41%	45%	45%
10	60%	60%	60%	41%	45%	43%
100	60%	60%	60%	36%	35%	33%

D. LMC is not prone to the discussed attacks

Our third hypothesis states that LMC, in contrast to swapping, is not prone to the three types of attacks described in Section VI.

Analogously to our resilience evaluation of the swapping algorithm, we created scenarios for LMC including one, ten, and 100 attackers behaving according to one of the three attacker models. The distance distribution, the distribution of hops to a node's successor, as well as the distribution of IDs in the identifier space are hardly altered by the presence and actions of attackers, independent of their number and behavior. Most importantly, the impact on the success rate of routing attempts performed on networks including any number of attackers is insignificant (cf. Table I).

Even though the embedding is not significantly affected by the presence of attackers, LMC in general results in a rather clustered ID space that does not fulfill the quality criteria of a uniform ID distribution as the corresponding example in Figure 3 indicates. As evidenced by Figure 4, upper left, the quality criteria of topological closeness to successors is far better met by LMC than by swapping: When applying LMC 33.46% of all successors are at most two hops away. Using swapping only 7.5% of successors are reachable over only one or two hops.

We conclude that LMC is not prone to the attacks discussed in this paper and does not suffer greatly from the actions of attackers. Except for the partition of the ID space, it performs better than swapping regarding the evaluated quality criteria.

VIII. CONCLUSION & OUTLOOK

In this paper, we have evaluated the resilience of Sandberg's swapping embedding against three different attacks - divergence, contraction, and non-optimal distance distribution. We introduced LMC, a network embedding for Darknets strictly based on the local knowledge of nodes. Our evaluation of both embeddings indicates that LMC achieves better performance considering different quality measures in the absence of adversaries at the cost of a non-uniform ID distribution. In face of one, ten, or 100 adversaries, the quality of the swapping embedding decreases drastically while LMC is not significantly influenced. In a scenario with 100 adversaries (approx. 1% of the network size) the success rate of routing attempts drops by more than 25% for swapping while LMC's performance is not influenced at all. Achieving only non-uniform ID distribution is LMC's sweetspot. Considering this property for swapping, however, a single malicious node using the contraction attack can effectively destroy the uniformity of the ID distribution entirely, concentrating most IDs around a selected spot. This indicates that LMC is to be preferred over swapping as an embedding algorithm for Darknets, since it offers a higher resilience against all considered attacks.

The results presented in this paper represent mere work in progress, and a more thorough analysis of LMC is needed. This includes the evaluation of additional quality metrics and a thorough analysis of the impact, individual and collusion attacks have on LMC. Furthermore, we will analyse the impact of attacks on other proposed embedding approaches, like, e.g., the one proposed in [8]. Analyzing the different embeddings using additional datasets like PGP's Web of Trust will give additional insight on the dependence between network model, embedding and its attack resilience.

REFERENCES

[1] I. Clarke *et al.*, "Private communication through a network of trusted connections: The dark freenet." [Online]. Available: http://freenetproject.org/papers/freenet-0.7.5-paper.pdf

[2] I. Stoica *et al.*, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[3] O. Sandberg, "Distributed routing in small-world networks," in *Proceedings of the Eighth Workshop on Algorithm Engineering and Experiments (ALENEX06)*, 2006, pp. 144–155.

[4] J. M. Kleinberg, "The small-world phenomenon: an algorithmic perspective," in *STOC '00 Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 2000, pp. 163–170.

[5] N. S. Evans *et al.*, "Routing in the dark: Pitch black," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, 2007, pp. 305–314.

[6] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[7] B. Schiller *et al.*, "GTNA: a Framework for the graph-theoretic Network Analysis," in *Proceedings of the 2010 Spring Simulation Multiconference, SpringSim 2010*, 2010, pp. 111–119.

[8] M. Dell'Amico, "Mapping small worlds," in *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*, 2007, pp. 219–228.