Stealth Probing: Securing IP Routing through Data-Plane Security

Ioannis Avramopoulos* and Jennifer Rexford†

June 27, 2005

Abstract

Securing IP routing is a task that is central in diminishing the Internet's liability to misconfigurations and malicious attacks. In this paper, we argue that securing the data plane is necessary in providing comprehensive routing defense and we propose the *stealth probing* monitoring tool that securely detects data plane inconsistencies that may be the outcome of either control plane or data plane attacks. Stealth probing correctly detects the fate of data traffic in a non-intrusive, coarse-grained, end-to-end fashion. Stealth probing achieves this by creating an encrypted tunnel between two end-routers and diverting both the data and probing traffic into the tunnel so that probes are indistinguishable from data packets and, therefore, cannot receive preferential treatment by an adversary. We also propose the use of a *Byzantine tomography* tool that complements stealth probing in achieving fine detection granularity by combining stealth probing output from multiple vantage points. We demonstrate the wide application span of stealth probing and Byzantine tomography by illustrative deployment scenarios in intradomain and interdomain routing.

1 Introduction

In IP routing, the entity that is responsible for detecting and bypassing failures is the routing protocol. Failures are detected by a periodic beaconing process and announced to other routers by means of routing advertisements. After receiving these announcements, each router independently builds a new routing table to circumvent the failed components. This pattern of operation has influenced IP routing design since the inception of the Internet and it is most effective if failures are fail-stop (i.e., fail and stop working). However, if failures are *arbitrary* (or *Byzantine*), perhaps due to misconfigurations or malicious attacks, this process tends to create "black holes". For example, consider a router that is controlled by an adversary and that is configured to normally send beacons but drop the received data packets.[1] The routing protocol cannot by itself detect this malfunction.

In this paper, we propose mechanisms that protect the routing function from such Byzantine failures. We assume that an adversary is present at network locations that are unknown to the defending entity and that the adversary uses those locations to mount attacks with the goal of disrupting the packet delivery service. In developing countermeasures, we assume that these attacks can be of arbitrary nature. We note, however, that the repertoire of attacks that are likely to happen in practice is limited by technological constraints. For example, those attacks that require changes in a router's firmware are less likely to happen than misconfiguration ones. Yet, a simple misconfiguration of a router's access control list (ACL) can invoke a network partition.

Our objective is to ensure that communication will not be disrupted in a significant majority of non-compromised source-destination pairs even in the presence of such attacks. Ensuring the availability of the network despite the presence of adversaries prevents financial losses and other detrimental societal impacts that these adversaries could otherwise inflict.

In protecting from Byzantine failures, we argue that we should consider all three dimensions of Internet routing, i.e., the *data*, *control*, and *management planes*: The data plane supports packet forwarding functionality such as simple destination-based forwarding, filtering, and tunneling. The control plane implements the routing protocol that discovers the topology and selects routes. The management plane, which typically serves as the "interface" between humans and the network, monitors the network and intervenes in the routing process by changing router configuration parameters.

The management and control planes have been the primary focus of the effort on devising counter-

*Department of Electrical Engineering, Princeton University. Email: iavramop@princeton.edu

†Department of Computer Science, Princeton University. Email: jrex@cs.princeton.edu

[1] This attack configuration is easy to assemble: The control plane must be left intact and the data plane must be configured with an access control list that selectively discards ingress traffic.

measures to Byzantine failures. As a management plane countermeasure, vendors issue Best Common Practices for configuring routers [20] but BCPs cannot protect from an adversary that has managed to gain presence inside a network. As a control plane countermeasure, several secure routing protocols have been proposed [10, 12, 13, 15, 25] with the goal of ensuring that the topological information in valid routing advertisements correctly identifies the connectivity between non-faulty routers (or Autonomous Systems). However, these protocols do not prevent an adversary from falsely advertising that faulty routers are connected to other faulty routers (as in the collusion [25] or wormhole [13] attacks) or non-faulty routers. These advertisements can skew the topological graph so that adversarial routers will receive and, therefore, control increased traffic volumes with potentially adverse impacts on network connectivity. Fortunately, these attacks can be troubleshooted with secure data plane monitors such as the one that we propose in this paper.

Although data plane countermeasures that securely monitor the flow of data packets can both identify control plane inconsistencies (such as wormholes) and forwarding attacks that target the data plane alone, the problem of devising data plane countermeasures apt for deployment in the operational Internet has not received due attention. The stealth probing protocol that we propose in this paper can serve as a secure data plane monitor that can correctly decide whether a router-to-router path is operational even if an adversary is present in the intermediate routers of the path and actively tries to coerce a false decision. Stealth probing achieves this by creating an encrypted tunnel between two end-routers and diverting both the data and probing traffic into the tunnel. In this way, the adversary cannot drop data packets without also dropping the probing packets, because they are indistinguishable from each other, and, therefore, cannot evade detection.

It is worth noting here that the general idea of using encryption to make data and control traffic indistinguishable from each other was first proposed by Perlman [22]. Perlman proposed the use of hop-by-hop encryption between neighboring routers in order to hide beaconing traffic and prevent topology-discovery-targeted "man-in-the-middle" attacks at network links. The novelty of stealth probing is in applying this general idea to network paths in an end-to-end manner in order to achieve secure data plane troubleshooting.

The following properties that stealth probing achieves argue to its viability: First, because stealth probing relies on efficient, end-to-end, symmetric cryptographic primitives, its processing overhead is kept low. Furthermore, its network overhead due to the probes is minimal and independent of the rate of the data flow. Second, because of its end-to-end design, it does not require explicit support from legacy routers in the core of the network. Third, the use of tunnels allows selectivity in the traffic that receives protection and, therefore, the extra overhead that is associated with the protection mechanism can only be applied to the critical traffic. Finally, it is simple enough to allow efficient implementations that will match the increasing line speeds of operational networks. However, stealth probing (intentionally) trades accuracy for efficiency and, therefore, cannot exactly locate offending routers in the communication paths that are monitored. We advocate that this diagnosis should be made outside of the data plane, for example, in the management plane by using a *Byzantine tomography module* that analyzes the output from multiple vantage points.

Stealth probing offers a secure detection capability, which must then be integrated in a secure failure recovery system that will process probing output and accordingly select non-faulty paths to forward packets to. Such recovery system is necessarily a deviation from the Internet's routing paradigm that uses beaconing to closely couple topology discovery and route selection. Since secure failure recovery requires data plane feedback that is more general than simple beaconing, a question that naturally arises is how to integrate this recovery system in the operational Internet. In Section 3, we propose viable deployment paths for stealth probing and Byzantine tomography at the intradomain and interdomain routing levels, which we believe provide insight in answering the above question. Related work is discussed in Section 4 and this paper is concluded in Section 5.

2 Stealth Probing

Stealth probing is a secure data plane monitoring tool that relies on the efficient symmetric cryptographic protection of the IPsec protocol suite that is applied in a end-router-to-end-router fashion. Section 2.1 discusses limitations of existing approaches to secure data plane monitoring. The idea and benefits of stealth probing are presented in Section 2.2 followed by the detailed workings in Section 2.3.

2.1 Limitations of Existing Approaches

The problem that stealth probing addresses is that of securely deciding whether a node-to-node path correctly delivers data packets from one end of the path to the other end. The decision process must be secured so that an adversary that is present at one or more intermediate nodes of the path cannot coerce a false decision. Furthermore, the overhead of

the decision process must be practical for deployment in operational networks.

Consider two routers u and v. Lets assume for simplicity that u is a source of data traffic and that v is a sink for this data traffic. We want to verify that data traffic is flowing properly in the forward $u \rightarrow v$ direction.

Probing One approach to meet our objective is for router u to send to v one or more ICMP echo request packets and infer the fate of data traffic based on the receipt of ICMP echo reply packets [24]. This method has the advantage that it is non-intrusive since it reaches a decision by using a small number of probes. However, if an adversary is present in the path between u and v, he can selectively drop data packets and selectively forward echo requests and replies. In this way, the adversary's misbehavior will go undetected.

Network-layer ACKs A second approach to meet our objective is to request from v to explicitly acknowledge the receipt of data packets from u. This approach has the following disadvantage: if an adversary is present in the path between u and v, he can drop data packets and avoid detection by forging destination acknowledgments. So, lets further assume that u and v share a secret symmetric key. Using this key, we can prevent this attack by requiring u to authenticate data packets by means of a message authentication code (MAC) and from v to authenticate destination acknowledgments in the same way. This scheme has the advantage that the adversary cannot prevent u from detecting the failure in the path. However, it requires the acknowledgment of data traffic at the network layer, which would be harmful to network overhead.

Transport-layer ACKs A third approach to path failure detection is by a host-to-host, cryptographically protected, transport layer protocol such as TLS (Transport Layer Security) [8]. However, the failure of a host-to-host path does not necessarily imply a failure in the routers that comprise the path. This scheme would, therefore, suffer from "false alarms" due to host failures that would complicate fault localization.

Traceroute A fourth approach to path failure detection is that adopted by the traceroute tool [24] that uses ICMP "time exceeded" and "port unreachable" messages to either determine the full path from a source to a destination or identify the first upstream router before a black hole. Traceroute has fine detection granularity at the *link* level but cannot prevent the preferential treatment of traceroute packets by an adversary who can in this way easily avoid detection.

2.2 A Minimal Secure Data Plane Monitor

Stealth probing has a "minimalist" design: By adhering to the *end-to-end principle* and providing secure *path-level* failure detection capability, it enables comprehensive recovery strategies from routing attacks and major misconfigurations, while keeping the data plane support to a minimum.

The idea in stealth probing is to use probes (ICMP echo requests and replies) to reach a secure decision on the fate of data traffic by establishing an encrypted and authenticated *tunnel* between two routers in the traffic's path and diverting both the data traffic and the probes into this tunnel; the role of encryption is to conceal the probing traffic so that it is indistinguishable from the data traffic; the role of authentication is make the tampering of data traffic detectable. Stealth probing also conceals the packet size and timing of probes in the following two ways: First, the size of probe packets is set to be equal to the size of data packets that are padded to the closest of a small number of sizes. Second, inter-probing intervals are chosen from a random distribution.

Stealth probing has the following primary benefits:

- Because stealth probing is an *end-router-to-end-router failure detection mechanism* the intermediate routers of a monitored path do not need to provide any support to the stealth probing monitor. This property has the implications that stealth probing is suitable for deployment across legacy routers and over interdomain paths.

- Stealth probing is *non-intrusive*. Intermediate routers do not need to process tunneled packets as they are tunnel agnostic. Processing requirements at tunnel endpoints are simple. Tunnel entry points must distribute packets to tunnels and apply the cryptographic processing step. Tunnel exit points must restore packets to their initial format by applying their portion of cryptographic processing. Furthermore, network overhead due to probing is minimal and independent of the rate of the data flow.

- The use of tunnels permits a great degree of *selectivity* in the traffic that is protected. For example, network management can use ACLs to classify traffic as critical and non-critical and apply the protection of encrypted tunnels only to the critical traffic while letting non-critical traffic bypass the tunnels.

- Stealth probing admits a simple implementation using *off-the-shelf* software components from

available IPsec (Section 2.3) and ICMP implementations.

The infrastructure that supports stealth probing can also be leveraged in the following ways:

- End-router-to-end-router encryption and encapsulation protects from the eavesdropping of unencrypted host-to-host communications and prevents traffic analysis attacks that host-to-host encryption cannot itself protect from (for example, it hides the source and destination addresses of the data traffic).

- Encrypted tunnels are extensively used as virtual links in VPNs [27]. Points-of-Presence in ISP networks are increasingly capable of terminating encrypted tunnels because of the value that the offering of a VPN service adds to the ISP service model. Such encrypted tunnels can serve as a shared infrastructure between stealth probing and VPNs.

- The property of fate sharing that stealth probing enforces between data traffic and probes is also useful for troubleshooting network problems in the absence of an adversary. For example, simple ICMP echo requests and replies may be treated differently from data packets either because of MTU size limits or packet filters that filter based on the protocol or port numbers. Stealth probing does not face similar problems due to the tunneling and padding steps that it applies to the data packets and probes.

- Tunneling is broadly useful for regulating interdomain traffic inside an ISP network according to traffic engineering or other objectives [26].

2.3 Mechanics

Stealth probing requires the endpoints of a monitored path to share a secret and use this secret to create an *IPsec tunnel*. This section charts the workings of the IPsec protocol suite and the process that administers packets into tunnels.

The IPsec Protocol Suite: IPsec provides strong end-to-end cryptographic protection at the IP layer. To this end, it specifies the Internet Key Exchange (IKE) protocol [11] through which two communicating parties negotiate the establishment of a Security Association (SA). Following the SA establishment, IP packets are protected using an Encapsulating Security Payload (ESP) module [14] that includes a tunnel mode of operation.

In the tunnel mode of ESP, a new outer IP header is added to each packet at the tunnel entry point that is followed by the ESP header and trailer that wraps the original IP packet. The role of ESP is to provide encryption using a standard encryption algorithm such as AES [7] and to ensure the authenticity and integrity of protected packets using a standard message authentication code (MAC) such as HMAC-SHA1 [17]. The tunnel exit point removes the outer IP header and restores the inner IP packet after a cryptographic processing step. The cryptographic protection that stealth probing requires to be applied to data traffic is, therefore, only based on efficient symmetric cryptographic primitives. Thus, packet processing can readily proceed at the line speeds of operational networks at the core of the Internet. ESP can be implemented at the kernel but also admits "bump-in-the-stack" (BITS) and "bump-in-the-wire" (BITW) implementations. A BITW implementation can reside in a standalone "IPsec box" [9].

The authentication methods used in IKE rely on either preshared secret keys or public-private key pairs and an associated Public Key Infrastructure (PKI) that issues certificates of the public keys. In intradomain routing the key exchange problem can be naturally assumed by the corresponding domain's authority. In interdomain routing the key exchange problem becomes more challenging due to the absence of a central trusted authority. Because of its end-to-end design, stealth probing has a significant advantage in this regard. IKE is typically implemented at the application layer.

Administering Packets into Tunnels: Network management will use ACLs to specify the securely monitored traffic based on the five-tuples of source and destination address prefixes, port numbers, and protocol numbers. Tunnels will be accordingly deployed across the network to match this specification (see Section 3). A multidimensional packet classifier at tunnel entry points will determine, based on the ACLs, the portion of ingress traffic that will enter the encrypted tunnels. For protected packets, a longest prefix match table lookup will determine based on a packet's destination address the tunnel exit point. A simple table lookup will then retrieve the associated encryption keys before the application of ESP processing.

3 Deployment Scenarios

In this section, we illustrate and contrast possible deployment scenarios for stealth probing in the intradomain and interdomain routing levels.

3.1 Intradomain Routing

Identifying tunnel endpoints: Medium to large ISP networks are typically organized in a three-tier hierarchy [18, 28]. At the lowest level of the hierarchy lies the *network edge layer* whose routers are

responsible for aggregating customer networks and terminating transit and peering connections. At the middle level of the hierarchy lies the *network aggregation layer* whose routers are responsible for aggregating edge routers. At the top level of the hierarchy lies the *network core layer* whose role is to connect routers of the aggregation layer in a dense high-speed mesh. Apt locations to deploy stealth probing monitors are the edge routers because in this way we leverage the benefits of an end-to-end design: First, aggregation and core routers can be tunnel agnostic and need only support simple destination-based forwarding and, second, processing requirements are distributed over a large number of edge routers.

As in Section 2.3, network management will use five-tuples to specify protected traffic. Tunnel endpoints will be accordingly determined by the NEXT_HOP attribute of BGP UPDATE messages exchanged in the iBGP mesh. A longest prefix match on a packet's destination address will determine the tunnel exit point for each packet and a simple table lookup will then determine the encryption and authentication keys. The number of edge routers in a large ISP network is on the order of a few hundred. The number of tunnels that each edge router will terminate and the authentication and encryption keys that must be stored will, therefore, be in this order of magnitude. Compared to the overhead of a FIB (Forwarding Information Base) lookup that must be normally performed for each IP packet (independent of the stealth probing protection step), the overhead of retrieving encryption keys is low.

Byzantine tomography: If the network is under attack, stealth probing monitors will detect dysfunctional paths. This fault knowledge will be made available to the management plane, which must then assume the responsibility of enforcing a recovery strategy. The management plane is in an advantageous position to perform a recovery plan because of its capability to restore the behavior of routers from arbitrary to normal. Restoring the behavior of a router from arbitrary to normal can take place, in the simplest case, by correctly reconfiguring that router or, in the worst case, by reinstalling the router's operating system. However, this restoration entails the associated penalty of a gratuitous "downtime" in case of a false alarm, according to which a correctly functioning router is selected to be restored. Fine detection capability would, therefore, be beneficial in reducing false alarms. This level of granularity can be gained by a *Byzantine tomography module*.

Byzantine network tomography satisfies the requirement for detection capability at a finer level than the end-to-end level that stealth probing offers by combining the stealth probing output from multiple vantage points. It generalizes previous work on network tomography [5], which addresses the problem of identifying the loss rates of network links using end-to-end probing traffic, by further addressing the case that (the unknown) malicious routers may lie to other routers about their collected measurements.

Byzantine tomography estimates the *faulty configuration* of the network, i.e., which routers are faulty and non-faulty, from the cumulative fault knowledge that is obtained by the probes. The faulty configuration that Byzantine tomography proffers minimizes over all possible faulty configurations the number of routers that explain the fault knowledge. Algorithmically this problem is an instance of the Minimum Hitting Set (MHS) problem [6]: If S is the set of routers in the network and C is a collection of subsets of S that corresponds to the fault knowledge (i.e., the collection of paths that are faulty), a *hitting set* for C is a subset S' of S such that S' contains at least one element from each subset in C. MHS can be solved using one of the algorithms presented in [3, 16].

The adversary's goal is to disorient the management plane into missed and false detections by manipulating the fault knowledge. The adversary is in the position to do this manipulation in the following two ways: In the first, he can instruct his routers to drop data traffic and, in the second, he can instruct his routers to spuriously announce dysfunctional paths.[2] However, to what extent can the adversary use these manipulation strategies as a vantage to disrupt communication and evade detection? Byzantine tomography estimates the faulty configuration by the minimum fault explanation whose accuracy increases as the size of the fault knowledge increases. In order to limit the detection accuracy, the adversary must, therefore, confine the scope of his attacks, which in turn implies that the adversary's yield (and the associated impact to the availability of network) is fairly limited.

3.2 Interdomain Routing

Securing interdomain routing is a problem arguably harder than that of securing intradomain routing because, first, in the absence of a central trusted authority, key distribution becomes more challenging and, second, failure detection and recovery must detect and bypass faulty networks outside of one authority's administrative control. Because stealth probing is an end-to-end failure detection mechanism it simplifies the key distribution problem

[2]Note, however, that these spurious announcements either identify faulty paths, if the origin of the path is the same as the origin of the announcement, or are immediately identifiable as spurious.

in a way that is both incrementally deployable and backward compatible.

Incremental deployability: We envision that in the initial stages of its deployment the ASes that are willing to deploy stealth probing over interdomain paths will engage in bilateral or small-scale multilateral agreements and exchange pairwise keys either by manual configuration involving the network operators or by small-scale PKIs. In this scenario, encrypted tunnels may well serve a dual functionality as virtual private links in multi-site VPN deployments in addition to their robust troubleshooting purpose that will amortize the key exchange overhead. ISPs will have the incentive to join small-scale groups in order to be able to both provide an enriched service model with VPN capabilities and securely detect (and then troubleshoot) connectivity problems and, therefore, ensure higher availability of their services. Because stealth probing can be deployed across tunnel-agnostic legacy routers, early adopters will see an immediate benefit. As better availability will be an incentive for more ISPs to join these groups, the population base of stealth probing will increase and scalable key distribution will have to be addressed perhaps by a distributed trust model.

Circumventing the adversary: Although securely detecting routing failures is an important capability in its own right, the ability to bypass routing failures offers an additional significant advantage. In the following, we present a technique that can achieve this objective.

Consider two stub ASes, AS_1 and AS_2, and assume that AS_1 is m_1 − multihomed and AS_2 is m_2 − multihomed. For simplicity, also assume that each of AS_1 and AS_2 has a single border router. Using intelligent route control techniques [1], the border router of AS_1 can choose among $m_1 m_2$ different BGP paths to forward traffic in the $AS_1 \rightarrow AS_2$ direction. The border router of AS_1 can forward traffic to any of its m_1 outgoing links in a straightforward manner. Furthermore, this border router can choose any of the m_2 incoming links to AS_2 in the following way: First, AS_2 advertises a different primary prefix to each of its m_2 providers. Second, the border routers of AS_1 and AS_2 establish m_2 tunnels. Each of these tunnels has a destination address taken from the m_2 distinct prefixes that AS_2 has advertised. In this way, the border router of AS_1 can select any of the m_2 incoming links to AS_2 by diverting packets to the corresponding tunnel. Forwarding traffic in the $AS_2 \rightarrow AS_1$ direction is completely analogous. In this configuration, stealth probing can detect which of those $m_1 m_2$ paths are failing and workable paths can be accordingly chosen.

4 Related Work

Listen [25] is a low-overhead data plane monitor that infers prefix reachability problems by passively observing the connection establishment phase of TCP flows. An adversary that passively drops data packets and actively impersonates hosts in the monitored prefixes can coerce Listen to falsely decide that a prefix is reachable.

Mizrak et al. [19] present a secure data plane monitoring protocol that detects failures based on the comparison of hash-based data traffic *aggregates* or *summaries*. In the two variations of the protocol that are presented, one has fine detection granularity at the link level at the expense of high overhead whereas the second has practical overhead at the expense of a detection granularity at the *path* level. Except for noting that stealth probing and end-to-end traffic summarization are conceptually disparate techniques to achieve a similar objective, we will defer a detailed comparison between the two approaches.

Secure data plane monitors with fine detection granularity at the *link* level [2, 4, 21] require path-specific authentication that increases overhead, complicates the implementation, and faces a more demanding key distribution problem.

Perlman [22] proposes two data plane mechanisms for recovery from routing attacks using multipath routing and disjoint paths. The less computationally intensive of these mechanisms relies on a route establishment phase that is protected with digital signatures, which is followed by a forwarding phase that only requires end-to-end cryptographic protection of data packets. Stealth probing is well-suited in this case to monitor the quality of active paths in order to dynamically recompute the active path set.

5 Concluding Remarks

In this paper, we investigated the capability of the IP routing system to recover from misconfigurations and attacks and we proposed the stealth probing and Byzantine tomography tools for secure failure detection and recovery. Through deployment scenarios we showed that stealth probing and Byzantine tomography have a wide application span as secure "point fixes" of the IP routing system.

Consider, however, the failure recovery technique of Section 3.2. In this technique, although end-routers gain control of the first and last AS hops between the communicating endpoints, the intermediate ASes of each path that is selected in this way are chosen by BGP. Because BGP is oblivious to routing attacks, better recovery capability would be gained if end-routers were able to select complete end-to-end AS paths. Satisfying this latter requirement would

require support for source routing as in, for example, the Platypus source routing system [23]. Alternatively, such support could be provided in a more scalable manner by the interdomain routing architecture, however, such architectural help would require a significant shift from the operational routing paradigm. The fittest strategy for better control of interdomain paths is a topic of future work.

In the future, we also plan to empirically evaluate stealth probing and Byzantine tomography by deploying them in an operational network.

References

[1] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In *Proc. ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[2] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy. Highly secure and efficient routing. In *Proc. IEEE Infocom*, Hong Kong, Mar. 2004.

[3] I. Avramopoulos, A. Krishnamurthy, H. Kobayashi, and R. Wang. Nicephorus: Striking a balance between the recovery capability and the overhead of Byzantine detection. Technical Report TR-710-04, Dept. of Computer Science, Princeton University, Oct. 2004.

[4] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An on-demand secure routing protocol resilient to byzantine failures. In *Proc. ACM Workshop on Wireless Security*, Atlanta, GA, Sep. 2002.

[5] T. Bu, N. Duffield, F. Lo Presti, and D. Towsley. Network tomography on general topology. In *Proc. ACM SIGMETRICS*, Marina Del Rey, CA, Jun. 2002.

[6] P. Crescenzi and V. Kann. *A Compendium of NP Optimization Problems*. http://www.nada.kth.se/~viggo/problemlist/compendium.html.

[7] J. Daemen and V. Rijmen. The block cipher Rijndael. In J.-J. Quisquater and B. Scheier, editors, *Smart Card Research and Applications*, LNCS 1820, pages 288–296. Springer-Verlag, Oct. 2000.

[8] T. Dierks and C. Allen. The TLS protocol version 1.0. RFC 2246, Internet Engineering Task Force, Jan. 1999.

[9] S. Frankel. *Demystifying the IPsec Puzzle*. Artech House, 2001.

[10] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In *Proc. Network and Distributed System Security Symposium*, San Diego, CA, Feb. 2003.

[11] D. Harkins and D. Carrel. The Internet Key Exchange (IKE). RFC 2409, Internet Engineering Task Force, Nov. 1998.

[12] Y.-C. Hu, A. Perrig, and D. Johnson. Efficient security mechanisms for routing protocols. In *Proc. Network and Distributed System Security Symposium*, San Diego, CA, Feb. 2003.

[13] Y.-C. Hu, A. Perrig, and M. Sirbu. SPV: A secure path vector routing scheme for securing BGP. In *Proc. ACM SIGCOMM*, Portland, OR, Sep. 2004.

[14] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2406, Internet Engineering Task Force, Nov. 1998.

[15] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (Secure-BGP). *IEEE Journal on Selected Areas in Communications*, 18(4):582–592, Apr. 2000.

[16] R. Kompella, J. Yates, A. Greenberg, and A. Snoeren. IP fault localization via risk modeling. In *Proc. Symposium on Networked System Design and Implementation*, Boston, MA, May 2005.

[17] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-hashing for message authentication. RFC 2104, Internet Engineering Task Force, Feb. 1997.

[18] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the Internet's router-level topology. In *Proc. ACM SIGCOMM*, Portland, OR, Aug. 2004.

[19] A. Mizrak, Y.-C. Cheng, K. Marzullo, and S. Savage. Fatih: Detecting and isolating malicious routers. In *Proc. International Conference on Dependable Systems and Networks*, Yokohama, Japan, Jun. 2005.

[20] O. Nordstrom and C. Dovrolis. Beware of BGP attacks. *ACM SIGCOMM Computer Communication Review*, 34(2), Apr. 2004.

[21] V. Padmanabhan and D. Simon. Secure traceroute to detect faulty or malicious routing. In *Proc. ACM SIGCOMM HotNets Workshop*, Princeton, NJ, Oct. 2002.

[22] R. Perlman. *Network Layer Protocols with Byzantine Robustness*. PhD thesis, Massachusetts Institute of Technology, Aug. 1988.

[23] B. Raghavan and A. Snoeren. A system for authenticated policy-compliant routing. In *Proc. ACM SIGCOMM*, Portland, Oregon, Sep. 2004.

[24] R. Stevens. *TCP/IP Illustrated Volume 1: The Protocols*. Addison Wesley, 1994.

[25] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. Katz. Listen and Whisper: Security mechanisms for BGP. In *Proc. Symposium on Networked System Design and Implementation*, San Francisco, CA, Mar. 2004.

[26] R. Teixeira, T. Griffin, M. Resende, and J. Rexford. TIE: Tunable interdomain egress selection. Technical Report TD-69EJBE, AT&T Labs, Feb. 2005.

[27] R. Yuan and W. Strayer. *Virtual Private Networks: Technologies and Solutions*. Addison Wesley, 2001.

[28] R. Zhang and M. Bartell. *BGP Design and Implementation*. Cisco Press, 2004.