Submission to DSN-2002 – Main Track: Dependable Computing and Communications

When Ants Attack: Security Issues for Stigmergic Systems

Weilin Zhong
University of Virginia
Department of Computer Science
151 Engineer's Way, P.O. Box 400740
Charlottesville, Virginia 22904-4740
USA
Phone: (+1) 434 9822292
Fax: (+1) 434 9822214
Email: weilin@virginia.edu

David Evans
University of Virginia
Department of Computer Science
151 Engineer's Way, P.O. Box 400740
Charlottesville, Virginia 22904-4740
USA
Phone: (+1) 434 9822218
Fax: (+1) 434 9822214
Email: evans@virginia.edu

Abstract

Inspired by biology, stigmergic systems solve global problems by using indirect communication mediated by an environment. Because they are localized and dynamic, stigmergic systems can produce complex distributed systems that are self-organizing, robust and adaptive. Properties of stigmergic systems raise new security concerns and opportunities. Indirect communication makes systems more vulnerable in an open and hostile environment, but also presents opportunities for resilient algorithms without the need for expensive cryptographic mechanisms. In this paper we use AntNet, an adaptive routing algorithm inspired by biological ant foraging, to explore some of the security issues for stigmergic systems. We identify possible attacks and analyze and report on results from simulation experiments. We propose a lightweight mechanism for defending against these attacks and evaluate its effectiveness.

Keywords: stigmergy, ant routing, security, swarm computing, attack models, secure routing protocols

Submission category: Regular paper

Word count: 5485

The material included in this paper has been cleared through authors' affiliations.

Contact author:

Weilin Zhong
University of Virginia
Department of Computer Science
151 Engineer's Way, P.O. Box 400740
Charlottesville, Virginia 22904-4740
USA
Phone: (+1) 434 9822292
Fax: (+1) 434 9822214
Email: weilin@virginia.edu

CARTER Award: NO

When Ants Attack: Security Issues for Stigmergic Systems

Weilin Zhong and David Evans
University of Virginia, Department of Computer Science

[weilin, evans]@virginia.edu

Abstract

Inspired by biology, stigmergic systems solve global problems by using indirect communication mediated by an environment. Because they are localized and dynamic, stigmergic systems can produce complex distributed systems that are self-organizing, robust and adaptive. Properties of stigmergic systems raise new security concerns and opportunities. Indirect communication makes systems more vulnerable in an open and hostile environment, but also presents opportunities for resilient algorithms without the need for expensive cryptographic mechanisms. In this paper we use AntNet, an adaptive routing algorithm inspired by biological ant foraging, to explore some of the security issues for stigmergic systems. We identify possible attacks and analyze and report on results from simulation experiments. We propose a lightweight mechanism for defending against these attacks and evaluate its effectiveness.

Keywords: stigmergy, ant routing, security, swarm computing, attack models, secure routing protocols

1 Introduction

Distributed systems that consist of a large number of nodes dispersed over a large-scale network are becoming common. The development of hardware technology will enable the construction of complex systems such as sensor networks from massive numbers of small computing and communicating devices. In these systems, numerous simple and locally interacting units collaborate to achieve complex system tasks. We call such systems *computing swarms***.** Computing swarms pose many new challenges: the limited resources of individual members restricts the amount of computation they may do; the high cost of long range communication permits only local communication with nearby neighbors; the huge number of members and need

for scalability exclude any centralized mechanisms. The key challenge here is to find an effective and efficient coordination mechanism that allows these locally interacting members to collaborate to achieve sophisticated global behaviors.

Research on biological systems, such as ant colonies, has provided much inspiration for the design of complex systems [Bonabeau99]. Social insects are well known for their complex group behaviors emerging from the cooperative behaviors of many small and simple members. Without any leader or centralized control, a swarm of social insects is able to collaborate to finish tasks that are far beyond the capability of any individual insect, such as finding food or building nest. As a group, social insects possess remarkable collective intelligence in solving complex problems. This intelligence lies in their interaction network, including direct interaction among members and indirect interaction mediated by the common environment. The indirect interaction is achieved by altering and sensing the common environment, often by secreting chemical pheromones and altering behavior based on the sensed pheromone concentration. This indirect communication mechanism using a shared environment is known as *stigmergy* [Grassé59]. An example of stigmergy is exhibited by ants that deposit a trail of pheromone on the way back from a food source, thereby recruiting more ants to follow this trail to the food source. A large number of ants following this process will find the quickest path to the food source since that path will build up the highest concentration of pheromone.

Stigmergy can also be an efficient communication model and coordination mechanism for large scale distributed systems. By employing this model, systems are able to achieve many desirable features, in particular they can achieve organization without the need for centralized control and they can automatically adapt to changes in their environment. Stigmergy has been applied to various complex systems, including communication network routing problems [DiCaro98a,

DiCaro98b, Scho96, White97], distributed intrusion detection and response [Fenet01], graph exploration [Yanovski01] and terrain coverage [Koenig01].

The distribution and locality of stigmergic systems make them intrinsically resilient to certain classes of attacks. On the other hand, stigmergic systems may be vulnerable to new kinds of attacks. The indirect communication characteristic of stigmergic systems offers malicious intruders opportunities to wreak havoc not available with traditional systems. To our knowledge, this is the first work to study the vulnerabilities of stigmergic systems operating in hostile environments.

In order to study the security issues and opportunities for stigmergic systems, we use AntNet as representative system. AntNet [DiCaro98a, DiCaro98b] is an adaptive routing algorithm based on the foraging behavior of biological ants. In AntNet, routers periodically send out *ant* agents to explore the network and collect network information. Routers maintain probabalistic routing tables with entries indicating the *goodness* of each link. These probability values are updated by every returning ant so that a better links obtain higher probabilities. By modifying these routing tables collectively and persistently, ants collaboratively solve the global routing problem: good paths are discovered and reinforced among all the routing tables.

The AntNet routing algorithm does not incorporate any security mechanisms to protect and verify the information carried by ant agents. In a hostile environment, this makes it vulnerable to some attacks. At the same time, the desirable properties of stigmergy point towards a resilient system without using any heavyweight cryptographic security mechanisms.

This paper introduces the potential security issues and opportunities in AntNet. The next section introduces the AntNet routing algorithm and illustrates its desirable properties towards a resilient algorithm. Section 3 analyzes the vulnerabilities of AntNet and identifies three effective attacks. In Sections 4-6, we report on results from simulating these three attacks on AntNet and propose ways to mitigate these attacks. Section 7 discusses the locality property of attacks in AntNet and defines the critical region, which helps to reduce the application of heavyweight security mechanisms against attacks. Section 8 discusses related work.

2 AntNet

AntNet [DiCaro98a, DiCaro98b] is an adaptive routing algorithm inspired by the stigmergy model in ant colonies. It uses mobile agents (ants) to cooperatively maintain routing tables. The routing table of a node *k*, organized as in distance-vector algorithms [Kurose01], stores a probability value P_{dn} for each pair (*d, n*), where *d* is every possible destination node from *k*, and *n* is every neighbor of *k*. The probability value P_{dn} expresses the goodness of choosing *n* as next node when the destination node is *d*.

AntNet nodes periodically send out mobile agents known as *ant packets* into the network to explore paths to a specified destination. There are two kinds of ant packets: *forward ants* and *backward ants*. Forward ants are sent out from a node to a specified destination, chosen based on the locally generated traffic pattern. A forward ant explores the network to find a feasible and low-cost path according to specified criteria, recording every node it visits. Loops in the path are noticed if it encounters the same node twice, and removed from its path record. Once it arrives at the destination, it is converted into a backward ant. The backward ant returns to the source node following the path in reverse. The goodness of this path is measured based on the estimated link latency at each node. Corresponding probability values for the links on the path are changed accordingly in the local routing tables for every node on the path from source to destination. Ant

packets are transmitted using separate queues from data packets, so they are not delayed by normal network congestion. Instead, the returning backward ant estimates the link latency based on the local workload and queue length at each node along the path.

As in biological ant foraging, the path an ant has just explored is positively reinforced by increasing the probability value for every link on this path. The fastest path will be reinforced by a positive feedback mechanisms as more ants travel along it. Indirect communication plays a critical role. Ants interact and communicate indirectly by updating the local routing tables, thus collaboratively solve the global network routing optimization problem.

2.1 Adaptability

With probabilistic routing, AntNet can dynamically balance workload among multiple paths. Because ants are continuously collecting path information and exploring new paths, AntNet is able to adapt to changes in network topology and traffic load. Now we present results from two experiments that demonstrate AntNet's adaptability. We simulate AntNet using OMNET++ [Varga01]. We use the simple network topology shown in Figure 1. Each link is bidirectional and all link parameters (transmission rate and propagation delay) are identical. The link propagation delay is set at 1ms. Both the network flows and ants are generated by node 0 only, destined for node 4. Ants are generated every 100ms and contain 192 bits. Data packets are generated at a constant rate with an average size of 1024 bits. We name paths by listing the nodes traversed in

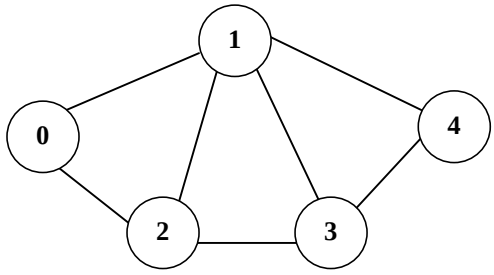

Figure 1. Experimental network.

the forward direction. Hence, path 014 is the best path when there is no congestion. We will mainly consider the probability values on routing tables and evaluate the results based on packet latency (and occasionally throughput).

2.1.1 Finding The Best Path

The first experiment demonstrates that AntNet finds the best path in normal network workload. The data generation rate is 0.2Mbit/sec and the link bandwidth is 2Mbit/sec. Since there is no delay caused by buffering and the data packet size is 1024 bits, the single link latency will be 1.6ms. The packet latency is 3.2ms for path 014 and 4.8ms for path 0234. Data packets are routed based on the probability values in the routing table. Once the value for a link reaches the threshold (set as 0.7), all the data packets will choose that link. The results of the experiment (shown in Figure 2 and 3) demonstrate that path 014 is quickly found as the best path and the latency quickly approaches the optimal 3.2ms.

2.1.2 Handling node failure and recovery

To demonstrate the adaptability of AntNet, we make node 1 fail (drop all incoming packets) after 200 seconds, and recover 300 seconds later. AntNet adapts quickly to node failure and recovery, as illustrated in Figures 4 and 5.

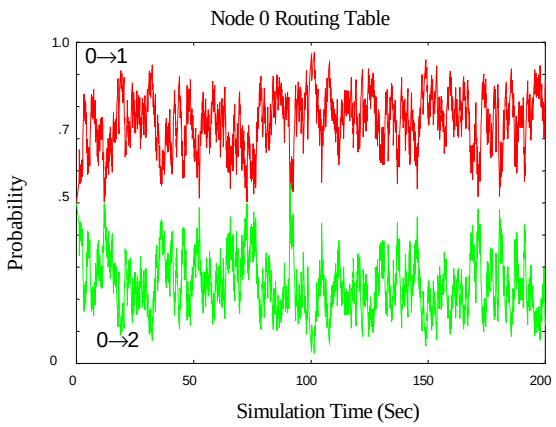

Figure 2. Routing table for node 0. Link 0→1, with a probability value higher than 0.7, will always be chosen for packets destined for node 4.

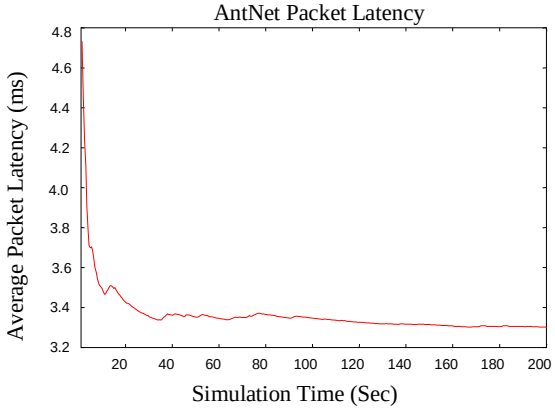

Figure 3. Average packet latency. The average latency approaches 3.2ms, the latency of path 014.

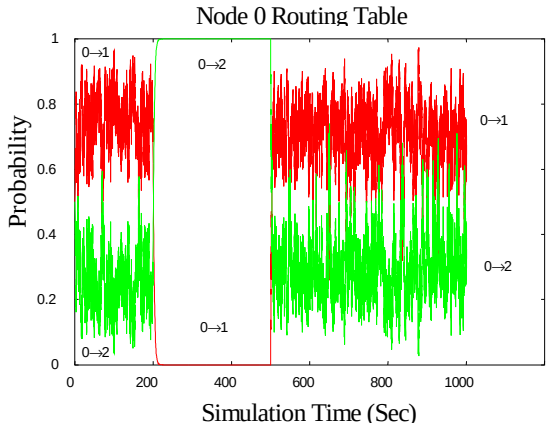

Figure 4. Failure and recovery. Node 1 fails at the 200th sec and recovers at the 500th sec. Path 014 has probability 0 during the failure; while path 0234 has probability 1. After recovery, the best route is restored.

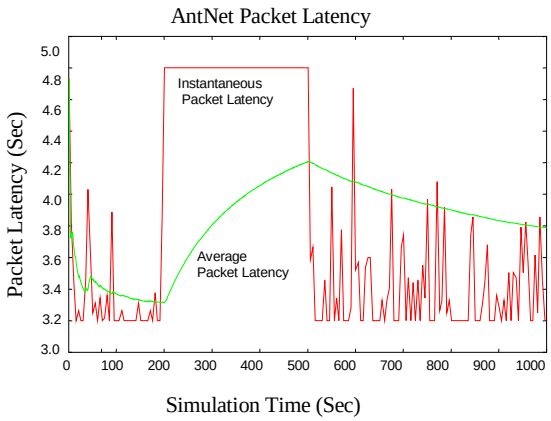

Figure 5. Average and Instantaneous Packet Latency. Latency switches between latencies of path 014 and path 0234 during node 1 failure and recovery.

2.3 Resilience

The stigmergic approach gives AntNet many desirable security properties that make it resilient to certain attacks. The adaptability of AntNet enables it to quickly route around failed nodes. Probabilistic routing and multiple paths increases the fault-tolerance of AntNet under attacks. Once a faulty node is repaired, network function quickly returns to normal. The effect of a single attack or attacks highly dispersed over time is negligible. An intruder must send out enough number of ants with a high enough frequency during a relatively long time period. This can make an attack very expensive and easy to trace.

3 Vulnerabilities and Attacks

Currently, there is no security mechanism in AntNet to protect and verify the routing information carried by the ant agents. If AntNet is used in a hostile environment, it would be vulnerable to various attacks. We focus on threats due to a compromised node; similar threats would result if a link was compromised and an intruder was able to inject or tamper with ant packets on the wire. We assume that a node subverted by an intruder can monitor, fabricate, replay, modify and delete

ant packets. The routing information itself is not considered confidential so we ignore routing information disclosure threats and focus on integrity.

There are two properties of AntNet that facilitate certain attacks. First, routing tables are updated based on the path information carried by backwards ants. Without any protection of this path information, a malicious node can very easily tamper with both the network topology and trip time information by altering passing ants or by generating bogus ants with false path information. Second, forward ants are routed based on the goodness measurement of each link, which is based on the activities of previous ants. Although this is the key for ant coordination and cooperation in stigmergy, this property makes AntNet more susceptible to attack. An intruder can take advantage of this positive feedback to attract more ants and enhance the effectiveness of an attack.

The potential threats towards routing functions can be classified according to the attacker's goals:

1) Increase latency of particular packets,
2) Decrease overall network throughput,
3) Break down a particular node or link, or
4) Divert packets away from certain links to usurp link bandwidth.

All the attack goals listed above can be achieved by attracting data packets to go through particular paths. Packet latency can be increased by attracting packets to a slower path. Network throughput can be decreased by attracting data packets towards a malicious node that simply drops those packets. Flooding a node or link can be achieved by making a path appear to be the best path even when there is congestion, thus counteracting the balancing capability of AntNet. Packets can be diverted away from a link by making the path appear to be slow. In general, we consider attacks that perturb the probability values in routing tables to be effective attacks. These

attacks lead to changes in packet latency and throughput. The three most basic attack mechanisms available to an attacker who has compromised a node are to:

1) Fabricate ant packets,
2) Drop ant packets, or
3) Tamper with information in ant packets.

In the following sections, these three attacks are simulated and analyzed. We also propose some solutions to mitigate the impact of these attacks.

4 Fabrication Attacks

An attacker who compromises a node or link can inject fabricated ant packets into the system or replay observed ants. We simulate this attack using the network topology shown in Figure 1. A subverted node 2 begins generating bogus ants at the 200th second. It injects ten bogus backward ants for every incoming backward ant to falsely promote the link 0→2. The bogus ant has a path 0234 and a trip time 4.8ms, which is the optimal trip time of path 0234 without congestion. We can see from Figures 6 and 7 that node 2 can easily deceive node 0 into believing link 0→2 was the best link towards node 4.

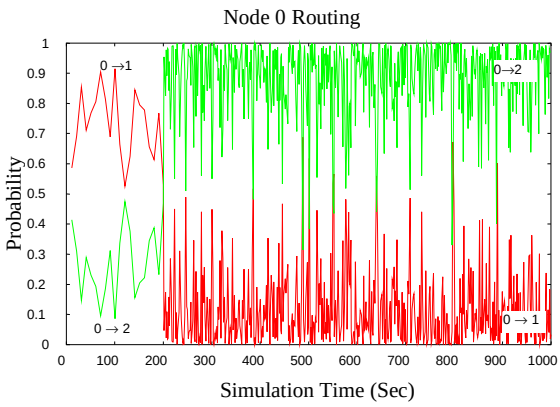

Figure 6. Node 2 generates bogus ants with path 0234 starting from 200th sec. Link 0→2 achieves probability higher than threshold value 0.7 soon and becomes the choosen path.

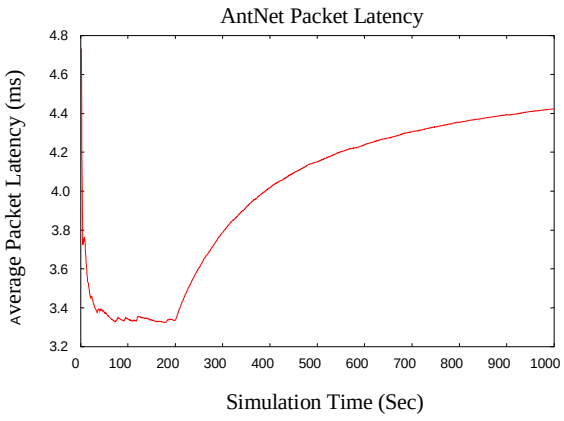

Figure 7. Node 2 generates bogus ants starting from 200th sec. Average packet latency goes up to the path latency of path 0234 because nearly all data packets are routed through path 0234.

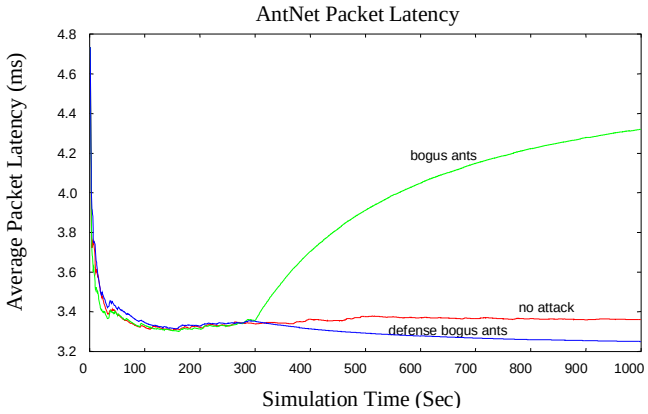

Figure 8. Average packet latency of three scenarios: no attack, bogus attack, and attack with defense method

The easiest way to defend against a fabricated or replayed ant attack is to simply record legitimate ant packets by assigning them unique identifiers. We can uniquely identify each ant by the tuple <*source*, *time stamp*> where source uniquely identifies the node that generates the ant and the time stamp is a local counter on the generating node. Each node maintains a list of all passing forward ants and only accepts those backward ants whose identifier is contained in that list. Once a legitimate backward ant arrives its identifier is deleted from the list, thereby preventing replay attacks. Backward ants that do not have a valid ID are dropped and ignored. To prevent denial-of-service attacks, entries in the list should time out if a corresponding backward ant does not arrive within a threshold time. Figure 8 demonstrates that the ant ID mechanism effectively defends AntNet from a bogus ant attack.

5 Drop Attacks

When an attacker subverts a node on the best path between two points, it can discredit good paths by selectively dropping ant packets. Note that if the compromised node is not on the best path, dropping ant packets is usually not an effective attack, since it would just repel the traffic from this path, which would actually make AntNet find the best path more quickly. To illustrate how dropping ant packets can be effective, we use a revised network topology shown in Figure 9 with

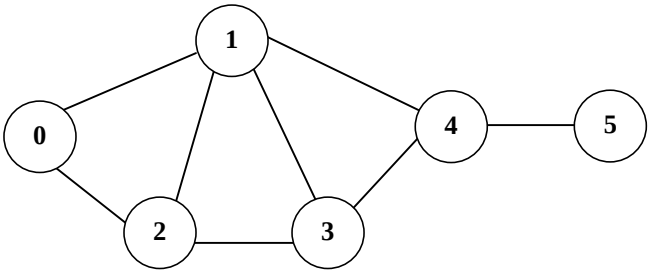

Figure 9. Network topology for drop ant packets experiment.

both data and ants flowing from node 0 to node 5. We simulate the attack where node 4 selectively drops ant packets that have visited node 1. Under this attack, path 014 is not reinforced and will soon be abandoned. Data packets will all be routed through path 0234. The result of this attack is demonstrated by node 0's routing table shown in Figure 10 and the average packet latency under attack compared with packet lantency without attack shown in Figure 11. Node 4 could use this attack to usurp the link between itself and node 1. Instead of just harming another node, this attack directly benefits the attacker by removing competition for network resources.

Dropping ant packets is not easy to detect and is often indistinguishable from real network failure. The effectiveness of dropping ant attacks is limited by the location of the compromised node, as

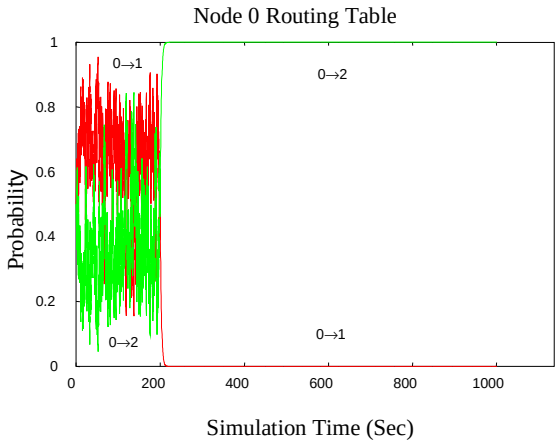

Figure 10. Drop ants attack. Node 4 drops forward ants that have visited node 1. Link 0→1 is soon abandoned with probability 0.

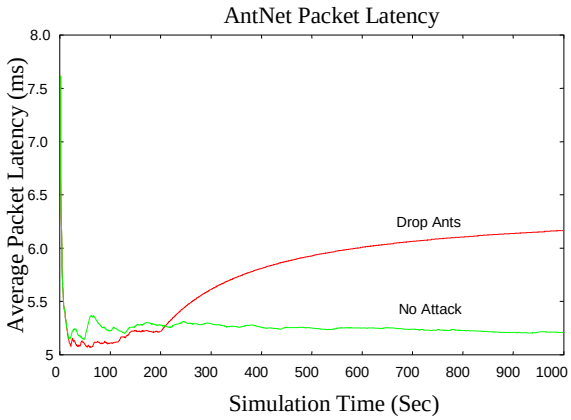

Figure 11. Average Packet Latency. Node 4 drops forward ants that have visited node 1. Packet latency under attack will soon approach 6.4ms, the trip time of path 02345.

discussed further in Section 7.

6 Tamper with Passing Ants

A backward ant records the path trip time by maintaining the sum of the local link latency estimates along the reverse path as illustrated in Figure 12. Beginning at the destination node, the trip time is set to 0. When a backward ant arrives in a node *x* coming from an adjacent node *y*, the link latency *L(x, y)* of the link $x \rightarrow y$ is estimated based on the local workload and queue length and added to the trip time $T_{y \rightarrow dest}$ carried by this ant to get $T_{x \rightarrow y}$: $T_{x \rightarrow y} = L(x, y) + T_{y \rightarrow dest}$. When a backwards ant reaches the source node, the whole trip time of source node to destination node, $T_{src \rightarrow dest}$ is known.

Given network topology as in Figure 1, single link latency is 1.6ms without congestion. Suppose a malicious attacker compromises node 2 and can tamper with passing backward ants, setting up the trip time $T_{2->4}$ to 0ms (or a negligibly small value; a negative value may also be possible but can be easily detected). At node 0 the trip time of path 0234 is calculated as 1.6ms instead of 4.8ms, making path 0234 appear faster than path 014. The experiment shown in Figure 13 illustrates what happens when node 2 tampers with the trip time information carried by passing ants. Beginning at the 200th second, node 2 alters the recorded trip time on the return path from node 4 to be 0ms. This attack is effective in switching the routing probabilities at node 0 to direct all data packets through the compromised node.

To defend against tampering attacks requires authentication and integrity of path information

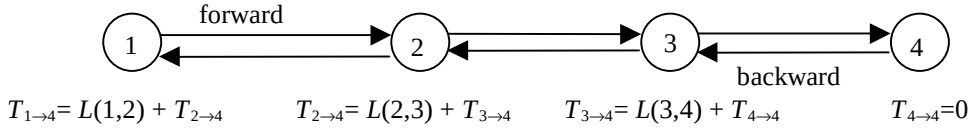

Figure 12. Backward ant estimates trip time.

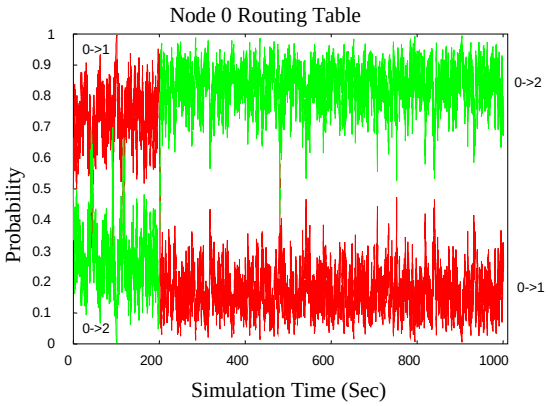

Figure 13. Trip time tampering attack. Node 2 tampers with passing ants. Link 0→2 achieves probability higher than threshold 0.7 soon and becomes the exclusively chosen path.

carried by ants. Lightweight techniques, such as comparing the reported trip time with the actual two-way trip time calculated by comparing the time the backward ant arrives with the time the corresponding forward ant was generated, would limit the effectiveness of tampering attacks to some degree. However, they cannot prevent them entirely since the ant packets use separate queues to avoid network congestion. It is possible, even without tampering, for an ant to return faster that its recorded trip time so there is no way to distinguish tampered ants.

Hence, a defense against tampering attacks requires cryptographic techniques. Our defense uses digital signatures combined with ant identifiers, as illustrated in Figure 14. Each node has a public-private key pair whose public key can be securely obtained by all other nodes. Along the forward path, each node checks the path information and adds a signed tuple containing the identifier of the forward ant, the node's identity, and the identity of the next node. The destination node signs the complete path. When the backward ant returns, each node signs the locally estimated link latency. The source node checks that a backward ant contains valid signed path and link latency information for all the intermediate nodes. Since each intermediate node signs the partial path, the source node can verify the path taken. A malicious node can only lie about the delay of the link between itself and its successor.

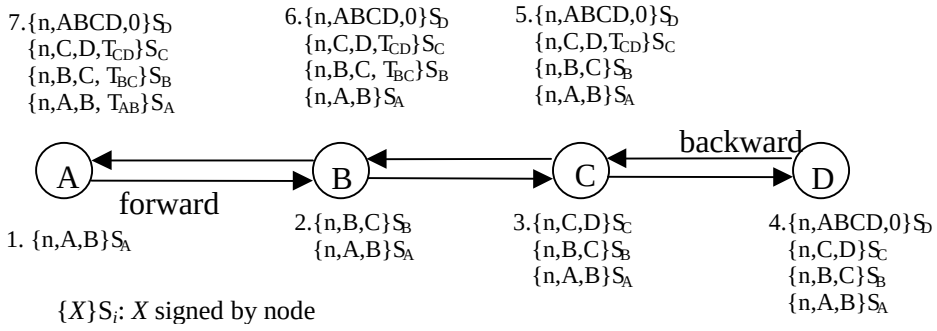

Figure 14. Defense Tamper Attack: Digital Signature to Path

The cryptographic techniques used to defend against ant tampering are too expensive for many applications. Further, this mechanism requires source node to check a number of signatures equal to the number of nodes in the path, which might cause problems in both ant size and computation overhead with the size of network grows. Fortunately, some desirable locality properties of stigmergy (discussed in the next section) limit the need for this expensive mechanism. Further, given the self-stabilization property of AntNet, a single attack will not be effective to AntNet so authentication can be required periodically over a certain population of ants depending on current security situation and required security level.

7 Locality of Damages

In AntNet, routing decisions are made locally based on the trip time information carried by backward ants. The effectiveness of an attack is highly dependent on the location of compromised nodes on the network.

Consider an attacker who wants to prevent data packets between a particular source and destination from being routed through the best path. An attacker who subverts a node along the best path between those nodes, can simply treat ant packets normally and drop or delay all data packets along that path. Hence, we can only defend against attackers who are not able to compromise a node along the best path. After compromising a node along an inferior path, the attacker's goal is to attract packets towards that path. There are two ways to achieve this: reduce

the trip time information carried by ants along this path or inject bogus ants. As discussed in Section 4, bogus ants can be easily detected using ant identifiers, so the only attack likely to succeed is to reduce recorded trip times. The capability of a malicious node to succeed with this attack is highly location dependent.

As discussed in Section 6, when a backward ant arrives at each node, the path latency from current node to the ant's destination is estimated by adding locally estimated link latency to the trip time information carried by this backwards ant. A malicious node can alter the trip time from itself to the target destination. The maximal lie is to record the trip time as 0. As long as the path between the source node and malicious node is trustworthy, a malicious node can do nothing to alter the trip time for this part of path.

Consider the network in Figure 15. Suppose link latency is 1 for all links. Here the best path is 014, whose trip time is 2. The maximal lie that malicious node 2 can tell is that the trip time from itself to node 4 is 0. So malicious node 2 can make path 0234 appear to have a trip time 1 at node 0, thus making path 0234 better than path 014. Instead, if node 3 were malicious, it could not tamper with the trip time on the return path from 3→0, which has a trip time 2, so it can only make path 0234 as good as path 014, hence, 50% of network traffic will go through this link. While if node 5 is malicious, it can do nothing to make the path 02354 better than the best path

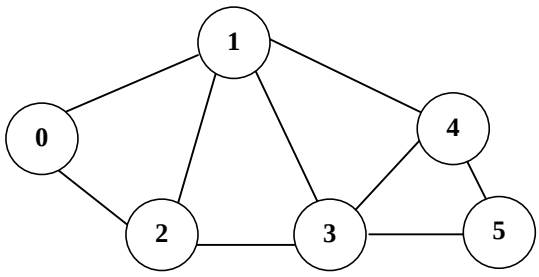

Figure 15. Network topology for locality.

014. Thus, if the trip time between a compromised node and the source node is longer than the trip time for the best path, the attacker can do nothing by tampering with passing ants to make the path going through it appear to be a better path than the currently best path. We call the subset of the network that can perturb route information between a source and destination node a *critical region*.

Given a pair of source and destination nodes, if best path between the source and destination node has trip time *t*, all nodes within trip time *t* are within the critical region (illustrated in Figure 16). The radius of a critical region represented by the hop count or trip time of the best path between source and destination node, can either be gained from known network topology or can be estimated by some statistic model maintained over time. Nodes inside the critical region are *internal nodes* while outside this region are called *external nodes*.

An external node cannot influence the traffic flows between the source and destination node by tampering with the trip time or by dropping ant packets. Hence, the three major attacks we have identified before will not have significant effect on traffic between two nodes if the compromised node is outside the critical region of these two nodes. Since the radius of critical region varies with the trip time of the best path, an external node can introduce some traffic congestion on the best path to enlarge the radius of the critical region so that it will be included in the critical

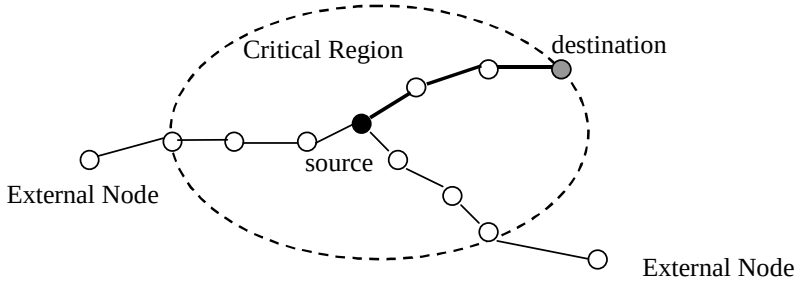

Figure 16. Critical region. The shortest path is 3 hops. With no congestion, nodes further than 3 hops away from source are external to the critical region between source and destination.

region, thus be able to attack the network effectively. This attack is highly dependent on network topology, and requires that the external node transmit a large number of packets. Once the critical region has grown to include the external node, it must continue to create the network congestion otherwise the routing information will quickly recover to the original best path.

The locality properties of AntNet, typical of stigmergic systems, provide strong security properties without any cryptographic mechanisms. Knowing that external nodes have limited capabilities to influence the traffic flow means authentication mechanisms need only be applied to nodes within this region, thus reducing the overhead of using expensive cryptographic mechanisms.

8 Related Work

To our knowledge, this is the first work to study security vulnerabilities of stigmergic systems. Several researchers have argued for the survivability of distributed, decentralized systems. Fisher and Lipson propose using *emergent algorithms* as a way to build survivable systems and present an Internet routing protocol based on this approach [Fisher98]. They define emergent algorithms as any computation that achieves predictable global effects by communication directly with only a bounded number of immediate neighbors and without any central control or global knowledge. Like stigmergic algorithms, emergent algorithms use local interactions to establish global properties.

Many researchers have investigated mechanisms for making routing protocols more secure. Most of this work builds on traditional distance vector routing protocols. AntNet is similar to distance vector routing protocols in that a node possesses only knowledge of local links and receives summaries of path latency information from its neighbors. Smith proposed a security solution to distance vector routing protocol by adding predecessors in routing update messages and signing

the update messages by the originating router's signature [Smith96]. Since distance vector nodes summarize the information they receive before transmitting their own information to the net, the ultimate source of the information cannot be determined and a subverted node could still fabricate destination and path latency information.

Vetter, Wang and Wu studied the effectiveness of compromised node attacks on the OSPF routing protocol [Vetter97]. They found it to be highly susceptible to disruption by a single compromised router and demonstrated that one subverted router can cause global damage to the network with minimal effort. Perlman developed shortest-path routing protocols that are resilient to faulty routers that depended public key infrastructure where each router signs outgoing messages [Perlman88]. Murphy and Badger proposed a digital signature scheme to prevent forged link-state advertisements [Murphy96]. Other researchers have developed more efficient schemes based on hashing that sacrifice some security for performance [Zhang98, Hauser99]. The cryptographic techniques described in Section 6 are similar, except with our scheme ants record all the intermediate nodes on the path. By signing each link alone the path, a single compromised node cannot fabricate path information. The only vulnerability left is the local link latency signed by the compromised node, which can be fabricated. In addition, because of the locality of AntNet we do not require a network-wide public key infrastructure. Cryptographic techniques can prevent link compromises from inflicting damage to the network, but if a node is compromised it is likely that the attacker will be able to obtain cryptographic keys stored on the node.

9 Summary

Stigmergic systems offer new opportunities for building secure systems by taking advantage of decentralized control and indirect communication. They also present attackers with new methods of disruption. We have studied a representative stigmergic system, AntNet, to analyze its

vulnerability to new attack methods. Our findings indicate that with minor modifications, AntNet can be resilient to all the new attack classes we identified except for attacks that involve tampering with the data carried by ants conducted by nearby compromised nodes. Because these attacks depend on locality, they are a less serious threat in most cases; in others, they can be prevented using cryptographic techniques. Stigmergy offers the possibility for efficient, adaptive and robust systems. It is important, however, to also keep in mind the new opportunities it presents for attackers. We have analyzed attacks on the integrity of a stigmergic system, and presented some simple defensive mechanisms for mitigating the effectiveness of those attacks. Additional work needs to be done on generalizing our analysis to other stigmergic systems, and considering other security issues including confidentiality and availability.

Acknowledgements

The authors thank Gianni Di Caro for providing source code to a preliminary implementation of AntNet, and graciously assisting us in adapting it for our experiments. This work was funded by the National Science Foundation ("Programming the Swarm", CCR-0092945).

References

[Bonabeau99] Eric Bonabeau, Marco Dorigo, Guy Theraulaz. *Swarm Intelligence: from Natural to Artificial Systems*, Santa Fe Institute, Oxford University Press, 1999.

[DiCaro98a] Gianni Di Caro, *AntNet: Distributed Stigmergic Control for Communications Networks*, Journal of Artificial Intelligence Research 9 (1998): 317-365.

[DiCaro98b] Gianni Di Caro. *Two Ant Colony Algorithms for Best-effort Routing in Datagram Networks.* In Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'98), edited by Y.Pan, S. G. Akl, and K. li, 541-546. Anaheim, CA: IASTED/ACTA Press, 1998.

[Fenet01] Serge Fenet, Salima Hassas. *A distributed Intrusion Detection and Response System Based on mobile autonomous agents using social insects communication paradigm*. In Proceedings of First International Workshop on Security of Mobile Multiagent Systems, 2001.

[Fisher98] David A. Fisher and Howard F. Lipson. *Emergent Algorithms - A New Method for Enhancing Survivability in Unbounded Systems*. Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences. 1998.

[Grassé59] Grassé, P.-P. *La Reconstruction du nid et les Coordinations Inter-Individuelles chez Bellicositermes Natalensis et Cubitermes sp. La théorie de la Stigmergie:Essai d'interprétation du Comportement des Termites Constructeurs.* Insect. Soc.6 (1959): 41-80.

[Hauser99] Ralf Hauser, Tony Przygienda and Gene Tsudik. *Lowering security overhead in link state routing*. Computer Networks, Volume 31 Number 8. 1999.

[Koenig01] Sven Koenig, B. Szymanski and Y. Liu. *Efficient and Inefficient Ant Coverage Methods*. Annals of Mathematics and Artificial Intelligence. Volume 31, Issue 1/4. 2001.

[Kurose01] James F. Kurose, Keith W.Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, 2001

[Murphy96] S. Murphy and M. Badger. *Digital Signature Protection of the OSPF Routing Protocol*. Internet Society Symposium on Network and Distributed Systems Security, 1996.

[Perlman88] R. Perlman. *Network Layer Protocol with Byzantine Agreement.* MIT PhD Thesis (available as MIT LCS TR-429). October 1998.

[Scho96] R.Schoonderwoerd, O.Holland, J.Bruten and L.Rothkrantz. *Ant-based load balancing in telecommunications networks.* Adapt. Behav. 5 (1996): 169-207.

[Smith96] Bradley R.Smith, Shree Murthy, J.J. Garcia-Luna-Aceves. *Securing Distance-Vector Routing protocols.* IEEE/ISOC Symposiums on Network and Distributed System Security, 1996.

[Varga01] András Varga. *OMNeT++: a discrete event simulation tool*. http://www.hit.bme.hu/phd/vargaa/omnetpp.htm.

[Vetter97] Brian Vetter, Feiyi Wang and S. Felix Wu. *An Experimental Study of Insider Attacks for the OSPF Routing Protocol*. 5th IEEE International Conference on Network Protocols, Atlanta, GA. IEEE press, October 1997.

[White97] T. White. *Routing with swarm intelligence*. Technical Report SCE97-15, Systems and Computer Engineering Department, Carleton University, September, 1997.

[Yanovski01] Vladimir Yanovski, Israel A. Wagner, Alfred M. Bruckstein. *Computer Vertex-Ant-Walk – A robust method for efficient exploration of faulty graph.* Annals of Mathematics and Artificial Intelligence. Volume 31, Issue 1/4. 2001.

[Zhang98] Kan Zhang. *Efficient Protocols for Signing Routing Messages*. Symposium on Network and Distributed Systems Security (NDSS). 1998.