

CapMan: Capability-based Defense against Multi-Path Denial of Service (DoS) Attacks in MANET

Quan Jia

Department of Computer Science
George Mason University
Fairfax, Virginia 22030
qjia@gmu.edu

Kun Sun

Center for Secure Information Systems
George Mason University
Fairfax, Virginia 22030
ksun3@gmu.edu

Angelos Stavrou

Department of Computer Science
George Mason University
Fairfax, Virginia 22030
astavrou@gmu.edu

Abstract—This paper presents a capability-based security mechanism called *CapMan*. Our approach is designed to prevent Denial-of-Service (DoS) attacks on wireless communications, particularly against multi-path communication in Mobile Ad-hoc Networks (MANETs). *CapMan* offers a mechanism for a per flow, distributed bandwidth control by all the participating nodes along multiple communication paths. By exchanging summary capability messages, each node can maintain a global view of the overall throughput of flows in the network, and then dynamically adjust local constraints to prevent potential DoS attacks against a specific node or the network. Our approach is capable of scalably curtailing sophisticated DoS attacks that target multi-path routing protocols, even in the case that both the initiator and the responder of a network flow are malicious insiders and collude to deprive the network of valuable resources. We provide a theoretical analysis of our algorithms and also evaluate the protection and overhead of our prototype using AOMDV for routing.

Keywords: Capability, DoS, MANET, Multi-path.

I. INTRODUCTION

Mobile Ad-hoc Networks (MANETs) are increasingly being deployed in both military and civil operations for emergency rescue and disaster relief. Although they offer abundant deployment flexibility and support for mobility, MANETs are susceptible to a wider range of Denial of Service (DoS) attacks compared to the wired networks. An external adversary can jam the physical radio signals using wireless signal jammer, redirect and disrupt existing communication paths, or saturate a normal node with a large number of fake communication requests. It is even easier for malicious insiders to successfully achieve DoS attacks that deplete the constrained power energy of the mobile nodes by flooding the network with garbage packets.

In the past, researchers have proposed some potential solutions to prevent or mitigate flooding-oriented DoS attacks [1], [2], [3], [4], [5], [6]. Among them, capability-based mechanisms [4], [5] were raised to control resource usage by assigning a capability for each node to enforce. A capability is a token issued by the responder of any transport layer flow to its initiator, imposing a limit on the amount of traffic that can

be sent through the flow within a certain period of time. Unfortunately, existing capability-based solutions assume that a DoS attack only affects static routes between two end-nodes. They lack support for node mobility, frequent route changes, and communications over multiple paths. Therefore, even though they can ensure that the bandwidth consumption on each path may not go beyond the capability, the aggregated multi-path throughput can still be significantly higher than that. Moreover, they cannot protect the network against adversaries that create multiple flows over multiple links. In terms of quality of service (QoS) and resilience, multi-path routing provides better load balancing, improved fault tolerance, and more efficient use of bandwidth. However, it also generates a new threat vector multiplying the potential impact of DoS attacks.

In this paper, we introduce and analyze a novel capability-based secure communication mechanism called *CapMan*. Our aim is to defend MANETs against DoS attacks and especially those that take advantage of the multi-path nature of wireless communications. *CapMan* consists of two main components: the *capability distribution* and the *capability enforcement*. The capability distribution protocol empowers the responder of a traffic flow to issue and distribute a capability to all the nodes along the routing path. When a responder receives a connection request from an initiator, it sends a capability packet to the initiator as a notification of the acceptance of an end-to-end flow and the discovery of a new routing path. The capability is not only used as a ticket by the initiator to send data packets, but also saved by all intermediate nodes to restrict the number of packets they will forward for the flow. In addition, the capability enforcement mechanism implements the capability constraint on a per-hop basis across multiple routing paths. We assume multi-path routing between end nodes and that the routing paths do change dynamically. To account for that, all nodes periodically exchange bandwidth consumption reports. This enables each node to maintain a global view of the per flow throughput and capability between any pair of initiator and responder. Thus, our approach can effectively identify and mitigate sophisticated DoS attacks that target multi-path routing protocols, even if both the initiator and the responder

are colluding malicious insiders. We have implemented the CapMan mechanism in NS2 [7], using AOMDV [8] as the underlying multi-path routing protocol. Our results show that our mechanism effectively contain DoS attacks and reduce their impacts on existing normal communication flows.

II. RELATED WORK

Due to their inherent mobility and dynamic topology, MANETs are susceptible to DoS attacks [9]. Although more debilitating for MANETs, network DoS attacks have not received enough attention compared to wired networks. Indeed, numerous solutions have been proposed to protect Internet from DoS attacks, some of which have also shed lights on solving the problem in ad-hoc network.

Traceback mechanism [2], [10], [11], one of the early methods proposed to defend DoS attacks in internet, was used to detect the sources of DoS attacks. Unfortunately, traceback schemes cannot prevent attacks from happening nor be directly applied in a MANET environment. Therefore, many have suggested combining Traceback with Pushback [12], [6] to defeat DoS flooding attacks. The Pushback methods focused on rate-limiting inordinate senders with aggregate-based congestion control. However, it is often difficult to distinguish normal senders from malicious ones. To make matters worse, the power of Pushback will be severely impaired by mobility because in MANETs there are no fixed upstream routers but rather moving ordinary nodes who carry the traffic through.

Another avenue of defense are filtering based schemes [1], [13], [14], [15], [16], [17], [18]. Such solutions rely on sophisticated filters being installed on routers to block hosts from sending excessive traffic to a victim destination. Communication between any source and destination is allowed by default and is shut-off only when it is identified as an attack. Although specially designed identities (e.g. signatures) can be adopted to mark each host, IP addresses are used more often due to the scale of Internet. Consequently, several proposals ([13], [14], [18]) were raised to tackle source address spoofing. In contrast to these mechanisms that require high degree of deployment, other researchers suggested overlay filtering [15], [16], [17], [19] that can be deployed incrementally. For example, CenterTrack [17] employs routers with input debugging utilities in collaboration with borders routers to form a network overlay to investigate the sources of DDoS attacks; SOS [16] and Mayday [15] use a large overlay network to perform packet authentication. Deploying filters is easier in MANETs because they are small in scale and usually enforce strict group management policies.

Contrary to filtering, capability based mechanisms [20], [21], [22], [23] provide “control over resource usage to the owner of the limited resource” [20] by letting the destination host decide the capability to assign and the intermediate nodes to police. Alicherry et al. [4], [24], [5], [25] recently introduced the idea to address DoS attacks in MANETs. The proposed Deny-by-Default mechanism aims to protect both network and end hosts from DoS flooding attacks with security enhanced architecture. Despite its merits, the solution was

built upon an implicit assumption that all packets of any communication are forwarded through a single static route on each direction. This is unrealistic in most cases considering the dynamic nature of MANET topologies. Our work extends this approach to a more realistic scenario where packets of the same flow can traverse multiple paths.

There has been a long term debate about the superiority between the filtering-based and the capability-based methods [1], [26] and no consensus has been reached so far. Actually, both solutions have their advantages with the key difference being: filtering based methods are allow-by-default (reactive) while capability based approaches are deny-by-default (proactive). In this paper, We focus on using the latter to address DoS flooding attacks in MANET environment.

III. THREAT MODEL

Due to MANET’s dynamic nature and inherent mobility, there is no intranet or other network separation as in wired networks. Therefore, attackers can easily exhaust one victim’s constrained computation, communication, and energy resources. Although credential-based mechanisms can be employed to form independent groups with encrypted communication channels, malicious insiders can still flood other normal nodes with a large number of fake packets. Because the insiders can use the keying material to generate valid message authentication codes for fake packets, normal nodes will rebroadcast these packets to the entire network. In this paper, we aim at thwarting the flooding DoS attacks by insiders in MANETs.

Multi-path routing [27], [8] is desirable because it offers load balancing, fault tolerance and higher aggregate throughput in MANETs. Unfortunately, it also exacerbates the possibility and intensity for flooding DoS attacks. In realistic environments, any node in MANETs can be malicious and initiate flooding DoS attacks against other nodes. Based on the target of the DoS attacks, we classify the flooding DoS attacks into two categories.

- *Node-targeted*: It aims to either saturate the bandwidth or exhaust the CPU and power energy of the initiator or the responder of a flow. Such attacks can be carried out by individual malicious node or multiple colluding nodes. A malicious initiator can exploit multi-path routing to increase its traffic load, while malicious intermediate nodes can further augment the attack through packet replay.
- *Link-targeted*: Another form of DoS attacks focuses on the bottleneck links in the network. By sending excessive packets, malicious upstream nodes can congest the bottleneck link at downstream. This causes significant packet loss that can have debilitating results for other flows that share the same link. Although active queue management (AQM) schemes [28] are available to provide fairness among flows, they are not sufficient for characterizing or differentiating excessive or malicious flows and thus, can be abused.

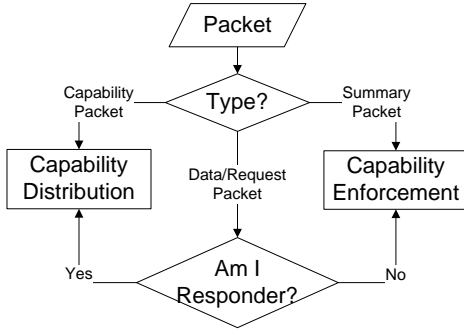


Fig. 1: Overview of the flow control for CapMan system.

IV. CAPABILITY SYSTEM DESIGN AND IMPLEMENTATION

A high level control flow for CapMan is depicted by Figure 1. CapMan is comprised of two interdependent components, the *capability distribution* and the *capability enforcement*. With the term *capability distribution*, we refer to the process of capability issuance by the responder to the initiator, as well as accurately propagating the capability among all employed routing paths. Meanwhile, *capability enforcement* involves all nodes along different routing paths, to collaborate in a distributed manner on enforcing the overall capability requirement. Each node in the network runs an identical copy of CapMan system. Both *capability distribution* and *capability enforcement* target at a uni-directional, transport layer flow stemming from an initiator to a responder. A bi-directional TCP connection is considered as one flow on each direction from the capability perspective. There are two cases when the capability distribution will be executed: 1) If a responder receives a connection request or data packet, it will check the recorded route in the packet header and send a capability packet back to the initiator if necessary. 2) Upon receiving a capability packet, all nodes will update their local values and forward the packet until reaching the initiator. In all other cases, the capability enforcement component will be activated. Especially when a node receives a capability summary packet that reports the sender’s local capability status, the receiving node will update its local capability setting accordingly to achieve cross-path collaboration.

A. Capability Distribution

For any flow to be established, the responder needs to reply to the initiator’s connection request with a capability packet that contains a capability specifying the maximum data rate for the flow. The packet is sent along the reverse route back to the initiator. After receiving the packet, intermediate nodes along the route extract the capability for the flow and save it into their local capability table. This cached capability is going to be the passport for future packets of the flow. In case no capability is issued or the cached capability expires, a minimal rate limit would be enforced on the flow to only allow connection requests to go through.

The *capability distribution* procedure starts when the initiator sends a connection request packet to the responder. For

TCP traffic, this is a SYN packet; for UDP traffic, this is the very first data packet. When the responder receives the connection request packet, it decides the capability for the flow under the consideration. The decision depends on available bandwidth, the application associated with this connection, and the capabilities already assigned to the same initiator. Once a capability is decided for a flow, the responder creates a capability packet and sent it back to the initiator along the reverse route of the request packet. Each intermediate node along the route compares the capability with the local entries in its capability table. Should the capability be new or changed, the node will update its capability table accordingly. Assuming the route stays unchanged within a single roundtrip time, the capability packet should eventually reach the flow initiator. After that, the initiator begins to send data packets to the responder using the capability. Each data packet includes the *flow identifier (FID)*, a combination of the source and destination IP addresses, as well as a unique sequence number issued by the responder) instead of the entire capability token. When a new route is used to forward packets for an existing flow, the responder will construct a new capability packet with the new path encoded (but the same capability value) in the header and send it back to the initiator reversing the route.

B. Capability Enforcement

The process of enforcing the advertised capabilities is a combination of local policing and flow-wide message exchange on each node. Local enforcement is conducted continuously by per path leaky bucket. Cross-path message exchange occurs periodically using a predefined time window with certain degree of freedom. The purpose of undertaking such message exchange is to make intermediate nodes aware of the flow-wide throughput and thus decide their bucket leaking rate. For that sake, a *path recording header (PRH)* is added to regular data packets to record the routing path through which the packets are delivered. It is incrementally filled by each intermediate node using the signature aggregation technique described in [29].

Local Capability Policing: For every data packet, each node inspects its header to extract the flow and path information. If a matched capability entry is found, the data packet is inserted at the end of the corresponding leaky bucket. If the bucket overflows the packet is dropped. However, If no capability is available for this flow, the packet will be placed in the bucket for anonymous traffic with a minimal capability (which can be zero). Packets in all buckets are leaked at different rates and sent to the next hop. The per bucket leaking rate is updated when summary messages are received from other nodes.

After forwarding a data packet, the size of the packet is added to the per path traffic counter which is used to compute a throughput at the end of a time window. The updated result is then used for constructing an aggregate summary packet by the node. The size of throughput calculation window should approximate a predefined value set by the network administration. However, certain degree of randomness should be added to avoid global synchronization. Each traffic counter

is reset at the start of the next window and summary packets are transmitted by broadcast.

Determination of Leak Rate: Upon receiving a summary packet, a node breaks it down and inspects every $\langle PRH, \text{Throughput} \rangle$ tuple inside. If a PRH is found to represent a path going through this node, the tuple will be discarded. Otherwise, it will be used for updating the local summary table. Stale summary packets (whose sequence number is smaller than or equal to a matched table entry) will always get dropped.

Path throughput for any flow is computed once per time window and the result is broadcasted via a summary packet. However, a periodical throughput check on all nodes is not enough to guarantee that the capability is abided all the time, since well tuned bursty traffic can potentially bypass such intermittent inspection. Therefore, leaky buckets are installed on all nodes to cope with attacks exploiting bursty traffic and to provide enhanced quality of service (QoS). Each bucket is associated with one route of a flow. Conceptually, the property of rate limiting by leaky bucket is consistent with our goal of enforcing capability. In addition, it can effectively curb the degree of burstiness. Therefore, the combination of leaky bucket and regular summary exchange ensures that the capability is enforced regardless of traffic patterns.

Each update in the summary table and capability table triggers a recalculation of the leaking rate for all leaky bucket pertaining to the same flow. For a flow F_K , we first calculate the number of routing paths that are used for packet forwarding. This is done by traversing the summary entries under F_K and find all unique $PRHs$. Also, we compute the throughput of each route for the last time window. By aggregating per route throughput for a flow F_K , we can get the overall throughput from all routes that are employed by the flow. We then compare this throughput value with the assigned capability. If the flow throughput exceeds the capability, the leaking rate should be dropped. Otherwise, if the capability is under utilized, the throughput can be increased accordingly.

V. SYSTEM EVALUATION

A. Security Analysis

Here, we analyze how effective CapMan is in identifying and limiting the effects of bandwidth DoS attacks launched by malicious insiders. In particular, we divide our discussion in accordance with the scenarios described in Section III.

Targeting the end nodes: DoS attacks against the responder can be launched by an individual adversary acting as the initiator, one or more adversaries as intermediate nodes, or a combination of the two.

The DoS attacks from individual malicious initiator in the network can be easily identified and stopped on the first hop of each route. Indeed, when excessive amount of traffic is sent by the initiator, the first hop along any path will detect that the aggregated traffic from the initiator is greater than the assigned capability. Consequently, it can seize forwarding packets for the initiator until the overall throughput falls below the capability. Similarly, if an adversary serves as an

intermediate node and attempts to send superfluous traffic via an existing flow by forging or replaying packets, the attack can be blocked at the next hop neighbor along each path. One exception occurs when the attacker is in the immediate neighborhood of the responder. In that case, the responder will receive and drop the fake packets.

Targeting the intermediate nodes: DoS attacks can be launched by pairs of malicious collaborating initiators and responders with the aim to deplete constrained network, CPU or energy resources from intermediate nodes along multiple paths. On one extreme, the responder of one flow can deliberately assign a high packet rate capability to the flow to consume as much bandwidth as possible. On another, the two colluding nodes can establish large numbers of smaller capabilities flows. In both cases, legitimate nodes will be deprived of valuable resources. As we have mentioned, besides communication bandwidth, CPU and energy are scarce in battery operated environments.

To mitigate the impact of such attacks, one potential solution is to correlate packet dropping probability with the relative value of capabilities. In the case of one large capability, on a congested node who forwards packets for multiple flows, the dropping probability derived by an existing queue management scheme can be further increased for flows with larger capabilities or lowered otherwise. In the case of plenty small capabilities, they can also be aggregated against a particular initiator or responder to make packet dropping decisions on a larger granularity.

B. Performance Study

We implement a prototype of CapMan in the NS2 simulator and evaluate its efficacy and performance through various simulation scenarios. For all simulations, we use IEEE 802.11 as the MAC layer protocol and AOMDV [8] as the multi-path routing protocol. The original AOMDV design focuses more on providing fault tolerance rather than load balance. Therefore, one optimal path is always used until it breaks. Alternative routes are cached and will be active only when a link failure occurs. To maximize the attacker's throughput by making full use of every routing path, we slightly modified the NS2 implementation of AOMDV to achieve round-robin routing among all available paths.

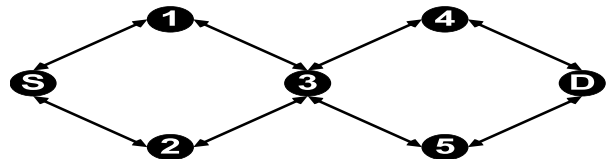


Fig. 2: Topology for multi-path routing with one flow

First of all, we want to verify the effectiveness of CapMan on preventing DoS attacks using the network topology described in Figure 2. There is one source node (S), one destination node (D), and five intermediate nodes. The packet flow between S and D is routed via four paths $1 \rightarrow 3 \rightarrow 4$, $1 \rightarrow 3 \rightarrow 5$, $2 \rightarrow 3 \rightarrow 4$, $2 \rightarrow 3 \rightarrow 5$. D issues a capability that specifies a maximum data rate of $100Kbps$ to S.

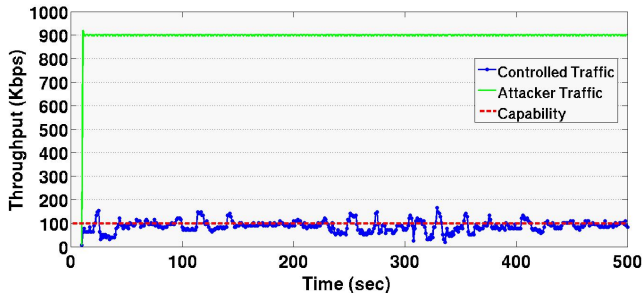


Fig. 3: DoS attack with CBR traffic

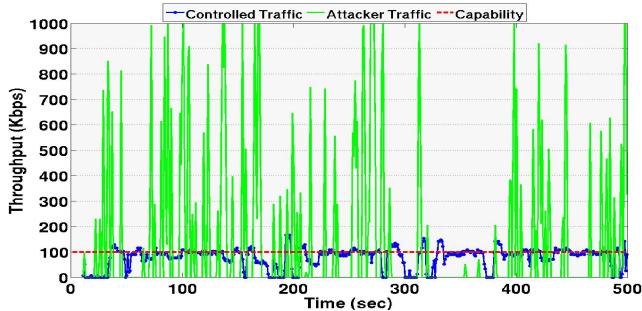


Fig. 4: DoS attack with bursty traffic

We initially assume the source node is the only malicious node. Figure 3 shows that our mechanism is able to defeat the attack by pulling down the throughput within the capability $100Kbps$, when S attempts to attack D using Constant Bit Rate (CBR) traffic at $900Kbps$, a rate nine times of the assigned capability. Figure 4 shows that CapMan can constrain the flooding DoS attacks using bursty traffic too. To generate a bursty traffic, an exponential on/off traffic with a peak rate of $1000Kbps$ is attached to the malicious S . With our capability mechanism enabled, the maximum throughput is reduced to around $150Kbps$. Thus, we can see that CapMan can prevent a single malicious source from launching flooding DoS attacks.

Now we assume both the joint node 3 and the source node S are malicious and collude on launching DoS attacks. S still floods the same amount of DoS traffic as in the previous CBR scenario. Node 3 enhances the attack by generating multiple (in the experiment, three) extra copies of each packet it receives from S and forwarding all of them to D . Moreover, node 3 acts as a black hole for capability enforcement in that it does not produce or forward any summary packet. The simulation result is shown in Figure 5. Apparently, this colluding attack cannot raise throughput substantially compared to single malicious source node case. The reason is that each normal intermediate node enforces capability on its upstream neighbors and won't forward the flooding packets to its next hop neighbors, so the packet throughput on the destination node is constrained by the capability.

Next, we show the effectiveness of CapMan on reducing the impact of DoS attacks against existing normal communication flows. Figure 6 shows the network topology containing two flows from two separate source nodes $S1$ and $S2$ towards the same destination D . Here, $S1$ is malicious and the flow between $S1$ to D is routed through two routes $S1 \rightarrow 1 \rightarrow 2 \rightarrow D$

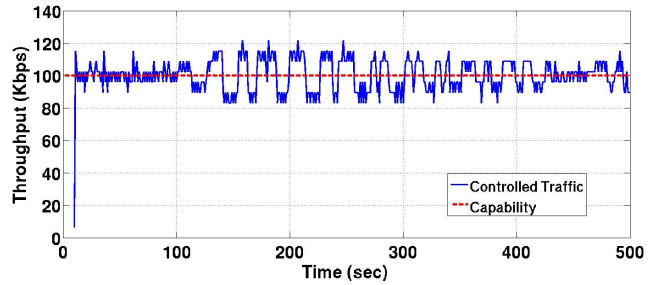


Fig. 5: Attack involves malicious intermediate node

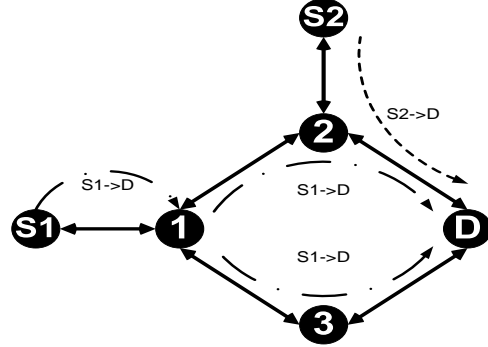


Fig. 6: Topology for studying capability with two flows

and $S1 \rightarrow 1 \rightarrow 3 \rightarrow D$. Meanwhile, $S2$ is benign and it uses only on route $S2 \rightarrow 2 \rightarrow D$ to send packets to D .

TABLE I: Defend against DoS attacks on network resources

Settings	$S1 \rightarrow D$		$S2 \rightarrow D$		
	Throughput	Latency	Throughput	PDR	Latency
Only $S2 \rightarrow D$	-	-	149.1Kbps	1.00	0.16s
$S1 \rightarrow D + S2 \rightarrow D$	384.7Kbps	18.25s	87.8Kbps	0.589	12.84s
$Cap_1 = Cap_2 = 150Kbps$	147.0Kbps	17.43s	119.1Kbps	0.799	8.40s
$Cap_1 = 120Kbps, Cap_2 = 150Kbps$	118.2Kbps	16.31s	142.2Kbps	0.953	2.82s

The experimental results are shown in Table I. $S1 \rightarrow D$ is the malicious flow while $S2 \rightarrow D$ is the benign flow. We measure the throughput and latency for both flows, and we also record the packet delivery ratio (PDR) for the benign flow. Cap_1 is the capability for flow $S1 \rightarrow D$ and Cap_2 is for $S2 \rightarrow D$. The first row of the table discloses our throughput settings on the two flows. The following rows reveal their statistics under different scenarios.

First, we ran only the benign flow without capability enforcement. Apparently, all packets were delivered with minimal latency. Next, we released the malicious traffic to compete for bandwidth, still without enabling the capability system. As expected, network congestion occurred at node 2, because we observed substantially increased transmission latency as well as drastically dropped throughput PDR on the benign flow. The malicious flow took over a major portion of the available bandwidth, resulting in a degraded QoS for normal nodes. For the next step, a $150Kbps$ capability was introduced on both flows to curtail the DoS traffic. As anticipated, the

bandwidth occupied by the malicious flow fell by more than 60% while the benign traffic throughput was recovered by 21%. The benign flow was further restored when we imposed an even lower capability on the attacker.

VI. CONCLUSION

CapMan is designed to mitigate Denial-of-Service (DoS) attacks on MANETs by regulating end-to-end traffic communicated over multiple paths. CapMan empowers individual nodes to maintain global flow state which in turn enables them to both setup and enforce bandwidth limits in a distributed fashion. Moreover, we consider malicious insiders that can subvert any participating entity.

We evaluate our approach through theoretical analysis and extensive NS2 simulations using AOMDV as the underlying routing layer. Both our experimental results and theoretical models indicate that CapMan is effective in rate-limiting the throughput of multi-path flows. Moreover, it is capable of protecting both the network and the end nodes from sophisticated DoS attacks.

REFERENCES

- [1] X. Liu, X. Yang, and Y. Lu, "To filter or to authorize: network-layer dos defense against multimillion-node botnets," in *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*. New York, NY, USA: ACM, 2008, pp. 195–206.
- [2] S. Bellovin, M. Leech, and T. Taylor, "Internet draft: Icmp traceback messages," <http://tools.ietf.org/html/draft-ietf-itrace-04>, 2003.
- [3] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," *Wirel. Netw.*, vol. 11, no. 1-2, pp. 21–38, 2005.
- [4] M. Alicherry, A. D. Keromytis, and A. Stavrou, "Deny-by-default distributed security policy enforcement in mobile ad hoc networks," in *Proceedings of the 5th International Conference on Security and Privacy in Communication Networks*, September 2009.
- [5] M. Alicherry and A. D. Keromytis, "Diploma: Distributed policy enforcement architecture for manets," in *Proceedings of the 4th International Conference on Network and System Security (NSS)*, September 2010, pp. 89–98.
- [6] J. Ioannidis and S. M. Bellovin, "Implementing pushback: Router-based defense against ddos attacks," in *Proceedings of Network and Distributed System Security Symposium*, 2002.
- [7] "The network simulator ns-2," <http://www.isi.edu/nsnam/ns/>.
- [8] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2001, pp. 14–23.
- [9] B. Wu, J. Chen, J. Wu, and M. Cardei, *A Survey on Attacks and Countermeasures in MANETs*. Springer, 2006, ch. 12.
- [10] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, "Practical network support for ip traceback," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 4, pp. 295–306, 2000.
- [11] A. C. Snoeren, "Hash-based ip traceback," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 3–14, 2001.
- [12] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, "Controlling high bandwidth aggregates in the network," *ACM Computer Communication Review*, vol. 32, pp. 62–73, 2002.
- [13] P. Ferguson and D. Senie, "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," RFC 2827 (Best Current Practice), Internet Engineering Task Force, May 2000, updated by RFC 3704. [Online]. Available: <http://www.ietf.org/rfc/rfc2827.txt>
- [14] K. Park and H. Lee, "On the effectiveness of route-based packet filtering for distributed dos attack prevention in power-law internets," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 15–26, 2001.
- [15] D. G. Andersen, "Mayday: distributed filtering for internet services," in *USITS'03: Proceedings of the 4th conference on USENIX Symposium on Internet Technologies and Systems*. Berkeley, CA, USA: USENIX Association, 2003, pp. 3–3.
- [16] A. D. Keromytis, V. Misra, and D. Rubenstein, "Sos: Secure overlay services," in *Proceedings of ACM SIGCOMM*, 2002, pp. 61–72.
- [17] R. Stone, "Centertrack: an ip overlay network for tracking dos floods," in *SSYM'00: Proceedings of the 9th conference on USENIX Security Symposium*. Berkeley, CA, USA: USENIX Association, 2000, pp. 15–15.
- [18] A. Yaar, A. Perrig, and D. Song, "Pi: A path identification mechanism to defend against ddos attacks," in *SP '03: Proceedings of the 2003 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2003, p. 93.
- [19] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," *IEEE/ACM Trans. Netw.*, vol. 12, no. 2, pp. 205–218, 2004.
- [20] T. Anderson, T. Roscoe, and D. Wetherall, "Preventing internet denial-of-service with capabilities," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 39–44, 2004.
- [21] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. Maggs, and Y.-C. Hu, "Portcullis: Protecting connection setup from denial-of-capability attacks," in *Proceedings of the ACM SIGCOMM*, August 2007.
- [22] X. Yang, D. Wetherall, and T. Anderson, "Tva: a dos-limiting network architecture," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1267–1280, 2008.
- [23] A. Yaar, A. Perrig, and D. Song, "Siff: A stateless internet flow filter to mitigate ddos flooding attacks," in *IEEE Symposium on Security and Privacy*, 2004, pp. 130–143.
- [24] M. Alicherry, A. D. Keromytis, and A. Stavrou, "Evaluating a collaborative defense architecture for manets," in *IMSA'09: Proceedings of the 3rd IEEE international conference on Internet multimedia services architecture and applications*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 229–234.
- [25] M. Alicherry and A. D. Keromytis, "Securing manet multicast using diploma," in *Proceedings of the 5th International Workshop on Security (IWSEC)*, November 2010, pp. 232–250.
- [26] K. Argyraki and D. R. Cheriton, "Network capabilities: The good, the bad and the ugly," in *ACM HotNets-IV*, 2005.
- [27] S. Mueller, R. Tsang, and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Performance Tools and Applications to Networked Systems, volume 2965 of LNCS*. Springer-Verlag, 2004, pp. 209–234.
- [28] W. chang Feng, D. D. Kandlur, D. Saha, and K. G. Shin, "Stochastic fair blue: A queue management algorithm for enforcing fairness," in *Proceedings of IEEE INFOCOM*, 2001, pp. 1520–1529.
- [29] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," in *EUROCRYPT*, 2003, pp. 416–432.