Decapitation of Networks With and Without Weights and Direction: The Economics of Iterated Attack and Defense

Josep Domingo-Ferrer and Úrsula González-Nicolás

Universitat Rovira i Virgili
UNESCO Chair in Data Privacy, Dept. of Computer Engineering and Mathematics
Av. Països Catalans 26, E-43007 Tarragona, Catalonia
E-mail {josep.domingo,ursula.gonzaleznicolas}@urv.cat

Abstract

Vulnerability of networks against one-shot decapitation attacks has been addressed several times in the literature. A first study on how a network can best defend itself by replenishing nodes and rewiring edges after decapitation was made by Nagaraja and Anderson. We extend the work of those authors on iterated attack and defense by: i) considering weighted and directed networks; ii) taking into account the economic aspects of the attack and defense strategies, namely the cost of node destruction/replenishment and a subtler cost such as the average path length increase. Extensive empirical work is reported which, among other things, shows the importance of obfuscating the network topology.

Key words: Network iterated attack and defense; iterated decapitation attacks; security; privacy; private social networks; peer-to-peer networks.

1 Introduction

Many empirically observed networks, including the world wide web, some social networks (with or without private relationships) and a good deal of peer-to-peer networks, can be modeled as scale-free networks. A scale-free network is a network whose degree distribution follows a power law, at least asymptotically. In other words, the fraction $P(m)$ of nodes in the network having m connections to other nodes (that is, degree m) is, for large values of m, $P(m) \sim m^{-\gamma}$, where typically $2 < \gamma < 3$.

A network may be attacked by organizations interested in dismantling it: for example, a peer-to-peer network for movie exchange may be attacked by movie companies; a private social network may be attacked by a dictatorial regime (there have been recent examples of this situation in Asia), etc. Albert et *al.* [2] showed that scale-free networks are more robust against random attacks (*i.e.* removal of randomly chosen nodes) than against targeted attacks. The reason is that scale-free networks get much of their connectivity from a minority of nodes having a high degree: it is hard to hit a significant number of those hub nodes in a random attack, but if an attack is targeted at removing one or more hub nodes, then connectivity decreases dramatically. Holme *et al.* [9] considered attacks on edges rather than nodes: the attacker removes edges connecting high-degree nodes; those authors also suggested using betweenness centrality as an alternative to degree for attack targeting (a node's betweenness centrality can be viewed as the number of paths along which it lies). Zhao *et al.* [13] studied under which circumstances a scale-free network suffers from cascading breakdown caused by the successive failure of hub nodes.

Anderson et *al.* [11,3] introduced the dynamic case in which at each round an attacker removes (or openly compromises) a certain number of nodes, but the defenders can recruit other nodes to replace the lost ones. The smaller the largest connected component after an attack-defense round, the more successful is the attack.

1.1 Terminology

In this paper, we use the term *network* in its engineering sense, that is, a set of nodes connected by a set of links. A network can be mathematically modeled as a graph, by mapping the nodes to vertices and the links to edges. In what follows, we will use network and graph indistinctly [1]. Also, we will use link and edge indistinctly.

The largest connected component (LCC) of a graph is the largest subgraph such for any two of its nodes, say i and j, node j can be reached from node i using a path formed by edges internal to the subgraph.

We will distinguish between *unweighted networks* and *weighted networks*, that is, between unweighted graphs and weighted graphs, where a weighted graph is a graph together with a function assigning a real number to each edge. Also, we will distinguish undirected networks and directed networks. An *undirected network* is a plain graph, that is, a set of nodes and a set of unordered pairs of nodes called edges. A *directed network* is a directed graph or digraph, that

[1] We will ignore the graph-theoretic convention of restricting the term network to denote weighted digraphs.

is, a set of nodes and a set of *ordered* pairs of nodes called arcs, directed edges or simply edges.

In graph theory, the centrality of a node is the relative importance of a node within the network. In engineering, centrality can be understood as the criticality of a the node, that is, how critical its operation is for the operation of the overall network. Centrality can be measured as:

- *Degree.* The centrality-criticality of a node is measured as the number of edges incident to a node. In directed networks, one distinguishes the in-degree (number of edges directed to the node) and out-degree (the number of edges the node directs to other nodes).
- *Path centrality.* The centrality-criticality is measured in terms of the paths to which the node belongs. For example, the betweenness centrality of a node is the number of shortest paths between other nodes along which the node lies. Reliability, which will be recalled and used below for weighted and directed graphs, is another form of path centrality.

1.2 Contribution and plan of this paper

Two attacks were examined in [11,3]:

- *Removal of high-degree nodes.* The defense strategies considered are: i) random replacement, that is, addition of random links to the neighbors of each removed node; ii) ring replacement, whereby the neighbors of a removed node add new links so as to form a ring connecting each neighbor to another neighbor; iii) clique replacement, whereby the neighbors of each removed node connect to each other forming a complete subgraph (*i.e.* a clique) or, alternatively, an entire new clique is added to replace the removed node, and the clique nodes inherit the links of the removed node. Simulations in [11] show that random replacement is ineffective: after three rounds the size of the LCC has fallen by a half. Ring replacement is only able to postpone network collapse from three to a dozen rounds. Clique replacement is the only defense strategy which manages to maintain the network robustly connected.
- *Path centrality attack.* The nodes having the highest path centrality in the network (according to some path centrality metric) are removed at each round. Clique replacement is less effective against path centrality attacks. Combining cliques with delegation is most effective, where *delegation* means that a node that is becoming too well-connected selects one of its neighbors as a "deputy" and transfers some of its links to it.

The above-mentioned seminal papers were mostly based on simulations and assumed no constraints on the budget of attackers and defenders. Also, the

networks considered were unweighted and undirected. We start from that work and extend it to consider weighted and directed networks. There are several reasons to consider weights and directions:

- Delegation strategies and/or the computation of node centralities may depend on the weight of the links. *E.g.* if weights are trusts or distances, they clearly matter when choosing delegates and/or computing centralities. Also, in the case of weights representing trusts, the direction of the links plays a role: two nodes may trust each other to a different extent.
- The physical distance or other kinds of distance between nodes may also be relevant to assess the impact of an attack on the LCC of the network: a connected component with large inter-node distances is worse than a connected component of the same size but with smaller inter-node distances.

Another extension regards the economic aspects of the attack and defense strategies: cost of node destruction/replenishment and a subtler cost such as the average path length increase.

The practical significance of the attack and defense strategies explored in this paper must be stressed here. Think of the networks under attack as power supply or communication networks. Attacks could be carried out by physical destruction of nodes or links (*e.g.* through selective bombing by an enemy army or a terrorist gang) or by functional disablement (*e.g.* through a denial-of-service attack by hackers, *e.g.* [1]). The best attack strategies are those targeted at the most critical nodes. Defense strategies correspond to the most effective actions to repair the attacked parts of the network: replacing nodes, rewiring them, etc. Very often, it is not possible to immediately repair the network after each attack round, because doing so is either too risky (*e.g.* in a battlefield) or it is economically prohibitive (sending a team to repair a set of nodes is less costly than sending the team to repair a single node). Therefore, the attack and defense dynamics assumed in this paper is as follows: an attacker performs a batch of attack rounds and only after this attack batch is complete is the defender allowed to initiate a single defense round whose reach is limited (not every destroyed node can be restored); then a second batch of attack rounds takes place, after which a second defense round is performed, and so on. In this way, individual defense rounds are interleaved with batches of attack rounds. We explore several defense actions and attack batch sizes (*i.e.* the attack batch size is the number of attack rounds allowed before a defense round is initiated).

Section 2 presents an economic model for attack and defense, including cost functions. Section 3 is about measuring the success of attacks and defenses. Section 4 presents extensive empirical work. Conclusions are summarized in Section 5.

A partial version of this article, without the extension to directed networks and without any simulation results, was accepted as a workshop paper [6].

2 An economic model for attack and defense

We assume an external attacker A with a maximum budget B_A that the attacker can spend on an attack round. In fact, B_A can vary from one round to the next, but, since our analysis will be round-based, we only need to focus on the budget for one round. Similarly, assume that the defender's budget for a defense round is B_D. We will allow the defender to concentrate his resources and undertake a defense round only after a number of attack rounds have taken place (rather than immediately acting after each attack round). Following [11], we assume a defense strategy consisting of a compound of delegation and cliques. However, we use those strategies in a different manner:

- Delegation is used before attacks start, but it is also checked at regular intervals whether delegation is needed again after attacks have started.
- Two delegation algorithms are considered:

(1) The first one focuses on node degree and reduces the degree of nodes with high degree by transferring some edges to their neighbors;
(2) The second one focuses on reducing path centralities by replacing nodes having high path centrality by cliques and transferring the edges of each replaced node uniformly to the clique nodes.

- Node replenishment after destruction is conducted in the same way as delegation with cliques: a destroyed node is replaced by a clique.

2.1 Attacks and their cost

We define the cost $C_A(k)$ of mounting an attack to destroy/remove a number k of nodes as

$$C_A(k) = C_{FA} + k * C_X$$

where C_{FA} is the fixed cost incurred to locate the target nodes and C_X is the cost of destroying one node.

2.1.1 Locating target nodes

A rational attacker identifies the target nodes as those with the highest criticality, that is, those whose destruction is most harmful. Several measures for the criticality of a node can be selected:

- In [9], criticality measures considered were node degree (the number of edges incident to a node) and node path centrality (specifically, those authors used Freeman's betweenness centrality [7]). In [11], the same measures were taken and Brandes's algorithm [4] was used to compute the betweenness centrality of a node.
- Other path centrality measures can be used to identify the most critical target nodes in the case of weighted graphs, *e.g.* the reliability index R_1 defined in [12], which we next recall. Consider a strongly connected undirected weighted graph $G = (V, E, w)$ where $w : E \to (0, 1]$ is an edge weight function. Given a path p in G, the reliability $w(p)$ of the path is defined as the product of the weights of the edges along the path. For two nodes $i, j \in V$, denote by F_{ij} the weight of the most reliable path between i and j. If the nodes in V are labeled from 1 to n, the reliability R_1 [12] of a node i is defined as

$$R_1(i) := \sum_{j=1}^{n} F_{ij} \tag{1}$$

In case the attacker has only partial information on the network topology (as it is likely to happen if the attacked network is a social network with private relationships, like the ones described in [5]), the most critical nodes in the subgraphs known by the attacker would be identified as target nodes.

2.1.2 Destroying critical nodes

The cost C_X of destroying one node can be spent on physical destruction, node jamming, node flooding to cause denial of service, etc.

2.2 Defenses and their cost

Define the cost $C_D(h, k, q)$ of defending the network as

$$C_D(h, k, q) = C_{FD} + C_{del}(h[, q]) + k * C_{cli}(q) \tag{2}$$

where C_{FD} is the fixed cost of starting a defense round (*e.g.* of sending a repair team), $C_{del}(h[, q])$ is the cost of implementing delegation for nodes with criticality δ higher than h (parameter q is the clique size in case delegation is based on cliques); C_{cli} is the cost of replacing each of k destroyed nodes by a clique consisting of q new nodes.

2.2.1 Delegation strategies and their cost

Delegation can be run before attacks start and should be re-run each time new edges are added to the network due to the defense strategy of clique formation.

We present two delegation algorithms. Denote by N_i the set of neighbors of node i. In the first delegation algorithm, node criticality is measured as node degree. A node i with degree $\delta(i) > h$ attempts to transfer some of its edges to its neighbors.

Algorithm 1 (Node degree delegation(h))

For $i = 1$ *to* n *do*
While $\delta(i) > h$ *and* $\min_{j \in N_i} \delta(j) < h$ *do*
(1) Node i picks its most trusted neighbor i' (or the neighbor i' with lowest degree if trust between nodes is not defined) among those with $\delta(i') < h$.
(2) Node i selects $\min(h - \delta(i'), \delta(i) - h)$ *edges linking i to its neighbors with highest degree (excluding i' from those neighbors);*
*(3) The selected edges are tranferred to i': one edge end is changed from i to i' and the edge weights are updated suitably (*e.g. *this is clearly needed if weights express physical distance, trust, etc.).*

The rationale of Algorithm 1 is that links to highly connected nodes are delegated first by node i, so that the neighbors that remain have low degrees and can absorb further delegation from node i.

The cost of node degree delegation is basically caused by edge transfer and can be expressed as

$$C_{del}(h) = \sum_{i:1 \le i \le n, \delta(i) > h} \sum_{j \in \Delta(i)} c_j$$

where $\Delta(i)$ is the set of edges delegated by node i and c_j is the cost of delegating the j-th edge in $\Delta(i)$. The value of c_j depends on a number of factors: i) the (lack of) trust and/or the difficulty of communication between i and the node i' to whom i delegates that edge (*e.g.* think of the difficulties of delegation within a network of terrorists); ii) the difference of weight/trust/difficulty of communication between i and the node i'' at the other end of the edge, and between i' and i''; etc.

In the second delegation algorithm, node criticality is measured using path centrality. We adopt the suggestion of [11] of reducing the path centrality of nodes through clique replacement: a node i with path centrality $\delta(i) > h$ is replaced by a clique of size q.

Algorithm 2 (Path centrality delegation(h, q))

While a node i with $\delta(i) > h$ exists do
(1) Create a clique consisting of q new nodes.
(2) For each node $j \in N_i$ do:
(a) Uniformly randomly sample one of the q nodes in the clique and call it i'.
(b) Transfer to i' the edge from i to j: one edge end is changed from i to i', and the edge weight is changed to adapt to the new end.
(3) Eliminate node i.

The cost of path centrality delegation is composed of both the cost of edge transfer and the cost of the new nodes forming the clique, that is,

$$C_{del}(h, q) = \sum_{i:1 \leq i \leq n, \delta(i) > h} (q * C_N + \sum_{j \in \Delta(i)} c_j)$$

where C_N is the cost of one new node, $\Delta(i)$ is the set of edges delegated by node i to the clique nodes (the clique nodes are sampled with replacement to decide which node inherits each edge delegated by i) and c_j is the cost of delegating the j-th edge in $\Delta(i)$. As discussed above, c_j depends on the node i' inheriting the edge and on the node i'' at the other end of the edge.

2.3 The cost of clique replenishment

A destroyed node is replaced by a clique of q nodes. The algorithm for clique replacement is a simplified version of the algorithm for path centrality delegation:

Algorithm 3 (Clique replacement(i,q))

(1) Let the N_i be the set of surviving neighbors of node i after destruction of node i (and possibly destruction of other nodes).
(2) Create a clique consisting of q new nodes.
(3) For each node $j \in N_i$ do:
(a) Uniformly randomly sample one of the q nodes in the clique and call it i'.
(b) Transfer to i' the edge from i to j: one edge end is changed from i to i', and the edge weight is changed to adapt to the new end.

The cost of replenishing one destroyed node, say node i, with a clique of size q is

$$C_{cli}(q) = q * C_N + \sum_{j \in \Delta(i)} c_j$$

where C_N is the cost of one new node and $\Delta(i)$ is the set of edges incident to the destroyed node i. The edges in $\Delta(i)$ are reassigned to the nodes of the clique built to replace that node (the clique nodes are sampled with replacement to decide which node inherits each of the edges formerly incident to i); c_j is the cost of reassigning the j-th edge in $\Delta(i)$ and it depends on the node i' in the clique inheriting that edge and on the other end i'' of the edge, as discussed above for delegation.

Note 1 If cliques with large q are inserted by the above algorithm to replace destroyed nodes, it can be argued that the nodes in the added cliques have large degree (at least q), which makes them a likely target for successive attacks in case node criticality is measured as node degree. While this is true, it is also true that an attack to a clique node is among the most harmless ones: indeed, destroying one clique node does little damage to connectivity, because most edges of a clique node link it with the rest of clique nodes, themselves all directly connected to each other.

2.4 Extension to directed graphs

The above attack and defense algorithms can be easily adapted to directed graphs (digraphs), with the following considerations:

- When evaluating the criticality of a node in a digraph via node degree, one may use the in-degree or the out-degree. The choice will depend on the specific use of the network represented by the digraph.
- When using path centrality to assess node criticality in a digraph, Freeman's betweenness centrality naturally extends to directed paths. As to path centrality measures for weighted digraphs, in [12] two different versions of the measure in Expression (1) are actually presented, namely the out-reliability $R_1^+(i)$

  $$R_1^+(i) := \sum_{j=1}^{n} F_{\vec{ij}} \tag{3}$$

  and the in-reliability $R_1^-(i)$

  $$R_1^-(i) := \sum_{j=1}^{n} F_{\vec{ji}} \tag{4}$$

  The out-reliability of a node is the capacity of the node to transmit data in a reliable form (*i.e.* using high-weight paths) to all other nodes. The in-reliability of a node is the capacity of the node to receive data in a reliable

form (using high-weight paths) from all other nodes. To what extent the out-reliability or the in-reliability characterize the criticality of a node in a digraph depends on the specific use of the network represented by the digraph. In the empirical work reported in Section 4 below, we have used the in-reliability.

- An edge transferred due to delegation or clique replacement should preserve its direction: if the edge origin is unchanged, the transferred edge should depart from that same origin; if the edge destination is unchanged, the transferred edge should arrive to that same destination.
- A clique of size q in a digraph should have two directed edges between each pair of nodes.
- When transferring an edge, the new weight of the transferred edge should be computed by the origin of the edge. *E.g.* if weights represent trusts, clearly it is the origin of the edge which must decide how much it trusts the destination of the edge.

3 Measuring the success of attacks and defenses

In the network decapitation literature, a common metric for the success of an attack is the size of the LCC which remains in the graph after the attack [2]. The rationale is that a reduction in the network connectivity makes communication impossible between some pairs of nodes.

We claim that the variation of the average path length (APL) before and after the attack is a measure which should also be considered, especially in the case of weighted networks. Indeed, in case the LCC experiences little or no variation as a result of an attack but there is a substantial increase in the APL, this should be credited as an attacker's success. The rationale here is that an APL increase means that communication is still possible, but has become more difficult. By way of example, consider a city spanning a river with several bridges; if a bombing destroys all bridges except one, the LCC of the street graph remains the entire city, but urban communication has clearly been impaired by the attack: people need longer trips to move between places on different sides of the river.

Analogously, the defender is successful if she manages to preserve the size of the LCC and the APL within the LCC.

Attack strategy A is said to be better than attack strategy B if, with the same budget, strategy A yields better attack success metrics than strategy B. That is: i) either A yields a smaller LCC than B; ii) or A and B yield an LCC of the same size but A's LCC has a longer APL than B's.

Similarly, defense strategy A is said to be better than defense strategy B if, with the same budget, strategy A yields better defense success metrics than strategy B. That is: i) either A preserves a bigger LCC than B; ii) or A and B yield an LCC of the same size but A's LCC offers a shorter APL than B's.

4 Empirical work

In this section we report on experimental results. We first specify the experimental setting and then discuss the results. Throughout the simulations, it became apparent that, *ceteris paribus*, attacks using path centrality as a criticality measure are always more effective than those using node degree, in that the former cause a greater reduction of the size of the LCC. This is consistent with what was reported in [3] for unweighted undirected networks. Therefore, and for the sake of brevity, we will only discuss path centrality attacks.

4.1 Initial networks

We used synthetic networks and one large real network as initial networks for our experiments.

The SFNG Matlab function described in [8] was used to synthesize undirected scale-free networks. Different types of *initial* networks were generated:

- An unweighted and undirected scale-free network with 400 nodes was created by directly using the Matlab function.
- An unweighted and directed scale-free network with 400 nodes was created using the Matlab function twice, thereby obtaining two undirected networks with 400 nodes each, which were subsequently merged into a directed network. To do the merging, we took as upper triangular adjacency matrix of the directed network the upper triangular adjacency matrix of the first undirected network and, as lower triangular adjacency matrix, the lower triangular adjacency matrix of the second undirected network. This procedure ensured that the directed network was scale-free with respect to in-degrees and out-degrees of nodes.
- Weighted versions of the previous networks were generated by assigning to each edge a random weight in the interval $(0, 1)$. The weight of a path was computed as the product of its edge weights, that is, weights were multiplied as if they were reliabilities. The closer a path weight to 1, the better was the path. When computing the APL, paths with weight close to 1 were considered "short" and those with weight close to 0 were considered "long" (*i.e.* multiplicative $(0, 1)$ weights were converted to additive weights

by taking negative logarithms, see Section 4.2 below).

Furthermore, we used one large real network, a snapshot of the structure of the Internet at the level of autonomous systems taken on July 22, 2006 [10]. This unweighted and undirected network is formed by 22963 nodes.

4.2 Path centrality computation

For unweighted networks, we used Freeman's betweenness centrality [7]. This path centrality measure can be applied to both directed and undirected graphs.

For weighted networks, we used as path centrality the reliability measures of [12]: the reliability R_1 given by Expression (1) for weighted undirected networks, and the in-reliability R_1^- given by Expression (4) for weighted directed networks. The above reliability expressions use values F_{ij} which were computed as follows:

(1) If P_{ij} is the set of different paths between two nodes i and j and, given $p \in P_{ij}$, W_p is the set of edge weights in path p, the expression for F_{ij} is

$$F_{ij} = \max_{p \in P_{ij}} \prod_{w \in W_p} w$$

(2) The above maximum can be rewritten as the following minimum

$$\frac{1}{F_{ij}} = \min_{p \in P_{ij}} \frac{1}{\prod_{w \in W_p} w} = \min_{p \in P_{ij}} \sum_{w \in W_p} \log(1/w) \tag{5}$$

(3) Expression (5) is a shortest path problem with non-negative weights (since $w \in (0, 1)$, we have that $\log(1/w)$ is non-negative). Thus, we computed the values F_{ij} by applying Dijkstra's algorithm to the original network after replacing edge weights w with $\log(1/w)$.

4.3 Attack-defense dynamics

We considered attackers with several levels of partial knowledge on the attacked network: 100%, 80%, 60%, 40% and 20%. For example, an attacker with 40% knowledge is assumed to know a subnetwork formed by a (randomly drawn) subset of 40% of nodes in the network and the edges *internal* to that subset. Criticality computations (degree or path centrality) in view of determining the target node to be destroyed in the next attack were performed by the attacker on the subnetwork she knew. All subsequent attacks and defenses took place in that subnetwork.

For each initial network (unweighted undirected, unweighted directed, weighted undirected, weighted directed, real Internet network), for each criticality measure (node degree, path centrality) and for each level of attacker knowledge (100%, 80%, 60%, 40%, 20%) we conducted two simulations: one with no defense and another with interleaved defense rounds. For the larger real Internet network, we conducted two additional, longer simulations, one with no defense and the other with interleaved defense rounds. Thus, 102 different simulations were conducted altogether.

Each of the first 100 simulations consisted of 30 attack rounds; the two additional simulations for the Internet network consisted of 1722 attack rounds. In each attack round, the attacker chose a single node, the one with the highest criticality, and destroyed it. In those simulations including interleaved defense rounds, after each batch of q attack rounds, the defender was allowed to perform one defense round in which the most critical node destroyed in the last q rounds was replaced by a clique of q nodes. Thus, the defense round restored the total number of nodes of the network. In the 100 first simulations, we took $q = 5$ for the reasons given in the next section; in the two additional simulations for the Internet network, we took $q = 287$ for the reasons explained further below. For weighted networks, when an edge starting at a destroyed node was transferred to a new node (*e.g.* one of the new clique nodes), a new random weight was generated for that edge; in the case of weighted directed networks, a new random edge weight was computed only if the edge origin had changed.

In simulations with defense, before starting the attack rounds, the delegation procedure corresponding to the criticality measure in use was run (see Section 2.2.1 above); this might increase a bit the total number of initial nodes. After each defense round following a batch of attack rounds, it was checked whether delegation was again needed in the reconstructed network, but in our experiments this was never necessary.

4.4 Parameter choice for attack and defense

There are two relevant parameters that determine the defense strategy, namely the threshold criticality h and the clique size q:

- Parameter h was set in a network-dependent way: given a network, we took h such that only 5% of nodes had criticality $\delta > h$.
- The choice clique size q was oriented by the following experiments. As said above, in order to preserve the number of network nodes, when using q as clique size, q path centrality attack rounds destroying one node each were allowed before a defense round replaced the most critical destroyed node

with a clique of size q. Figure 1 shows the LCC size after each defense round for different values of q in an unweighted undirected network initially having 400 nodes, where the attacker had a 50% network knowledge. We took an unweighted undirected network because it proved to be the most vulnerable network type among the four types we simulated (see results without defense further below). It can be observed that, the greater the value of q (that is, the longer one waits to start a defense round), the smaller the LCC size becomes. For $q > 5$, the LCC size decreases very sharply in the initial attack-defense iterations and stabilizes thereafter at very low levels (well below 200 nodes). For $q \leq 5$, the initial decrease is much milder, and stabilization of the LCC size occurs at considerably higher levels (200-300 nodes). Since the results for $q = 5$ are not much worse than for $q = 2$ and starting a defense round for each five destroyed nodes is much more affordable than doing it for each two destroyed nodes (see discussion in Section 1.2), we took $q = 5$ as a reasonable trade-off value in the rest of experiments reported in this article.

Note 2 For larger networks containing substantially more than 400 nodes, larger values of q could be affordable: while Figure 1 shows that *e.g.* $q = 15$ is too damaging for a 400-node network, getting 15 nodes destroyed between defense rounds would have less relative importance in a 22963-node network like the one in Section 4.5.5 below. However, taking a value of q tailored for small networks is a safe choice also for large networks: even for a large network, it is much better to defend after the loss of, say, $q = 5$ nodes than after the loss of, say, $q = 15$ nodes. Taking a larger q in a larger network would only be justified by cost reasons, if the fixed cost C_{FD} in Equation (2) of starting a defense round were so high that defense rounds should be made less frequent. See also results and discussion in Section 4.5.5.

4.5 Discussion of results

The metrics considered to evaluate the success of attacks are: i) LCC size; ii) the relative APL, that is, the APL within LCC divided by the LCC size. The greater the LCC size and the shorter the relative APL, the worse for the attacker and the better for the defender.

In the following sections, we will present the evolution of the above attack success metrics for various network types. Results with no defense rounds and with one defense round after each $q = 5$ attack rounds will be given.

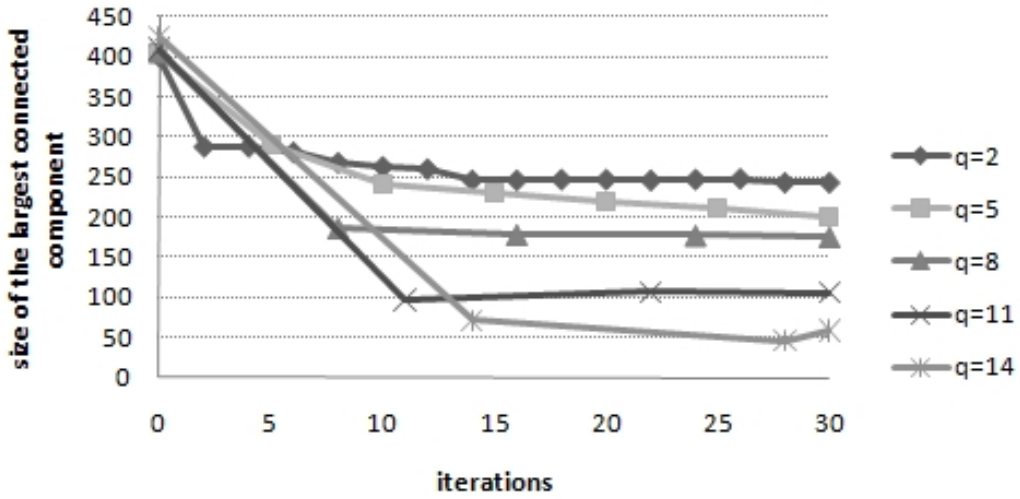


Fig. 1. For several values of q, evolution of the LCC size in an unweighted and undirected network with 400 initial nodes when one defense round is performed after each batch of q successive attack rounds by an attacker having a 50% network knowledge

4.5.1 Unweighted undirected network

Figure 2 shows the effectiveness of the proposed defense strategy to preserve the LCC size and the relative APL in a simulated unweighted undirected network of 400 nodes. Just compare the left-hand side graphics with the right-hand side ones.

Without defense, after 5 attack rounds, the top left graphic shows that the LCC size falls by more than 40% for an attacker with 20% network knowledge (best case). However, for that same attacker, the top right graphic shows that the first defense round brings the LCC size back to 381, which amounts to just a 5% fall; furthermore, with interleaved defense rounds, no further decreases of LCC size are suffered under a 20% attacker.

For attackers with more network knowledge, the situation is less brilliant, but still there is a clear advantage in using defense rounds. With an attacker with 40% or more network knowledge, the LCC size after attack consistently decreases, with defense rounds after every fifth attack round providing an increase which cannot entirely compensate the size reduction incurred over the last five attack rounds (top right graphic). However, the evolution of the LCC size with no defense is much worse (top left graphic).

Similar remarks apply to the relative APL. When comparing the two bottom graphics, note that the ordinate range in the right bottom graphic is one third of the ordinate range in the left bottom graphic. The relative APL when using defense rounds is always much lower than without defense.

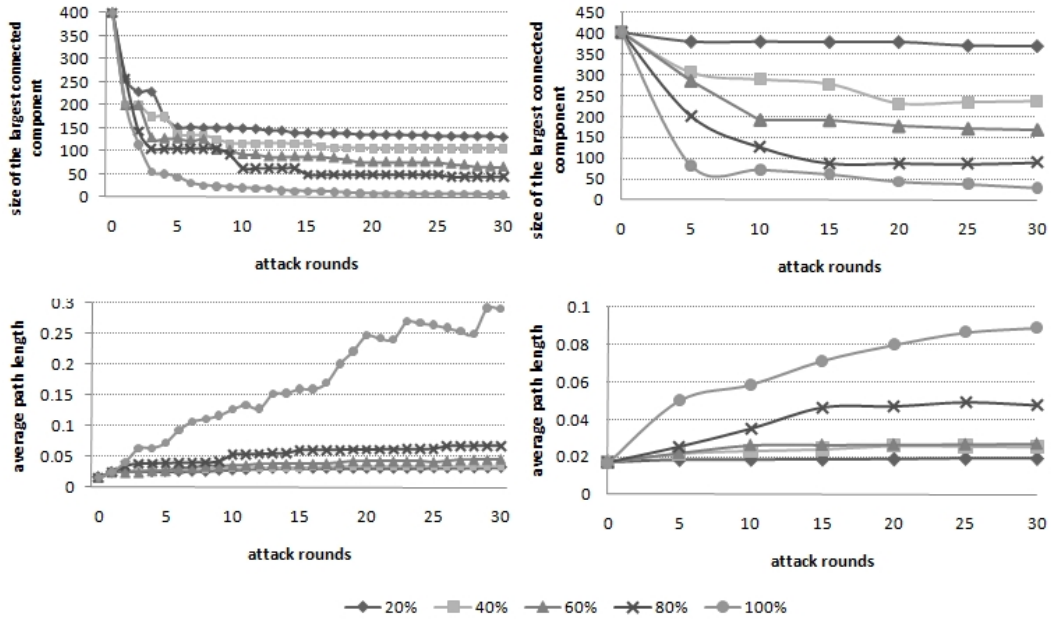


Fig. 2. Success metrics of attacks on an unweighted undirected network with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 5$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 5$).

4.5.2 Unweighted directed network

The results for a network of this type are shown in Figure 3. Comparison between left-hand side and right-hand side graphics in the figure reflects the effectiveness of the defense strategy (note that the ordinate ranges in those pairs of graphics differ). The falls in the LCC size reflected by the top left graphic (no defense) are substantially diminished in the top right graphic (defense). In fact, for an attacker with 20% network knowledge, there is no LCC size fall at all when using defenses: after the defense round following the 30th attack, the LCC size is still the initial one.

Comparison of the right-hand side graphics in Figure 3 with the corresponding ones in Figure 2 shows that the defense is more successful for the unweighted directed network. The reason of this more robust behavior is that we build a directed network by assigning opposite directions to two undirected networks and merging them, so that our directed network has higher connectivity and hence more resistance.

Attackers with 40%, 60% and even 80% network knowledge do less harm than the harm an attacker with 20% knowledge did on an unweigthed undirected network. As noted above, this is due to the superior connectivity of directed networks. *E.g.*, consider an attacker with 80% knowledge. The LCC size after the first $q = 5$ attack rounds falls from 404 (the initial value is above 400

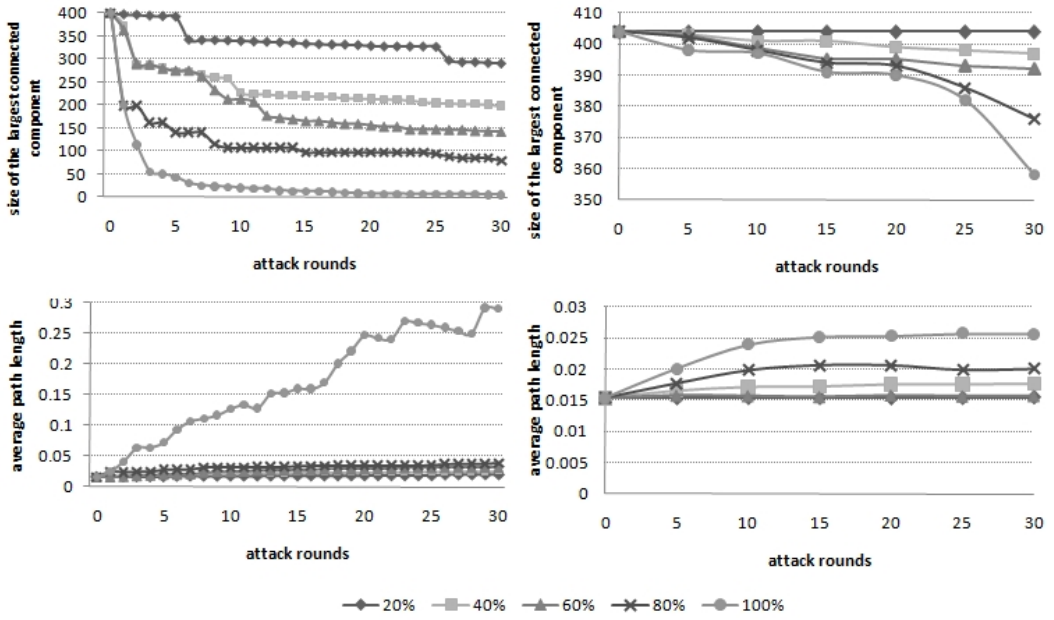


Fig. 3. Success metrics of attacks on unweighted directed networks with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 5$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 5$).

due to initial delegation) to 395. The first defense round after the fifth attack round restores this size to 402. Subsequent attack rounds cause this size to decrease to 393, 389, 388, 380 and 366, with corresponding defense rounds increasing it to 398, 394, 393, 386 and 376, respectively.

An attacker with 100% knowledge certainly causes more harm, but still less than the harm caused by an attacker with 40% knowledge on an unweighted undirected network. It is interesting to note that, while the relative APL stays nearly constant with or without defense under attackers with up to 80% network knowledge, there is a clear difference for an attacker with 100% knowledge: without defense, the APL length grows all the time, while with defense it stabilizes around 0.025.

4.5.3 Weighted undirected network

The path centrality measure used changes for weighted networks and the path length is computed by taking weights into account. Since we use multiplicative weights for links in our simulated network (see Section 4.1), the closer the product of weights along a path to 1, the "shorter" (*i.e.* the better) is the path. Yet, results, depicted in Figure 4, are similar to those obtained for unweighted undirected networks (Section 4.5.1).

The effectiveness of the defense strategy can be justified by comparing the left-

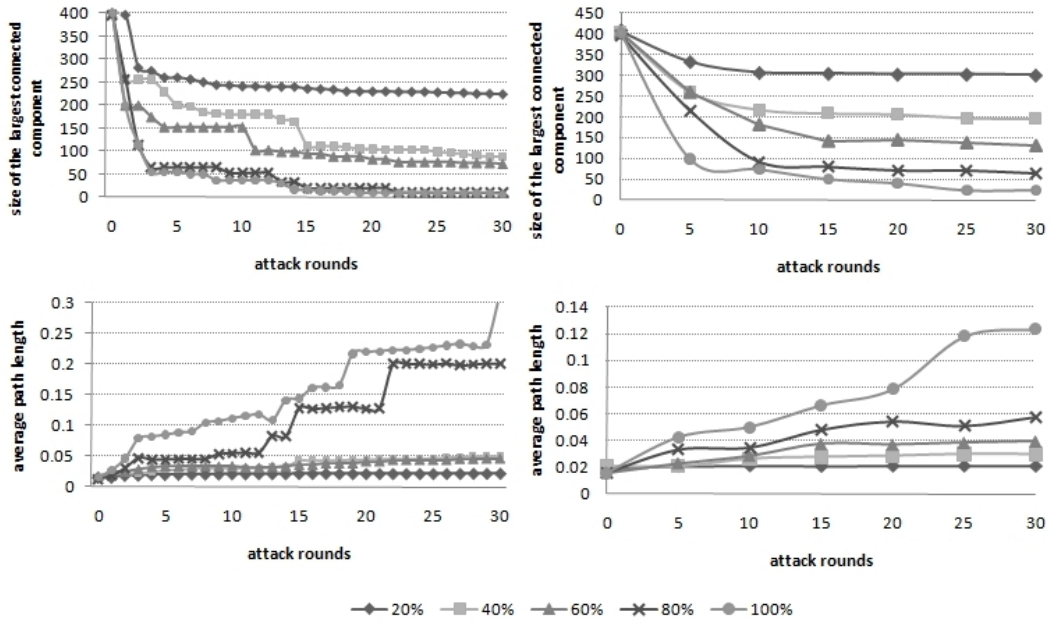


Fig. 4. Success metrics of attacks on weighted undirected networks with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 5$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 5$).

hand side graphics and the right-hand side graphics of Figure 4. The argument is parallel to the one given in Section 4.5.1 about Figure 2.

In the top right graphic of Figure 4 we can see that, under an attacker with knowledge of up to 20% of the network, the LCC size after the first $q = 5$ attack rounds falls from 403 to 371. The first defense round after the fifth attack round restores this size to 389. Subsequent attack rounds cause this size to decrease to 381, 360, 359, 358 and 356, respectively, with corresponding defense rounds increasing those sizes by about five nodes. The relative APL stays constant with the successive rounds (bottom right graphic of Figure 4) . So, overall, it can be said that an attacker with only 20% network knowledge is fairly well withstood.

For attackers with higher knowledge (from 40% onwards), the LCC size stabilizes at lower levels or even falls consistently, but less with defense than without defense. Also, the relative APL increases, but less with defense than without defense. Note that the ordinate range of the bottom right graphic is half the ordinate range of the bottom left graphic.

A careful comparison shows that, with the path centrality measures used, unweighted undirected networks are slighly more resistant than weighted undirected networks to attackers with low knowledge (up to 40%). For attackers with high knowledge (80% or more), the contrary happens.

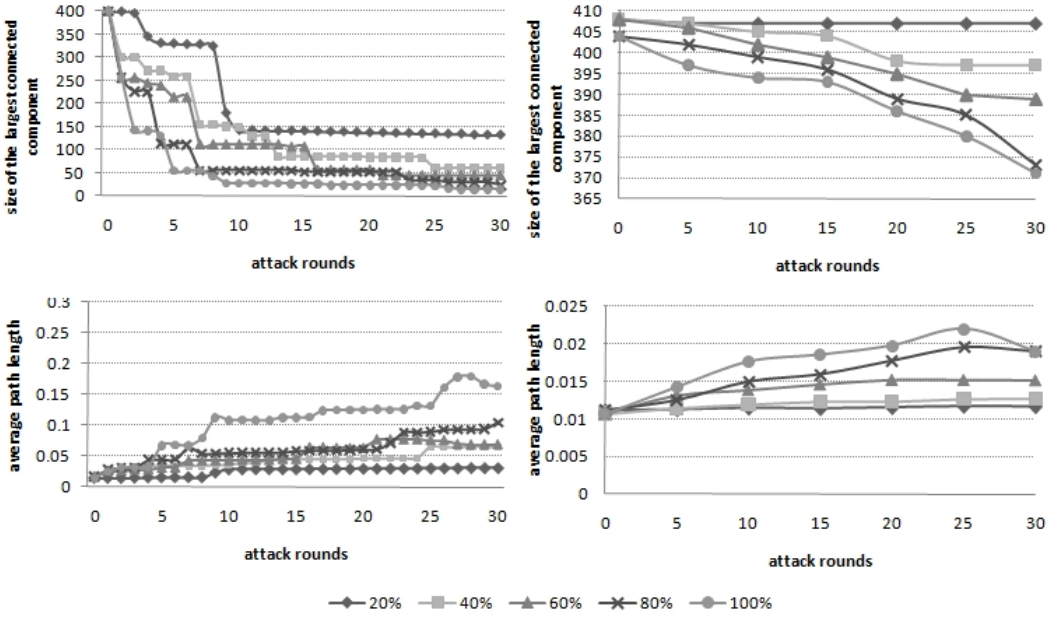


Fig. 5. Success metrics of attacks on weighted directed networks with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 5$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 5$).

4.5.4 Weighted directed network

The results for this network type, depicted in Figure 5, are very similar to those for unweighted directed networks (Section 4.5.2). At a closer look, they are even a little bit better (for the defender).

The effectiveness of the defense strategy can be justified by comparing the left-hand side graphics and the right-hand side graphics of Figure 5 (note that the ordinate ranges of those pairs of graphics differ). The argument is parallel to the one given in Section 4.5.2 about Figure 3.

When the attacker's network knowledge is only 20% and defense rounds are used, the LCC size starts with 408 (due to initial delegation) and it is restored to 407 by the first defense round after the fifth attack round. Exactly the same happens in subsequent defense rounds, which manage to neutralize the successive attack rounds. For a 20% knowledge attacker, the relative APL stays constant with the successive rounds when defense rounds are interleaved.

Attackers with 40%, 60%, 80% and even 100% network knowledge do less harm than an attacker with 20% knowledge did on an unweigthed undirected network. In all cases, the LCC size stays now above 370 nodes (92% of the original network size) if defenses are used. The relative APL stays in all cases pretty low: after 30 attacks by a 100% knowledge attacker it barely surpasses 2% of the LCC size (and this LCC size is 92% of the original network size).

4.5.5 A snapshot of the Internet network

To compute the effect of attacks and defenses on the 22963-node unweighted and undirected Internet snapshot network described in Section 4.1, we used a cluster formed by six computers running under the Linux operating system. Unlike the other networks above, which could be simulated in a single PC, this network was simply too big for its simulation to be handled by a single machine.

Figure 6 presents the results of using on this large network the same attack and defense parameters as in the previous smaller networks: 30 attack rounds and defense with $q = 5$.

By comparing the left-hand side graphics of Figure 6 with the corresponding right-hand side ones, we see the effectiveness of the clique defense rounds performed after each batch of $q = 5$ attacks. In fact, the attacks without defense are surprisingly good at reducing the LCC: by just destroying 30 nodes out of 22963, an attacker with 100% network knowledge manages to bring the LCC size well below 18000 nodes (23% reduction). Even attackers with less knowledge substantially reduce the LCC. As to the relative APL increase, substantial success is only achieved by the 100% attacker.

The use of defense rounds is clearly effective at restoring the LCC size; the usefulness is more apparent against attackers with more knowledge. If we focus on the worst case, in which the attacker has a 100% network knowledge, after 30 attacks rounds with 6 interleaved defense rounds, the LCC size decreases moderately from a value of 22979 (the initial 22963 nodes are increased a bit due to the initial delegation) to 20904 (top right graphic of Figure 6). This amounts to a reduction of only 9% with respect to the initial value.

In what regards the relative APL, for a worst-case attacker with 100% knowledge, it increases from an initial value of 1.109×10^{-5} to 1.790×10^{-5} after 30 attack rounds interleaved with 6 defense rounds (bottom right graphic of Figure 6). Just compare with what happens without defense (bottom left graphic), where the relative APL approaches 5.00×10^{-5} after 30 attack rounds.

In Figure 7, we chose to maintain the same proportion between the defense clique size q and the network size we had in the smaller networks. Therefore, we chose a clique size

$$q = \frac{5}{400} 22963 \approx 287$$

The number of defense rounds was kept at 6, just like in the previous simulations, but $q = 287$ network nodes had to be destroyed between two consecutive defense rounds. Thus, $287 \times 6 = 1722$ attacks were performed.

As it could be expected, the left-hand side graphics of Figure 7 show that,

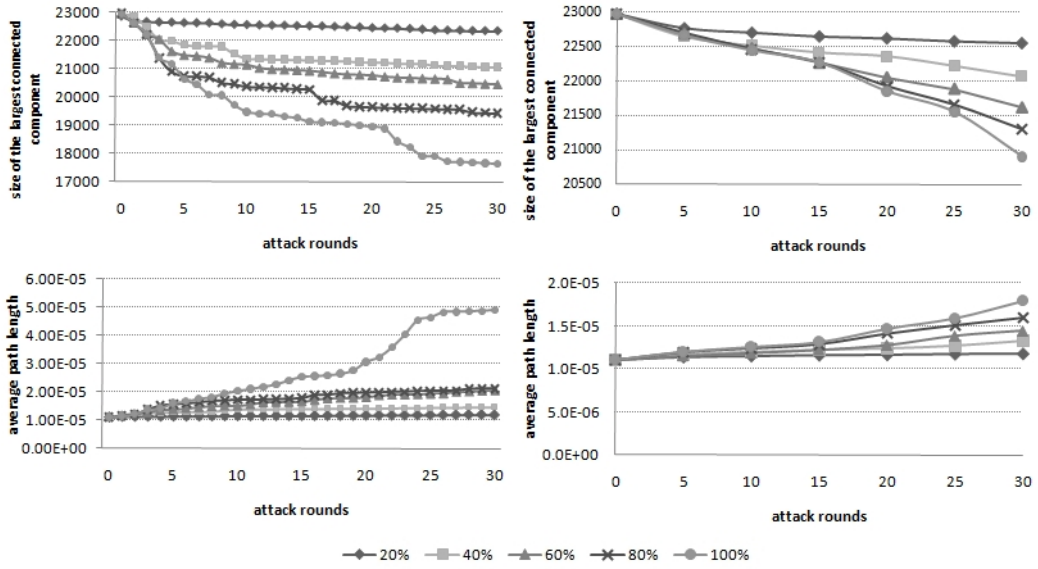


Fig. 6. Success metrics of attacks on a 22963-node Internet snapshot network with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 5$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 5$).

when no defense rounds are interleaved, the network is substantially damaged by the 1722 attacks, especially when the attacker has 100% network knowledge: for that attacker, already after the 861th attack, the LCC size goes down to 0 and the relative APL experiences a steep increase. For attackers with less knowledge, the LCC size decrease and the relative APL increase are gentler, but more substantial than in the corresponding graphics of Figure 6 (due to the ordinate scale, the relative APL increase seems to stay at zero in the bottom left graphic of Figure 7, but in reality it experiences an increase consistent with the one in the bottom left graphic of Figure 6).

Introducing defense rounds mitigates the reduction of LCC size as shown in the right-hand graphics of Figure 7. In the worst case, for an attacker with 100% knowledge, the LCC size remains nearly 5000 after the 1722 attacks, and the relative APL reaches less than a hundredth of the value it reaches when there is no defense.

If one compares the effects of defenses in Figures 6 and 7, that is, for $q = 5$ and for $q = 287$, it can be seen that the effect of increasing q is similar to the one we saw in Figure 1 above. A greater q implies that the initial sharp decline of LCC size lasts longer and the stabilization or gentle decline come later. However, provided that the knowledge of the attacker is not 100%, even for large q, successive defense rounds manage to keep a substantial part of the network connected. Therefore, if the available defense resources allow it, choosing a smaller q (*e.g.* $q = 5$) is preferable: this corresponds to the intuitive good practice of fixing problems as soon as possible.

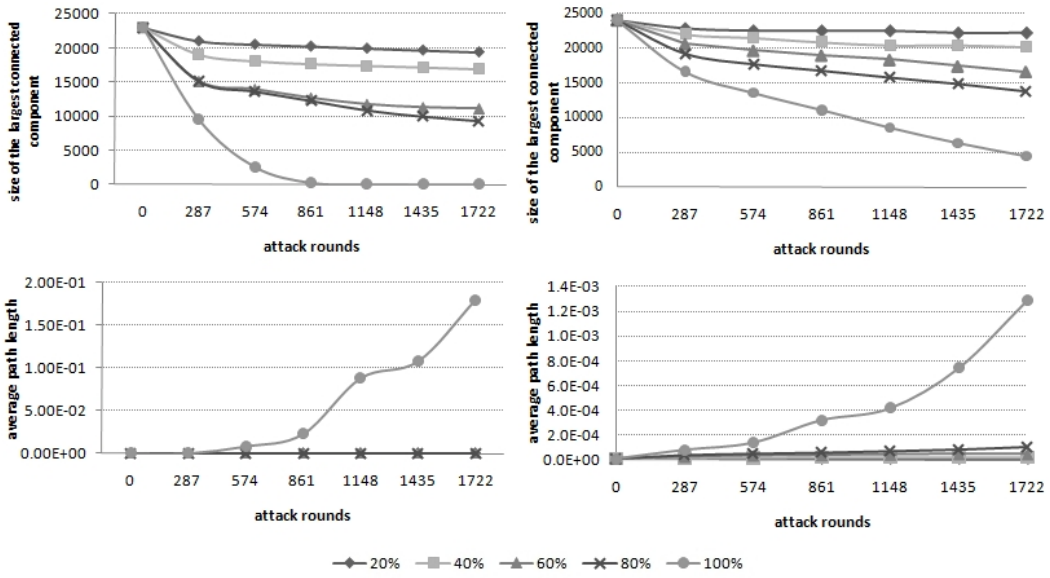


Fig. 7. Success metrics of attacks on a 22963-node Internet snapshot network with varying degrees of attacker's knowledge. Top left: LCC size (no defense used). Top right: LCC size (defense with $q = 287$). Bottom left: relative APL (no defense). Bottom right: relative APL (defense with $q = 287$).

All in all, the results obtained on this large real network are consistent with those obtained in the previous smaller networks. They show that the applied defense strategies provide a good network protection. By scheduling defense rounds to fix damaged portions of the network as soon as possible, even an attacker with 100% network knowledge can be largely kept at bay.

5 Conclusions

The novelty of this paper is to explore iterated attack and defense in scale-free weighted and/or directed networks and to do this by considering costs: there are bounds to what an attacker and a defender can do in each round. The scarce previous work in this area deals only with unweighted undirected networks and does not clearly say how much attackers and defenders can do. Scale-free weighted networks model a variety of real-life networks, including peer-to-peer networks, trust-weighted social networks, sensor networks and even terrorist organizations. In each of those applications, the semantics of the weights attributed to the edges may differ: weights may represent bandwidth, trust, distance, etc.

Attacks in the literature are based on node degree or node betweenness centrality. The presence of weights calls for path centrality attacks taking weights into account, which are introduced in this paper. As to defenses, we consider

delegation and node replenishment based on cliques. The cost of attack and defense strategies has been quantified, which was something lacking in the previous literature. An extension for directed networks has been also given. Finally, the impact of the various strategies has been empirically quantified.

Empirical results show that there are no significant differences in the resistance of unweighted and weighted networks, although we used different path centrality measures to attack them. There are however differences between undirected and directed networks. Due to the way we constructed the directed scale-free networks used in simulation (they are scale-free both in the in-degree and out-degree of nodes), these were more connected and had more resistance than undirected networks. One might argue that, for the same size and topology, a connected component in an undirected network has two-way paths and hence its connectivity is of "better quality"; yet, our results regarding path length (which take one-wayness into account for directed networks) contradict this intuition, because the APL in directed networks is not greater. Therefore, comparing undirected and directed networks by their LCC size is fair.

We conclude that, regardless of the network type, when the attacker knows only a 20% fraction of the network topology, her attacks are not very harmful (even those attacks based on path centrality). This emphasizes the importance of obfuscating the network topology.

If the network topology cannot be obfuscated, we have shown that directed networks which are scale-free both in the in-degree and out-degree of nodes can successfully withstand attacks where the attacker knows as much as 80% or even 100% of the network topology.

Acknowledgments and disclaimer

Special thanks go to one of the anonymous reviewers for insightful and helpful comments. We also appreciate Prof. Alexandre Arenas's advice about available data on large real networks. The authors are with the UNESCO Chair in Data Privacy, but the views expressed in this paper are their own and do not commit UNESCO. This work was partly funded by the Spanish Government through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", and by the Government of Catalonia under grant 2009 SGR 1135. The first author is partly supported as an ICREA-Acadèmia researcher by the Government of Catalonia. The authors are with the UNESCO Chair in Data Privacy, but the views expressed in this paper are their own and do not commit UNESCO.

References

[1] C. Aguilar-Melchor and Y. Deswarte, "Trustable relays for anonymous communication", *Transactions on Data Privacy*, vol. 2, no. 2, pp. 101-130, 2009.

[2] R. Albert, H. Jeong and A. L. Barabási, "Error and attack tolerance of complex networks", *Nature*, vol. 406, no. 1, pp. 387-482, 2000.

[3] R. Anderson, T. Moore, S. Nagaraja and A. Ozment, "Incentives and information security", in *Algorithm Game Theory* (eds. N. Nisan, T. Roughgarden, É. Tardos and V. V. Vazirani), chap. 25, pp. 633-649, Cambridge University Press, 2007.

[4] U. Brandes, "A faster algorithm for betweenness centrality", *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163-177, 2001.

[5] J. Domingo-Ferrer, A. Viejo, F. Sebé and U. González-Nicolás, "Privacy homomorphisms for social networks with private relationships", *Computer Networks*, vol. 52, pp. 3007-3016, 2008.

[6] J. Domingo-Ferrer, "Weighted Network Decapitation: The Economics of Iterated Attack and Defense", in *PAVLAD 2009*, Hong Kong, 2009.

[7] L. C. Freeman, "A set of measures of centrality based on betweenness", *Sociometry*, vol. 40, no. 1, pp. 35-41, 1977.

[8] M. George, "B-A scale-free network generation and visualization", Aug. 12, 2006 (updated Apr. 27, 2007). `http://www.mathworks.com/matlabcentral/fileexchange/11947`

[9] P. Holme, B. J. Kim, C. N. Yoon and S. K. Han, "Attack vulnerability of complex networks", *Physical Review E*, vol. 65, art. no. 018101, 2002.

[10] M. Newman, "Network data: Internet", July 22, 2006. `http://www-personal.umich.edu/~mejn/netdata/`

[11] S. Nagaraja and R. Anderson, "The topology of covert conflict", in *Proc. Fifth Workshop on Economics of Information Security*, June 2006.

[12] J. A. Rodríguez-Velázquez, A. Kamišalić and J. Domingo-Ferrer, "On reliability indices of communication networks", *Computers and Mathematics with Applications*, vol. 58, no. 7, pp. 1433-1440, 2009.

[13] L. A. Zhao, K. H. Park and Y. C. Lai, "Attack vulnerability of scale-free networks due to cascading breakdown", *Physical Review E*, vol. 70, art no. 035101, 2004.