An Application of Bloom Filters in Wireless sensor Networks

N.Ramadevi, Assistant Professor

Department of Computer Science and Engineering, Santhiram Engineering College, Nandayl-518502, A.P

Dr.P.Prabhakaran,Professor

Department of Computer Science and Engineering, Santhiram Engineering College, Nandayl-518502, A.P

ABSTRACT

Network-based attacks can be either persistent or sporadic. Persistent attack flows can be relatively easy to trace by mechanisms such as probabilistic packet marking, traffic logging, data mining etc. Sporadic attacks are sometimes easily detected by the Intrusion Detection Systems (IDSs) at the victims, but are hard to trace back to the attack origins. We propose CAPTRA, a Coordinated Packet Trace back mechanism, for wireless sensor networks (WSNs) that takes advantage of the broadcasting nature of the packet transmissions. By remembering packets in multi-dimensional Bloom filters distributed in overhearing sensors and later retrieving the information, CAPTRA identifies the path of the packet transfers using a series of REQUEST-VERDICT-CONFESS message exchanges between the forwarding and overhearing nodes. CAPTRA requires only small memory footprint on the sensors due to the usage of Bloom filters, and allows sensors to asynchronously refresh the Bloom filters so that the network traffic is continuously monitored. CAPTRA is simulated using J-Sim, and a few key parameters are tuned for the best tracing performance.

Keywords: Wireless sensor networks, packet trace back, Bloom filter.

INTRODUCTION

Wireless sensor networks (WSNs) are networks of a large number of small wireless devices, which collaborate to monitor environments and report sensing data via wireless channels. Wireless sensors are very limited in their computing, communication and storage capabilities due to extremely constrained battery power supply and costs. For instance, the typical Crossbow MICA mote MPR300CB [7] has a low speed 4MHz processor equipped with only 128KB flash, 4KB SRAM and 4KB EEPROM. It has a maximal data rate of 40kbps and a transmission range of about 100 feet, powered by two AA batteries. In addition, a wireless sensor network is usually deployed with high density in order to improve a WSN's reliability and lifetime.

Unfortunately, a lot of the existing research in the packet trace back techniques and architectures were designed for Internet packet trace back, and are not viable in WSNs because of the characteristics of WSNs. First of all, human intervention or in-network intrusion detection are not possible to prevent attacks due to the wide deployment nature for distributed processing, and the limited power and silicon supply. Secondly, the wireless transmission medium is openly accessible to anyone in the network adjacency, thus making the network vulnerable to malicious packet injections. Third, sensor networks are capable of in-network data processing, thus eliminating the need for accurate topology maintenance, which makes address-based packet filtering and location inference impossible.

The rest of the paper is organized as follows. Section 2 reviews the current techniques and architectures for intrusion detection and packet trace back in the Internet. Section 3 briefly describes the widely used Bloom filter, and introduces a new perspective regarding the dimensions of Bloom filter, which we call space-time Bloom filter (STBF), a special case of multi-dimensional hash table (MDHT), with the Bloom filter as a special case under this perspective. Section 4 discusses the methods of applying STBF for packet trace back. Section 5 describes our coordinated packet trace back protocol in details, and Section 6 presents the evaluation results of the protocol in a WSN using simulations.

2. ASSOCIATED WORK

Network attacks can be either persistent or sporadic [28]. In persistent attacks, the offenders must frequently launch attack packets to bombard the victims. Whereas in sporadic attacks, a single packet can render havoc at the potential victim, such as the WinNuke, Ping of Death and Teardrop attacks. Intrusion Detection Systems (IDSs) use attack signature or pattern to help distinguish malicious packets from normal traffic. At the very least, an attack signature is defined by the IP address or address range of the entity that is being attacked. A variety of methods for IDSs were discussed in [26] that can help detect if received packets have spoofed source addresses.

Source route identification problem is commonly referred to as network trace back, or IP packet trace back in the Internet arena. Two network tracing problems are currently being studied: “single-domain IP trace back” and “trace back across stepping-stones” (or a “connection chain”). Trace back across stepping-stones is to identify the origin of an anonymous attacker through a chain of connections before the attacker interacts with a victim host. The Decentralized Source Identification System (DECIDUOUS) uses IPsec security associations (SAs) and authentication headers to dynamically deploy secure authentication tunnels, and traces back to the attacks origins [5, 6]. The premise of DECIDUOUS is that if an attack packet has been correctly authenticated by a certain router, the attack packet must have been transmitted through that router.

Snoeren et al. proposed an architecture, Source Path Isolation Engine (SPIE), that integrates the IDS and single packet trace back engines [23] to reconstruct the identify the attach graph. According to SPIE, once the IDS detects an abnormal attack event, the attack packet is fed into the trace back manager to generate a trace back request, which is sent to multiple network agents for constructing the regional attack graph based on the attack packet. Afterward, the regional attach graphs are assembled into a complete attack graph at the traceback manager, and fed back the IDS. SPIE is based on Bloom filter [3] for packet logging purposes [21]. To our knowledge, packet traceback in WSNs is among the few with little research so far.

Multi-Dimensional Hash Table

Bloom Filter and Its Variants A Bloom filter is a space-efficient probabilistic data structure that is used to test whether or not an element is a member of a set [3]. Bloom filters are used in myriad of applications. Wherever a list or set is used, and space is a consideration, a Bloom filter is commonly considered [4]. Fig. 1(a) illustrates how the Bloom filter is updated when a new element is inserted into the set, where k hash functions based on the same input string produce k index keys to update the single Bloom filter. Traditional hash table is a special case of Bloom filter where k = 1, shown in Fig. 1(a). Usually, elements are only added to the set, but not removed. Given a finite set, false positive judgment of the membership are possible, but false negative judgment are not in the traditional Bloom filters without refreshing. When the more elements are added to the set, the probability of false positive becomes greater.

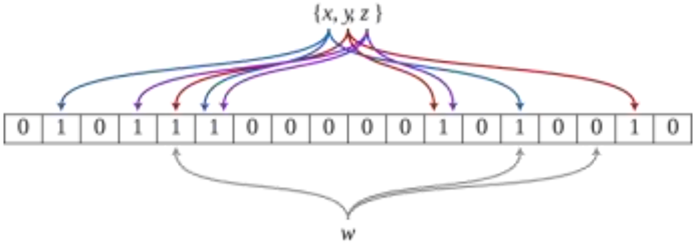


Fig.1. Bloom Filter.

Several variants of Bloom filters have been proposed. Attenuated Bloom filters [18] use arrays of Bloom filters to store shortest path distance information. Spectral Bloom filters [20] extend the data structure to support estimates of frequencies. In Counting Bloom Filters [11] each entry in the filter need not be a single bit but rather a small counter. Insertions and deletions to the filter increment or decrement the counters respectively. When the filter is intended to be passed as a message, compressed Bloom filters [16] may be used, where parameters can be adjusted to the desired tradeoff between size and false-positive rate. Spacecode Bloom filter provides frequency estimation of an element by probabilistically filling up multiple normal Bloom filters, from which to statistically infer the frequency of the element [13].

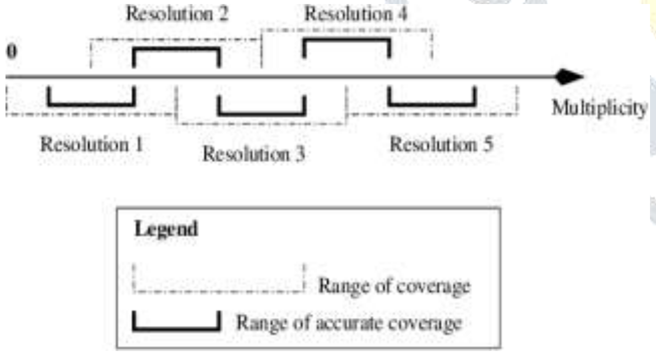


Fig.2. Space-Code Bloom Filter for Traffic Measurement

Fig. 2 illustrates the space-code Bloom filter (SCBF) for accounting flow volume, which is represented by the number of packet in the flow. A randomly chosen Bloom filter of the SCBF module takes the flow ID (source and destination IP addresses and port numbers) as the inputs, and record the membership in the table. When the Bloom filters in the SCBF module saturate, the contents of the Bloom filters are paged out into a log file. Later, according to the number of positive answers to a flow query into the log files offline, the flow volume can be probabilistically inferred.

3. GENERIC PACKET TRACE BACK USING SPACE-TIME BLOOM FILTER

We take similar assumptions about network operations for network-assisted packet trackback as in [23] that trace back operations are infrequent so that trackback overhead contributes only a minor addition to the overall network traffic, and that all trackback messages can be authenticated [24]. In addition, we assume that attackers may be aware of the packet trackback operations, therefore avoid being traced through certain other techniques, such as probabilistic packet marking scheme [12], and that router could be subverted, in which case the packet trackback works until it reaches the faulty router.

Packet Digest

Because the main purpose of packet trackback is to capture the path traversed by a specific packet through the network, without loss of generality, we extract the certain identification information to derive the packet digest. For instance, the packet digest of an IP packet may include IP version, header length, source and destination addresses, fragment information and portions of the payload [23], which is then concatenated and packed into a bit string as the input to hash functions. Note that not all packets are necessarily traced, but only those with potential vulnerabilities to certain network attacks. It is a configurable behavior as to which of the traffic flows are to be tracked and traced in the tracing protocols.

Trace back Operations

In support of packet trace back operations, three ICMP messages with new codes are proposed as shown in Table 2. Note that the use of ICMP messages is strictly for convenience, which is extended into our WSN simulations. Without loss of generality, similar message types can be implemented in real deployments for WSNs. These three messages jointly provide the majority-vote mechanism to corroborate the conviction of a node as one on the packet source route. Message TRACREQ is to initiate a traceback query to discover the predecessor of a packet, implemented using the normal ICMP message of type 8 “Echo Reply”, used for ping-ing a network host. It is started by the access point in WSNs, and recursively carried out by the nodes on the packet source route. In the payload field of the ICMP message, the packet identification is carried.

Coordinated packet trace back

We assume that a wireless sensor network consists of low power computers that are interconnected via a shared wireless channel using the same channel access control protocol, such as the simplified IEEE 802.11 [1] without collision avoidance mechanism using RTS/CTS. In many wireless sensor network deployments, the use of access points (APs) as an aggregation point and the exit to the Internet is a common practice. APs possess sufficient computing power to implement the intrusion detection mechanisms and initiate trace back operations. In addition, we assume that the sensors’ memories are limited so that the Bloom filters cannot be arbitrarily large, nor can there be permanent storage for logging purposes.

For simplicity, we assume that nodes are homogeneous in wireless sensor networks in terms of memory allocation, hashing algorithm and hit reporting frequency.

4. EVALUATIONS

Simulation Setup

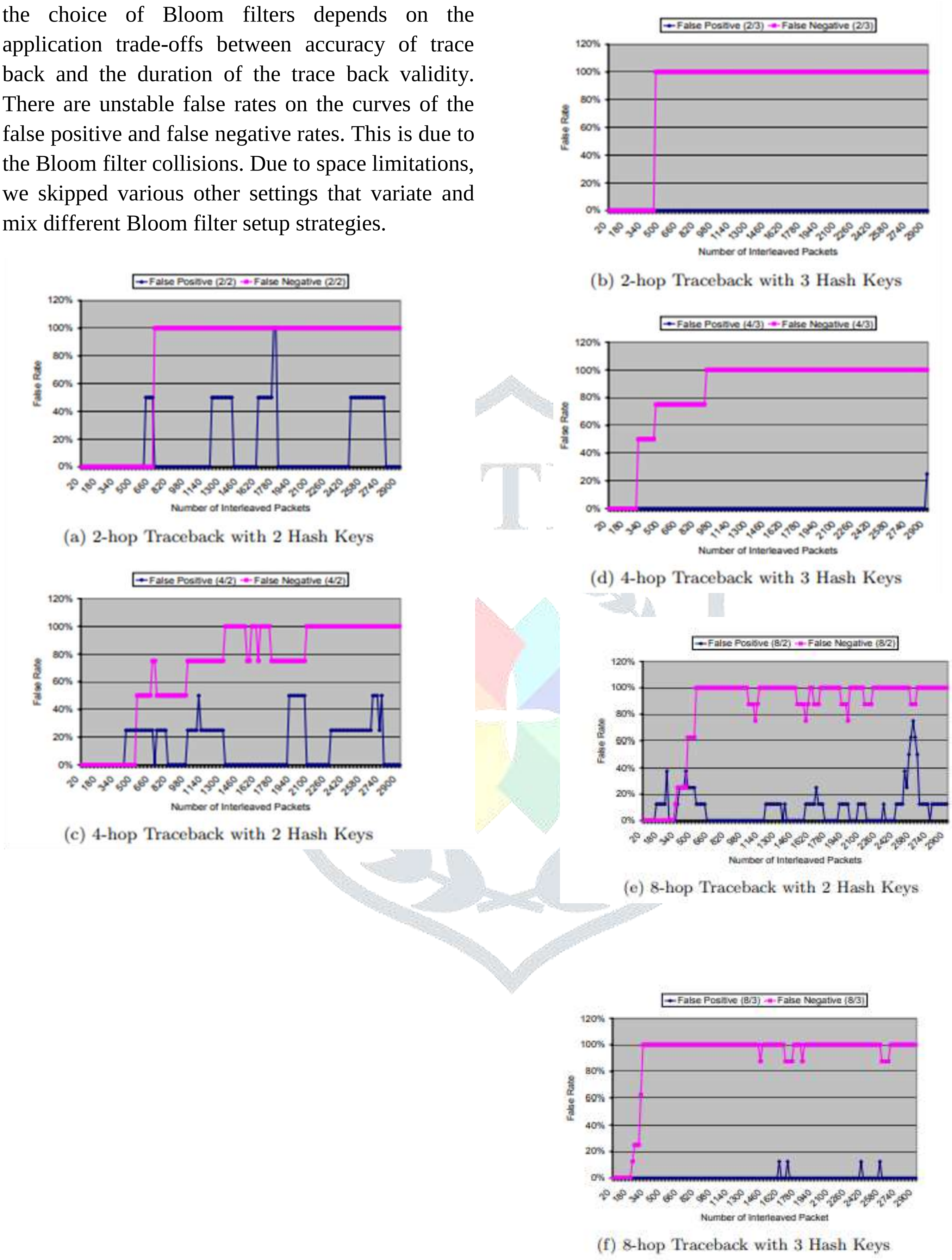
We used J-Sim [27] to simulate the packet traceback protocol in wireless sensor networks. J-Sim is a network simulator constructed entirely in JAVA. The MAC layer of the wireless sensors was modified to implement CAPTRA, RTS/CTS was turned off to simulate a WSN. For simplicity, the hashing function uses the senders MAC address as the pkt.tx, and the receiver’s MAC address as i in Eq. 1. In the simulations, twenty wireless sensors, numbered from 0 to 19, are deployed in a linear fashion, spaced 8 meters apart. Each node has a 25-meter transmission range, so that each node has up to 7 neighbors. The propagation and path-loss model use the free-space model. AODV is used as the underlying routing protocol. Each node contains a Bloom filter of size 4096 bits, which is refreshed using the “50% Golden Rule”. The number of hash functions for each packet, k, is a system-wide variable, and varies in two simulation scenarios from 2 to 3 for comparison purposes.

Simulation Results and Analysis

Fig. 7 presents the false positive and false negative rates under the various simulation scenarios. Two types of Bloom filters are implemented and compared side-by-side, in which one type of Bloom filter uses 2 hash functions in all the simulation scenarios, and the other uses 3 hash functions in each. To calculate false positive rates we simply counted the number of false positives by the sensors of the WSN and divided by the hop count. Similarly, for false negative rates, we had the number of false negatives divided by hop count.

In Fig. 7 (a) (c) and (e), the trace back performance is measured for 2-, 4- and 8-hop source route of a particular packet. As we can see, when the number of hops increases, the network has shorter and shorter memory of the traced packet, indicated by the less number of interleaved packets from the time when the traced packet is generated to the time when the packet is traced back. In addition, the duration of the network memory about the packet is depended on the network density, and the size of the Bloom filters, which are 4096 bits in our simulations.

On the other hand, Fig. 7 (b) (d) and (f) show the false positive and false negative rates of similar simulations when the number of hash functions is three. Comparing with the other three corresponding sub-diagrams, these three settings presents more stable trace back performance, but shorter memory of the traced packet due to the faster fill-up rate of the Bloom filters. Therefore, the choice of Bloom filters depends on the application trade-offs between accuracy of trace back and the duration of the trace back validity. There are unstable false rates on the curves of the false positive and false negative rates. This is due to the Bloom filter collisions. Due to space limitations, we skipped various other settings that variate and mix different Bloom filter setup strategies.



(a) 2-hop Traceback with 2 Hash Keys

(b) 2-hop Traceback with 3 Hash Keys

(c) 4-hop Traceback with 2 Hash Keys

(d) 4-hop Traceback with 3 Hash Keys

(e) 8-hop Traceback with 2 Hash Keys

(f) 8-hop Traceback with 3 Hash Keys

Fig. 3. The False +ve and –ve Rates in MPT.

5. CONCLUSION

We have introduced a new perspective on expanding Bloom filter dimensions, which led to the Space-Time Bloom Filter (STBF) for coordinated packet traceback (CAPTRA) in wireless sensor networks (WSNs). CAPTRA takes advantage of the broadcast nature of the wireless transmissions for coordinated traceback, and is especially suitable for WSNs due to the small memory and computational requirements in maintaining STBF. The space-time Bloom filter construction enables distributed maintenance and asynchronous and gradual refreshing of the Bloom filters so that the WSN keeps robust and gradually fading memory of the packet traversal event. These unique organization and utilization of Bloom filters in CAPTRA allows our future application of packet trace back in adaptive routing in WSNs.

REFERENCES

[1] IEEE Std 802.11. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Technical report, IEEE, Jul. 1997.

[2] S.M. Bellovin, Marcus Leech, and Tom Taylor. ICMP Traceback Messages. Technical report, Internet Draft, IETF, Mar. 2001.

[3] B.H. Bloom. Space/time trade-offs in hash coding with allowable errors. Communications of ACM, 13(7):422–426, Jul. 1970.

[4] A. Broder and M. Mitzenmacher. Network applications of Bloom filters: a survey. In Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing, 2002.

[5] H.Y. Chang, P. Chen, A. Hayatnagarkar, R. Narayan, P. Sheth, N. Vo, C. L. Wu, S.F. Wu, L. Zhang, X. Zhang, F. Gong, F. Jou, C. Sargor, and X. Wu. Design and Implementation of A Real-Time Decentralized Source Identification System for Untrusted IP Packets. In Proceedings of the DARPA Information Survivability Conference and Exposition, Jan. 2000.

[6] H.Y. Chang, R. Narayan, C. Sargor, F. Jou, S.F. Wu, B.M. Vetter, F. Gong, X. Wang, M. Brown, and J.J. Yuill. DECIDUOUS: Decentralized Source Identification for Network-Based Intrusions. In Proceeding of 6th IFIP/IEEE International Symposium on Integrated Network Management, pages 702–714, 1999.

[7] Inc. CrossBow Technology. http://www.xbow.com, 2005.

[8] D. Dean, M. Franklin, and A. Stubblefield. An Algebraic Approach to IP Traceback. In Proceedings of Network and Distributed System Security Symposium, Feb. 2001.

[9] T.W. Doeppner, P. N. Klein, and A. Koyfman. Using Router Stamping to Identify the Source of IP Packets. In 7th ACM Conference on Computer and Communications Security, pages 184–189, Athens, Greece, Nov. 2000.

[10] J. R. Douceur. The Sybil attack. In First International Workshop on Peer-to-Peer Systems (IPTPS), Mar. 2002.

[11] L. Fan, P. Cao, J. Almeida, and A. Broder. Summary cache: A scalable wide-area Web cache sharing protocol. In Proceeding of SIGCOMM, 1998.

[12] M.T. Goodrich. Efficient Packet Marking for Large-Scale IP Traceback. In 9th ACM Conf. on Computer and Communications Security (CCS), pages 117–126, 2002.

[13] A. Kumar, J. Xu, E.L. Li, and J. Wang. Space-code bloom filter for efficient traffic flow measurement. In Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, pages 167–172, Miami Beach, FL, 2003.

[14] J. Li, M. Sung, J. Xu, and L. Li. Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation. In IEEE Symposium on Security and Privacy, Berkeley, CA, May 2004.

[15] A. Mankin, D. Massey, C. Wu, S. F. Wu, and L. Zhang. On Design and Evaluation of Intention-Driven ICMP Traceback. In Proceedings of IEEE International Conference on Computer Communications and Networks (IC3N), 2001.

[16] M. Mitzenmacher. Compressed Bloom Filters. IEEE/ACM Transactions on Networks, 10(3):613–620, Oct. 2002.

[17] K. Park and H. Lee. On the Effectiveness of Probabilistic Packet Marking for IP Traceback. In Proceedings of SIGCOMM, pages 15–26, 2001.

[18] S.C. Rhea and J. Kubiatowicz. Probabilistic Location and Routing. In INFOCOM, 2002.

[19] R. Rivest. RFC 1321 - The MD5 Message-Digest Algorithm. Technical report, MIT Laboratory for Computer Science and RSA Data Security, Inc., Network Working Group, Apr. 1992.

[20] Y. Matias S. Cohen. Spectral Bloom Filters. In SIGMOD Conference on Management of Data, pages 241–252, 2003.

[21] L.A. Sanchez, W.C. Milliken, A.C. Snoeren, F. Tchakountio, C.E. Jones, S.T. Kent, C. Partridge, and W.T. Strayer. Hardware Support for a Hash-Based IP Traceback. In Proceedings of DARPA Information Survivability Conference and Exposition, Jun. 2001.

[22] S. Savage, D. Wetherall, A. Karlin, , and T. Anderson. Practical Network Support for IP Traceback. In Proceedings of ACM SIGCOMM Conference, Aug. 2000.

[23] A.C. Snoeren, C. Partridge, L.A. Sanchez, C.E. Jones, F. Tchakountio, S.T. Kent, and W.T. Strayer. Hash-based IP Traceback. In Proceedings of ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM), pages 3–14, 2001.

[24] D.X. Song and A. Perrig. Advanced and Authenticated Marking Scheme for IP Traceback. In Proceedings of IEEE INFOCOM Conference, 2001.