Friend or Flood? Social prevention of flooding attacks in mobile opportunistic networks

Iain Parris and Tristan Henderson
School of Computer Science, University of St Andrews
St Andrews, Fife, KY16 9SX, UK
{isp3,tnhh}@st-andrews.ac.uk

***Abstract*—Opportunistic networks enable decentralised and infrastructure-less social networking applications, through the cooperation of peer mobile devices to forward messages on one another's behalf. The decentralised and cooperative nature of these networks, however, introduces potential security threats. For instance, malicious nodes may modify messages, or send many messages in an attempt to drain other nodes' resources and thereby disrupt the network. Such attacks are well-studied for wireless ad hoc networks, but may need reconsideration in disconnected opportunistic networks.**

In this paper we define a simple flooding attack that can deny service in an opportunistic network. We simulate the attack and demonstrate its efficacy using real-world datasets. We furthermore develop a scheme for mitigating the attack, by using the social relations between nodes. The scheme is lightweight, requires only local knowledge to be stored by each node, and is shown to be effective: for one dataset, the median proportion of time spent offline by nodes was reduced from 42.7% to 6.3%.

I. Introduction

Mobile devices are increasingly carried by people throughout their daily lives and used for applications beyond mere voice calls. One popular use of mobile devices is to access online social networks (OSNs), for example microblogging on Twitter.[1] Current OSNs are normally accessed on mobile devices via infrastructure networks such as cellphone towers or Wi-Fi access points. But in infrastructure-less scenarios, where infrastructure is unavailable or otherwise undesirable to use (e.g., due to cost, or an inability to trust the infrastructure such as in political uprisings), it is then not possible for mobile users to access OSNs.

Opportunistic networks can enable communication in such infrastructure-less scenarios, including the use of OSNs and other social applications [15]. Peer mobile devices may directly exchange messages when in physical proximity, via a wireless protocol such as Bluetooth, without requiring any fixed infrastructure. If many such peer devices — acting as network nodes — cooperate to carry each other's messages, then a decentralised opportunistic network is formed, in a disconnected store-carry-and-forward architecture. Routing and forwarding in such networks is well-studied, but one area with many remaining challenges is security [7]. The decentralised and cooperative nature of such networks, where there may be no traditional infrastructure, and where nodes are expected to cooperate and forward data for each other, introduces many new attack vectors and possibilities for malicious behaviour.

In this paper, we focus on one type of attack. Since it is difficult to determine reliably the sender of a message in an opportunistic network, a malicious user can untraceably flood the network with spoofed messages. As the available resources of participating devices (e.g., battery, storage or bandwidth) are finite, and may be drained by receiving and retransmitting these messages, this flooding will act as a denial-of-service attack against participating network nodes.

Our goal is to mitigate such a flooding attack, while maintaining the utility of the opportunistic network. In short, the core idea is for each message to be signed by the original sender. Each message is then only retransmitted by the trusted social contacts ("friends") of its original sender. A friend will retransmit only after checking the message signature to verify that the message's origin is their trusted friend. Such a defence is lightweight, relying only on local knowledge at each node.

The contributions of this paper are to: (i) formalise a flooding-based resource-consuming attack, and simulate the efficacy of the attack using real-world traces; (ii) use social network information to build a routing protocol using social network information that is resistant to the attack; and (iii) demonstrate through trace-driven simulation that the attack-resistant protocol mitigates the attack, while at the same time maintaining the utility of the network.

We next discuss related work, and present both the attack and an attack-resistant routing scheme in Section III. Sections IV and V evaluate the attack and attack-resistant scheme. Finally we conclude by discussing the implications and limitations of our results, and pointers for future work.

II. Related work

Our goal is to study a flooding-based denial-of-service (DoS) attack against opportunistic networks, and social mitigation of this attack. Security of traditional networks against flooding attacks has been well-studied. For example, Mirkovic and Reiher present a taxonomy for distributed denial-of-service (DDoS) attacks [19], with particular reference to the Internet. Belenky and Ansari survey IP traceback methods [1], used to combat Internet DoS by allowing identification of the source of malicious packets. The defence scheme that we introduce similarly relies on identifying the source of malicious packets, but via a different mechanism (cryptographic signatures).

Flooding attacks in less traditional mobile ad-hoc networks (MANETs) are also well-researched. Guo *et al.* present a detection mechanism for MANET flooding attacks [13]. Kim and Helmy introduce a framework to traceback such attacks [17]. Tan and Seah demonstrate a possible countermeasure, through statistical filtering [26]. These methods, however, do not extend to disconnected networks such as opportunistic networks.

[1] http://twitter.com/

Considering spoofed identities more generally, MANETs and opportunistic networks are particularly vulnerable to the Sybil attack [9], where a malicious node masks its identity, presenting multiple "fake" identities to the network. Where a centralised authority is present, such as in traditional OSNs, social network analysis may aid detection of these fake identities [6]. Where no centralised authority is available, detection is challenging. Piro *et al.* [23] present a possible method to detect Sybil attacks in ad-hoc networks by monitoring transmissions, while Yu *et al.* [28], use social network information to detect abnormality. The detection methods are not, however, efficient in disconnected opportunistic networks.

Chen *et al.* explicitly consider security in opportunistic networks [7]. They identify the class of flooding attack that we investigate in this paper (which they term a "hypernova attack"), and benchmark its impact against simulated datasets. They do not, however, study real datasets or possible attack mitigations, as we do here.

In a superset of opportunistic networks, delay-tolerant networks (DTNs), Uddin *et al.* investigate countermeasures for a different spoofing attack, where one node steals the identity of another in order to absorb packets intended for the victim. More closely related to the flooding attack considered in this paper, Burgess *et al.* [5] investigate mitigation of various DTN attacks, but focus on bandwidth saturation rather than node energy. Choo *et al.* [8] also investigate the robustness of DTNs to various attacks, but do not consider mitigation. Lee *et al.* research mitigating flooding attacks in DTNs [18] — but with a mechanism relying on probabilistic routing protocols only, rather than utilising social network information.

Our proposed scheme relies on leveraging trusted social contacts. Using trusted social contacts to improve security in DTNs has been described by El Defrawy *et al.*, but in the context of preserving privacy rather than maintaining availability [11]. Whitelisting messages from immediate social contacts has been introduced in the context of email by Garriss *et al.* [12], and extended for more distant social contacts by Hameed *et al.* [14] — but both rely on a centralised architecture, and do not generalise to decentralised networks. Trifunovic *et al.* investigate blocking unwanted spam messages in opportunistic networks [27]. Their scheme relies on assigning trust values to all nodes rather than trusting messages from only immediate social contacts, and also is dependent on the manual classification of spam message content. To our knowledge, we are the first to use trusted social contacts to mitigate an attack on the availability of opportunistic networks.

III. Attack model and defence

Our goal is to investigate the impact of a flooding attack on an opportunistic network, and to mitigate this attack. In a flooding attack, the attacker floods the network with messages. Network nodes receive and relay copies of these messages throughout the network, consuming their finite resources (such as battery) in the process. The intent of the attacker is to overload these finite resources, causing nodes to fail, and consequently degrading overall network performance.

In order to formalise this attack, we consider an attacker with certain, limited capabilities, which we enumerate and formalise within the following attack model.

A. Attack model

We consider the attack against Simple Social Network Routing (SSNR) [21]. In SSNR, each node has a set of friends. The original sender of each message embeds a copy of their list of friends within the message, as part of its headers. This friends list then informs the routing of the message through the network: if a node appears in this list, then it will relay the message. For redundancy, the message is multiply copied, and thus may take more than one path to reach its destination.

We make the following assumptions, inspired by [5], about the capabilities of the attacker:

1) *Spoofing messages:* Messages are clear text, so the attacker can spoof any header of the message — or the entire message.
2) *Identity:* The attacker can spoof their MAC-layer address to hide their network identity.

Making these assumptions, the attacker may perform a simple flooding attack. When encountering another node, the attacker can generate a new message. This message, however, has spoofed headers, falsely indicating that it should be routed via the node — i.e., the node will believe that it is relaying the message on behalf of one of its friends.

Worse, the attacker can additionally spoof the "friends list" (i.e., the set of nodes which should relay the message) header, with a permissive set of nodes. This allows amplification of the attack: after the attacker injects the initial message into the network — by sending to the encountered node — the message will then be relayed, consuming further resources without additional cost to the attacker. This amplification is crucial to the attack: a relatively small number of messages generated by the attacker may be amplified many times throughout the network, thus consuming disproportionate network resources.

To further increase the attack, the attacker may spoof multiple MAC-layer addresses, in a manner similar to the Sybil attack [9]. This allow the attacker to send a larger number of messages to each encountered node: the node cannot blacklist a single MAC-layer identifier which generates numerous messages in a single encounter, because the messages appear to have been sent from numerous other encountered nodes.

Finally, the attacker may set an undeliverable destination address for the message. This ensures that the message will propagate as much as possible through the network (i.e., consuming greater resources), since it will never be delivered.

B. Defence

Due to the spoofing of headers, the above attack is difficult to detect at any node, using only its local knowledge. There is no way to determine a message's true origin. Therefore, even if a particular message should somehow be identified as an attack message, this lack of accountability and traceback means that only this one message would be locally dropped; the attacker may continue flooding other messages, under a new identifier.

We therefore introduce a new security requirement and assumption. The intention is to enable a lightweight scheme, where nodes authenticate that messages which they are willing to receive and relay are truly generated by one of their friends.

We require a public/private key pair for each node. Each message is signed by its original sender, allowing any node knowing the sender's public key to verify the message origin.

One limitation of this scheme is that we require key distribution, when PKI may be unrealistic for a fully decentralised network [21], [11]. We note, however, that nodes have friends with whom they communicate, and we assume that the nodes locally know who their friends are. We further assume that friends know one another's public keys. These public keys may be shared between friends out-of-band of the opportunistic network without requiring a fully-fledged PKI: possibly in a physical meeting, by earlier communication via traditional networking infrastructure, or even via snail-mail.[2]

Since messages in the network are only relayed by the original sender's friends, each relay node can thus verify that the message sender is truly their trusted friend by checking the signature (Algorithm 1): if the message is not signed by their friend, then it has been spoofed and is discarded. This mitigates the flooding attack.

Algorithm 1 Message check: only accept a message for relaying if the original message sender is a trusted friend.

```
1: if friends_with(message's original sender) and
   has_valid_original_sender_signature(message) then
2:     accept message for relaying
3: else
4:     discard message
```

It remains possible, however, for a node with genuine friendship links to other nodes to flood messages into the network; these messages will be authenticated and relayed by the attacker's friends. But this is a more expensive attack: the attacker must create genuine "friendship" relations with the nodes being attacked, and faking such a social relation is more expensive than spoofing a message. Additionally, even if a node can "trick" other nodes into becoming friends with it, the attack may still be mitigated. Each network node can now detect the attack, by looking locally at the messages which it has received for relaying. Each message can be linked back to its original sender. If a particular sender has generated excessive network traffic then this node can be blocked (i.e., blacklisted for relaying messages). This means that network nodes either (i) block the attacker locally, if the attacker has been successful in generating abnormally much traffic at that node, or (ii) do not see abnormal traffic from the attacker (perhaps due to the attack being throttled), in which case the attack is also unsuccessful. Either way, the attack is mitigated.

We note that other, more limited, wireless attacks may still be possible. For example, the attacker may attempt to overwhelm a single proximate node by transmitting invalid messages to it at a very high rate, as in a jamming attack [22]. The energy usage by the individual node to receive these messages — even if the messages are then immediately discarded as invalid — may drain its battery and take the node offline. But this is a weaker attack, i.e., without message amplification throughout the network. We consider these targeted attacks on individual proximate nodes as out of scope for this paper; our focus is on mitigating flooding attacks with message amplification throughout the network.

IV. Evaluation

We now present an evaluation of the flooding attack on network performance — with and without the defence — against three real-world datasets. Following [20], [21], we conduct trace-driven simulation using a custom Python opportunistic-network simulator, for real-world datasets containing encounters and social network information.

A. Datasets

We use three real-world datasets; two were collected in our previous work. All datasets are publicly-available:

- The *SASSY* dataset [4]. Twenty-five participants were equipped with 802.15.4 Tmote Invent sensors, and tracked for 79 days. We augment the trace as detailed in [21], resulting in a dense trace of encounters between participants. Social network information was obtained from Facebook friendships: friendship in the defence scheme corresponds to Facebook friendship.
- The *LocShare* dataset [2]. Locations of 80 participants were collected during four one-week runs of 20 participants. Encounters were defined as occurring when participants passed within 10 meters of one another — the approximate Bluetooth range. As for *SASSY*, social network information was obtained from Facebook friendships.
- The *Reality Mining* dataset [10]. In this well-known dataset collected at MIT, 97 university members carried mobile phones for an academic year. As in [21], [20], we define Bluetooth encounters between participants as opportunities for message exchange, and use mobile phone address book contacts to determine social relations for each user.

B. Simulation parameters

In line with our previous work [20], [21], we use the following parameters for our simulations:

- 100 runs per data point.
- One week of simulation time per run.[3]
- Average of one (non-attack) message per node per day.
- Message TTL of one day.
- Energy model for node batteries (following [3]):
  - Maximum energy: 1200 mAh; at the beginning of each simulation run, each node is assigned a random amount of energy between zero and this maximum.
  - Energy loss per second: 1.9×10^{-3} mAh.

[2] A similar approach to key exchange is used by Threema (https://threema.ch/en/), an existing non-opportunistic mobile messaging application. Security-conscious users may exchange keys in-person (scanning machine-readable QR codes), to enable later secure communication over untrusted networks.

[3] For *LocShare*, there are four one-week parts; we use each one-week segment with equal frequency. For *SASSY* and *Reality Mining*, following [20], [21] we select one-week intervals where there are sufficient numbers of nodes present for non-trivial routing to be possible.

- Energy per message sent/received[4]: 0.4 mAh.
- Nodes participate in the network until they run out of energy. They then recharge offline for 8 hours, during which they do not participate in the network, and return with full energy.
- Infinite buffers; no transmission loss.[5]

Following [20], [21], messages which arrive in zero-time (i.e., from a direct link between the original sender and final recipient) are excluded from analysis because, when sender and recipient are in proximity, there are presumably more efficient forms of communication than an opportunistic network. By excluding these transfers, we are able to focus on network performance in non-trivial opportunistic scenarios.

C. Flooding attack modes

Following [5], we pick one node from the trace in each run to act as the attacker; we do not add new attacker nodes because a model to generate synthetic node movement traces is beyond the scope of this paper. The attacker attempts to flood the network with attack messages during encounters with other nodes. It does not participate in relaying background traffic.

We simulate the following modes:

- *Baseline*: As a measurement of baseline behaviour of the network, no attack messages are generated.
- *Vulnerable*: Simulation of the default behaviour of SSNR, without any countermeasures to the attack as introduced in Section III-A. At each encounter, the attacker generates 100 spoofed messages.[6] As discussed when introducing the attack, message headers are spoofed to ensure that the message is (i) undeliverable (i.e., has no real final destination node), and (ii) eligible to be relayed by any node.
- *Resistant*: Simulation of a passive defence scheme, as introduced in Section III-B. The attacker must sign each attack message, so messages are only relayed via its genuine social contacts (friends). As for *Vulnerable*, the attacker sends 100 messages per encounter.
- *ResistantBlocks*: As for *Resistant*, but with an added active defence. Nodes locally maintain counts of messages they have received from each other node — with message origin verified since only signed messages from social contacts are accepted. Each node locally looks for any abnormal nodes, i.e., any node which has sent three standard deviations above the mean number of messages. If such a node is detected, then it is blocked at the detecting node; i.e., the node will discard further messages originating from this sender.

[4]We assume that the same amount of energy is used per message for each of the modes introduced in Section IV-C, i.e., that the energy cost of signing a message or verifying a message signature is small in comparison to the fixed radio energy usage for exchanging the message.

[5]As detailed in Section IV-D, we focus on measuring message loss caused by overloaded nodes which run out of energy. We only introduce this one source of message loss to avoid confounding the results.

[6]Some traces have artifacts, where a single logical encounter is stored as numerous, consecutive physical encounters. For example, a Bluetooth scan may detect the same node during consecutive scans. To avoid skewing results due to these artifacts, we limit the attacker to sending to encountered nodes no more than once every ten minutes.

D. Metrics

To evaluate the efficacy of the attack, we use three metrics:

1) *Proportion of the time that (non-attack) nodes spend offline recharging*. For example, if each node spends eight hours in every 24 hours recharging, then it is offline recharging for 33% of the time.
2) *Delivery ratio*. The proportion of (non-attack) messages which arrive at their intended destination.
3) *Delivery delay*. The delay between the first transmission of a message, and its first arrival at its intended final recipient node.

If the attack is successful in overloading network nodes, i.e., causing them to run out of energy and fail, then we would expect the nodes to spend a greater proportion of time offline recharging. We therefore use as a metric the proportion of the time that the non-attack nodes spend offline recharging. The remaining two metrics — delivery ratio and delivery delay — are widely used as indicators of overall network performance [16].

V. Results

Figures 1(a)–2(c) show our simulation results. Due to space constraints, we elide the plots for the *LocShare* dataset; trends for each metric were the same as for the *Reality Mining* dataset.

We consider two ways to measure the success of the attack: by examining the impact on individual nodes (with the metric of average proportion of time offline), and on the overall network performance (delivery ratio and delivery delay).

A. Impact of the attack

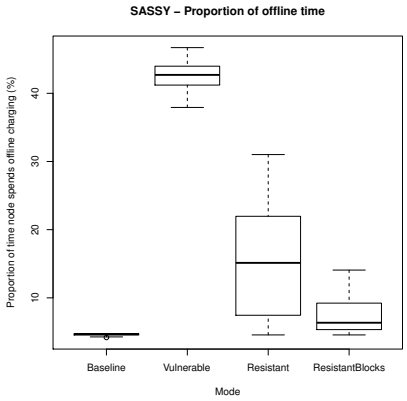
To determine the impact of the attack, we compare the metrics for each dataset in the *Vulnerable* mode, where the attack is performed, to the *Baseline* mode.

Figure 1(a) shows that there is a significant impact on nodes' proportion of time spent offline for the *SASSY* dataset. The attack has drained the nodes' energy, causing them to lose power. The median proportion of offline time is 42.7% for the *Vulnerable* mode, compared to the *Baseline* mode's 4.7%. For the sparsely-connected *Reality Mining* dataset, Figure 2(a) shows a more modest — but again significant — increase in node offline time, from 4.4% to 5.3%. The effect in the also-sparse *LocShare* dataset (plots elided due to space constraints) is similar (4.5% to 5.3%).

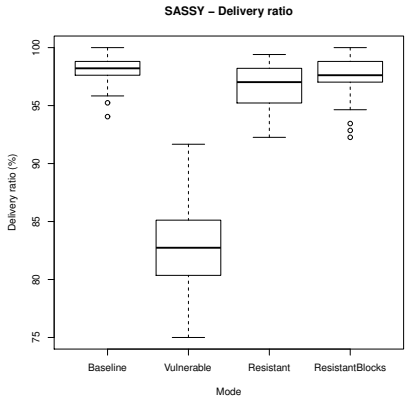
We have seen that individual nodes are affected. Is the network performance as a whole also impacted?

For the *SASSY* dataset, Figures 1(b)–1(c) show that there is a significant impact on the overall delivery ratio and delivery delay. The median delivery ratio falls from 98.2% to 82.7%, while delivery delay doubles from 3.3 hours to 6.5 hours. For the *Reality Mining* dataset, however, Figures 2(b)–2(c) do not show a significant difference in network performance. The *LocShare* results are similar. We believe this is a consequence of the datasets' sparsity: the absolute delivery ratios are so low and variable that any impact is lost in the noise.

Summarising, the attack significantly increases offline times for individual network nodes. For the dense *SASSY*



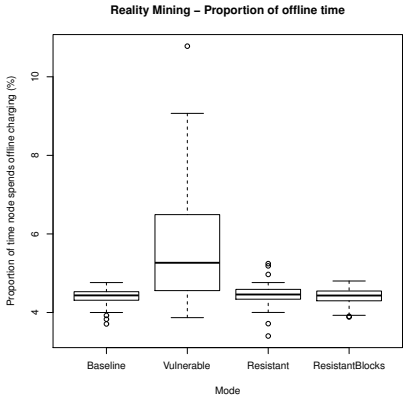
(a) The flooding attack (Vulnerable mode) overloads nodes, so they spend more time recharging compared to the baseline (median: 42.7% vs 4.7%). The passive (Resistant) and active (ResistantBlocks) defences mitigate this: median offline proportion falls to 15.1% and 6.3% respectively.



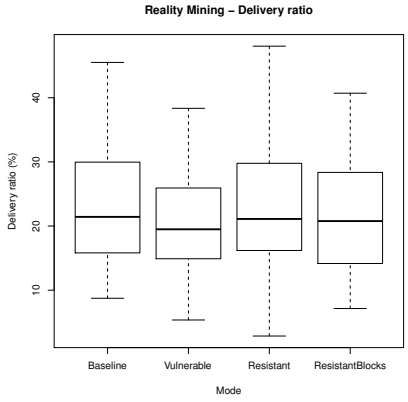
(b) Overall network performance, as measured by delivery ratio, falls during the attack (from 98.2% to 82.7%). The passive and active defences mitigate the attack.

(c) The attack worsens performance, increasing the delivery delay (3.3 hours to 6.5 hours). The passive and active defences mitigate this impact.

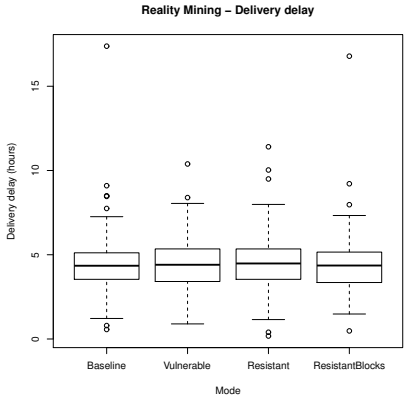
Fig. 1. *SASSY* dataset.



(a) In this sparse dataset, the attack causes a more modest — but still significant — increase in offline time (4.4% to 5.3%). The defences mitigate this.



(b) There is no significant difference in delivery ratios across the different modes.



(c) There is no significant difference in delivery delays across the modes.

Fig. 2. *Reality Mining* dataset.

dataset, the overall network performance impact is also directly measurable. This demonstrates the efficacy of the attack.

We also note that impacting the energy of individual nodes (i.e., the mobile devices carried by network users) may discourage users' participation in the network. By theories such as Metcalfe's Law and Reed's Law [24], this may further reduce the value of the network for other nodes.

B. Efficacy of the defence

The *Resistant* mode implements the passive defence, and *ResistantBlocks* the active defence. By comparison to the *Vulnerable* mode, we can determine their efficacy.

From Figure 1(a), we can see that the defence effectively mitigates the effect of the attack on nodes' offline times for the *SASSY* dataset. Compared to a median of 42.7% time offline for the *Vulnerable* mode, this falls to 15.1% with the *Resistant* mode, and further to 6.3% with the *ResistantBlocks* mode — almost to the *Baseline* level. A similar trend holds for the *Reality Mining* dataset, shown in Figure 2(a), and for the *LocShare* dataset. This is less pronounced, because the attack's impact was more moderate for these sparse datasets.

The network performance impact is also mitigated. For the *SASSY* dataset, Figures 1(b)–1(c) show an increased delivery ratio, and corresponding decreased delivery delay, using the *Resistant* mode (82.7% to 97.0%, and 6.5 hours to 4.4 hours). With the active defence, *ResistantBlocks*, the performance is further improved, to near-*Baseline* levels (97.6% delivery ratio, and 3.5 hours delivery delay). For the *Reality Mining* and *LocShare* datasets, the attack had less effect on network performance, but the defence still does not worsen performance.

VI. Conclusions and future work

This paper has described a simple flooding attack against opportunistic networks. We have simulated the attack using real-world datasets and shown it to be capable of disrupting an opportunistic network, both at the node level by taking nodes offline, and at the global network level by lowering delivery ratio. We have proposed a social-network-based mitigation strategy which is lightweight and appears effective.

Our results indicate that while it is possible to mitigate a flooding attack using our modified routing protocol, this does not come for free. We have introduced assumptions, outlined in Section III, which may impede opportunistic network use.

Specifically, we assume the existence of some mechanism for out-of-band key distribution amongst socially-connected nodes. On the one hand, this may seem a reasonable assumption. If a node is "friends" with another node, then they may well have had sufficient opportunity to exchange keys prior to encountering each other in an opportunistic network scenario, for instance via meeting physically or through an infrastructure network. On the other hand, by requiring keys to communicate, we may be impeding potential uses of opportunistic communication. For instance, epidemic routing applications such as emergency broadcast or content distribution, where nodes send messages to any available node, are no longer possible. If epidemic routing is allowed, then a recipient node may no longer be able to verify a sender's key, which means that malicious nodes could generate throwaway public-private key pairs for forged nodes and so conduct the flooding attack.

A future measurement-based study may reveal whether the public key cryptography assumption would hold in a real deployment. Alternatively, it may be possible to relax the requirement. Shikfa *et al.* propose the use of identity-based cryptography as an alternative to sharing public keys in opportunistic networks [25]. This introduces a new restriction, however, now requiring a globally-trusted third party.

Another avenue for future work might be to explore whether it is indeed possible to enable epidemic routing while maintaining public key cryptography, for instance by delegating trust to "friends of friends".

Our results also indicate that the flooding attack is more effective in dense than in sparse datasets. We used a variety of datasets as we do not know what an actual large-scale opportunistic network would look like. One can imagine a more sophisticated attack that dynamically changes its parameters depending on mobility or the density of node encounters; sophisticated attacks may require more sophisticated defences.

References

[1] A. Belenky and N. Ansari. On IP traceback. *IEEE Commun. Mag.*, 41(7):142–153, July 2003. doi:10.1109/MCOM.2003.1215651.

[2] F. Ben Abdesslem, T. Henderson, and I. Parris. CRAWDAD data set st_andrews/locshare (v. 2011-10-12). Downloaded from http://crawdad.org/st_andrews/locshare, Oct. 2011.

[3] G. Bigwood and T. Henderson. Bootstrapping opportunistic networks using social roles. In *Proc. AOC*, June 2011. doi:10.1109/WoWMoM.2011.5986139.

[4] G. Bigwood, D. Rehunathan, M. Bateman, T. Henderson, and S. Bhatti. CRAWDAD data set st_andrews/sassy (v. 2011-06-03). Downloaded from http://crawdad.org/st_andrews/sassy, June 2011.

[5] J. Burgess, G. D. Bissias, M. D. Corner, and B. N. Levine. Surviving attacks on disruption-tolerant networks without authentication. In *Proc. MobiHoc*, Sept. 2007. doi:10.1145/1288107.1288116.

[6] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the detection of fake accounts in large scale social online services. In *Proc. NSDI*, Apr. 2012.

[7] L.-J. Chen, C.-L. Chiou, and Y.-C. Chen. An evaluation of routing reliability in non-collaborative opportunistic networks. In *Proc. AINA*, May 2009. doi:10.1109/AINA.2009.54.

[8] F. C. Choo, M. C. Chan, and E.-C. Chang. Robustness of DTN against routing attacks. In *Proc. COMSNETS*, Jan. 2010. doi:10.1109/COMSNETS.2010.5432014.

[9] J. R. Douceur. The Sybil attack. In *Proc. IPTPS*, 2002. doi:10.1007/3-540-45748-8_24.

[10] N. Eagle and A. S. Pentland. CRAWDAD data set mit/reality (v. 2005-07-01). Downloaded from http://crawdad.org/mit/reality, July 2005.

[11] K. El Defrawy, J. Solis, and G. Tsudik. Leveraging social contacts for message confidentiality in delay tolerant networks. In *Proc. COMPSAC*, July 2009. doi:10.1109/COMPSAC.2009.43.

[12] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazières, and H. Yu. RE: reliable email. In *Proc. NSDI*, 2006. USENIX Association.

[13] Y. Guo, S. Gordon, and S. Perreau. A flow based detection mechanism against flooding attacks in mobile ad hoc networks. In *Proc. IEEE WCNC*, 2007. doi:10.1109/WCNC.2007.574.

[14] S. Hameed, X. Fu, P. Hui, and N. Sastry. LENS: Leveraging social networking and trust to prevent spam transmission. In *Proc. ICNP*, Oct. 2011. doi:10.1109/icnp.2011.6089044.

[15] T. Hossmann, P. Carta, D. Schatzmann, F. Legendre, P. Gunningberg, and C. Rohner. Twitter in disaster mode: Security architecture. In *Proc. SWID*, 2011. doi:10.1145/2079360.2079367.

[16] P. Hui, J. Crowcroft, and E. Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proc. MobiHoc*, May 2008. doi:10.1145/1374618.1374652.

[17] Y. Kim and A. Helmy. CATCH: A protocol framework for cross-layer attacker traceback in mobile multi-hop networks. *Ad Hoc Netw.*, 8(2):193–213, Mar. 2010. doi:10.1016/j.adhoc.2009.07.002.

[18] F. C. Lee, W. Goh, and C. K. Yeo. A queuing mechanism to alleviate flooding attacks in probabilistic delay tolerant networks. In *Proc. AICT*, May 2010. doi:10.1109/AICT.2010.78.

[19] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM Comput. Commun. Review*, 34(2):39–53, Apr. 2004. doi:10.1145/997150.997156.

[20] I. Parris, F. Ben Abdesslem, and T. Henderson. Facebook or fakebook? The effects of simulated mobile applications on simulated mobile networks. *Ad Hoc Netw.*, 12:35–49, Jan. 2014. doi:10.1016/j.adhoc.2012.05.008.

[21] I. Parris and T. Henderson. Privacy-enhanced social-network routing. *Comput. Commun.*, 35(1):62–74, Jan. 2012. doi:10.1016/j.comcom.2010.11.003.

[22] K. Pelechrinis, M. Iliofotou, and S. V. Krishnamurthy. Denial of service attacks in wireless networks: The case of jammers. *IEEE Commun. Surveys Tuts.*, 13(2):245–257, 2011. doi:10.1109/surv.2011.041110.00022.

[23] C. Piro, C. Shields, and B. N. Levine. Detecting the Sybil attack in mobile ad hoc networks. In *Proc. Securecomm*, Aug. 2006. doi:10.1109/SECCOMW.2006.359558.

[24] D. P. Reed. That sneaky exponential - beyond Metcalfe's law to the power of community building. *Context Magazine*, Spring 1999.

[25] A. Shikfa, M. Önen, and R. Molva. Privacy and confidentiality in context-based and epidemic forwarding. *Comput. Commun.*, 33(13):1493–1504, Apr. 2010. doi:10.1016/j.comcom.2010.04.035.

[26] H.-X. Tan and W. K. G. Seah. Framework for statistical filtering against DDoS attacks in MANETs. In *Proc. ICESS*, Dec. 2005. doi:10.1109/ICESS.2005.57.

[27] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre. Preventing spam in opportunistic networks. *Comput. Commun.*, Jan. 2014. doi:10.1016/j.comcom.2013.12.003.

[28] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against Sybil attacks via social networks. In *Proc. SIGCOMM*, 2006. doi:10.1145/1159913.1159945.