Towards Securing Interdomain Routing on the Internet

Sharon Goldberg

A Dissertation

Presented to the Faculty

of Princeton University

in Candidacy for the Degree

of Doctor of Philosophy

Recommended for Acceptance

by the Department of

Electrical Engineering

Advisors: Jennifer Rexford, Boaz Barak

September, 2009

Abstract

The Internet consists of multiple autonomous systems (ASes), each consisting of networks of devices that are prone to malfunction, misconfiguration, or attack by malicious parties, and each controlled by profit-seeking businesses with different economic goals. Despite these complex relationships, the interdomain routing system (that allows ASes to communicate over the global Internet) currently operates under the assumption that all nodes in the network can trust each other. The thesis contributes to the body of works that seeks to remedy this, by considering network protocols that operate correctly even in the presence of adversarial or selfish behavior.

We take a principled approach to analyze the types of security guarantees that are possible within the engineering and economic constraints of the Internet's interdomain routing system. We focus exclusively on protocols that can be used to improve availability in the Internet, *i.e.,* to increase the likelihood that packets arrive uncorrupted at their correct destination, and analyze two broad themes:

1. Which part of the system should be secured?

2. What is the right tradeoff between security and efficiency?

To address these questions, we consider securing the following two parts of the system: the routing protocols, used to set up paths through the Internet, and the data-plane mechanisms, used to forward packets along the paths set up by the routing protocols.

1. We start with a *game-theoretic* analysis that shows that even the strongest known secure routing protocol is not sufficient to prevent *selfish* ASes from lying about the paths that data packets take through the network. We then find sufficient conditions that ensure that ASes will not lie. Unfortunately, these conditions are highly unrealistic, and so we conclude that ASes may have an incentive to lie about paths, and thus potentially forward their customer's traffic via paths that drop or corrupt packets.

2. We next consider secure data-plane mechanisms. We use novel *cryptographic and data-streaming* approaches to design lightweight protocols that *detect* packet loss and corruption on a path through the network, even when some nodes on the path are *adversarial.* Our protocols allow a sender and receiver to securely monitor billions of packets using only a few hundred bytes of storage and a pair of comparably sized control packets.

3. Finally, we take the security guarantees above even further, by considering protocols that also *localize* an adversarial node that drops or corrupts packets. We use cryptographic proof techniques to design new protocols and argue that *any* secure localization protocol requires the participation of *every* node on the path. This requirement is considered severe in the setting of interdomain routing, where each node is owned by independent economic entity, with little incentive to participate in the localization protocol.

Our results have implications on the design of high-performance network architectures that can withstand selfish and adversarial behavior.

Acknowledgements

I could not have asked for better advisors than Jennifer Rexford and Boaz Barak. I thank them both for giving me complete freedom throughout my PhD. Jen's sharp intellect, excellent taste in problems, and ability to find exactly the right course of action in any situation has been a continual inspiration to me over the years. My career as a researcher has largely been driven by her unwavering support, mentorship and high standards. I thank Boaz for teaching me to think like a theorist, and for always being ready to give me the technical tools and the support I needed to do my research. Despite of his bent for "hardcore" theory, Boaz was always excited about whatever applied problem I happened to throw in his direction, and his ability to work through complex lines of thought in a matter of minutes never fails to amaze me.

I thank Shai Halevi for his unwavering support and the countless hours he spent reading my papers, working with me on proofs, and teaching me to tackle problems in a systematic way. Shai, Tal Rabin, and the rest of the cryptography group at IBM Research (Ran Canetti, Nelly Fazio, Rosario Gennaro, Craig Gentry, Charanjit Jutla, Jonathan Katz, Hugo Krawczyk, and Vinod Vaikuntanathan) have been invaluable mentors to me. I especially thank Tal for her wise advice and for letting me "squat" in the crypto lab at IBM for so many years.

I thank Maria Klawe for supporting my switch in research areas during my second year at Princeton. Maria kept me in grad school, and I cannot thank her enough for giving me the opportunity to work with fantastic researchers in computer science.

This thesis is based on work done jointly with several fantastic computer science researchers: Boaz Barak, Shai Halevi, Jennifer Rexford, Eran Tromer and David Xiao. I especially thank Dave Xiao for hours spent in the library teaching me to think like a cryptographer, and Michael Schapira, with whom I've never coauthored a paper (yet!), for hours spent on the phone teaching me to think like a game theorist.

As part of the Cabernet group, I was fortunate to be in an environment where there was always someone to question my assumptions, challenge my conclusions, or claim that my adversary models were too strong. I especially thank Yi Wang, Changhoon Kim, Haakon Ringberg, Rui Zhang-Shen and Elliot Karpilovsky for their thoughtful comments on many iterations of papers and practice talks. I spent a fantastic summer at Cisco Research in California, where Fabio Manio, Flavio Bonomi, Syam Appala and David McGrew went above and beyond to teach me about the practical side of network security and software engineering. I benefitted from conversations about life and research with Nadia Heninger, Alexandra Kolla, Eugene Brevdo, Alex Fabrikant, Guy Rothblum, Yaron Singer, Carmit Hazay, Alex Halderman, Ari Feldman, Hoeteck Wee, Bin Li and many others. I'd also like to thank various faculty members for sharing their expertise with me, including Robert Calderbank, Moses Charikar, Nick Feamster, Mike Freedman, Claire Gmachl, Piotr Indyk, Li-Shuan Peh, Leo Reyzin, Adam Smith, Dan Wallach and Jo Kelly. I especially thank Tal Malkin and Erich Nahum for being part of my committee.

I'd like to thank Professor Paul Prucnal for giving me my first taste of research during my early years at Princeton. I am also indebted to Andrea Civelli for his unwavering support during those years.

Finally, I'd like to thank various friends scattered in New York, Princeton, Boston, Toronto, San Francisco and Montreal for supporting me, in various ways, over the past five years. I am especially grateful to my family, my parents Sara and Doron, my brother Danny, and my "sister" Lorie, for being there for me through the ups and downs of graduate school. This thesis is dedicated to my Saba, who probably would have earned a PhD himself if he had had the time.

Bibliographical Notes

The material in Chapter 2 and Appendix A is from a paper that was originally coauthored with Shai Halevi [45], then merged with the work of Aaron Jaggard, Vijay Ramachadran and Rebecca Wright [65], and finally published as [46] at SIGCOMM'08.

Chapter 3 and Appendix B expands and clarifies material that originally appeared in a paper [49] that was coauthored with David Xiao, Eran Tromer, Boaz Barak, and Jennifer Rexford and published at SIGMETRICS'08. Sections 3.3, 3.5, and Appendix B present a number of new ideas that did not appear in [49], and Theorem 3.5.2 and Theorem B.5.4 are corrected versions of Theorem 5 and Theorem 6 from [49].

Chapter 4 and Appendix C is a clearer exposition of ideas that appeared in a paper [16] that was coauthored with David Xiao and Boaz Barak and published at EUROCRYPT'08. While the results of Section 4.3.2 were mentioned in [16], they appear in their entirety for the first time as Theorem 4.3.5.

Chapter 1

Introduction

Today's Internet is a collection of *autonomous systems* (ASes) (*e.g.,* Princeton's campus network, AT&T's global backbone network), each controlled by different profit-seeking businesses, each consisting of a complex network of *routers* and other devices. Connectivity on the Internet requires these competing economic entities to cooperate; communication from a source to a destination can traverse multiple devices inside multiple ASes. Despite these complex relationships, the Internet was originally designed under the assumption that devices inside the network could be trusted; security threats were perceived to come from outside the network. Furthermore, the system is notoriously resistant to change; because the Internet is not controlled by single centralized entity, it is extremely difficult to convince multiple independently-operated ASes to upgrade to a new protocol. As such, many protocols used on the Internet today were designed at a time when it still made sense to assume that all devices in the network can trust each other.

Because the Internet functions in a complex economic environment, its operation is challenged by the presence of *adversarial* or *selfish* parties that choose to deviate from correct operation of network protocols. For example, a profit-seeking Internet service provider (ISP) might misrepresent network performance in order to attract more of traffic from its paying customers. As another example, a router hacked by a malicious outsider may selectively modify traffic from a website like cnn.com, perhaps in order to drive up stock prices. Unfortunately, many of the network protocols used on today's Internet were not designed to deal with these types of malicious or strategic behavior. The thesis contributes to the body of works that seeks to remedy this, by considering the *design and analysis of network protocols that operate correctly even in the presence of adversarial or selfish behavior.*

1.1 The interdomain routing system on the Internet

When we purchase an item from Amazon.com, traditional cryptography prevents attackers from seeing our credit card numbers or impersonating the Amazon website. But how can we ensure that our request actually *arrives* at the Amazon.com server, without being dropped or corrupted along the way? This is exactly the challenge we address in this thesis – improving network *availability*, or improving the chances that packets arrive correctly at their destination.

We focus specifically on availability in the *interdomain* routing system, that enables communication between ASes in the global Internet. We separate the interdomain routing system into two parts: the control plane, *i.e.,* the routing protocols used to establish paths through the Internet, and the data plane, *i.e.,* the mechanisms used to forward packets over the paths set up by the routing protocols. Network protocols and devices handle control-plane (routing) and data-plane (forwarding) mechanisms in different ways; data-plane mechanisms are designed

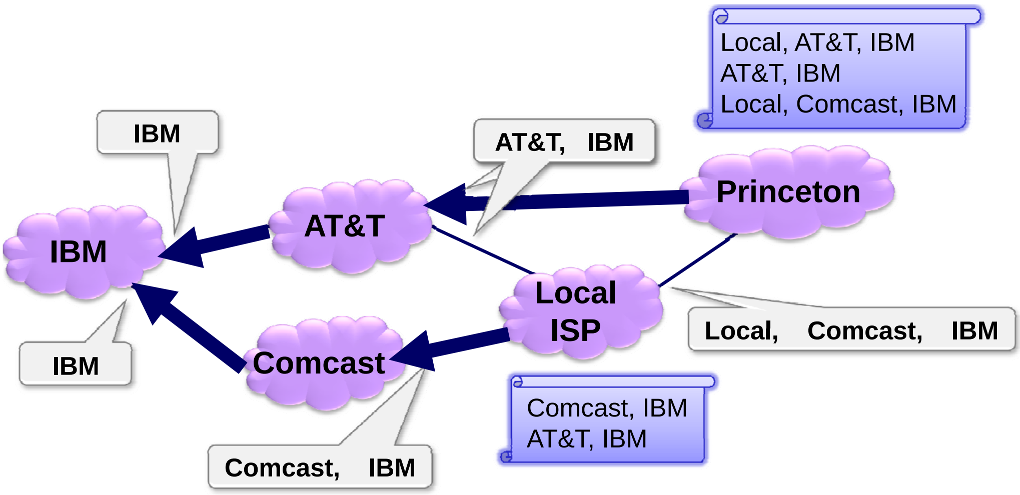

Figure 1.1: A stylized view of the interdomain routing system running BGP.

to be simple and fast, while control-plane mechanisms may be more complex and potentially slower. This separation exists because paths in the Internet typically change as a result of link or node failure, which happens on a much smaller timescale then the timescale used for packet forwarding (*c.f.,* a sensor network, where wireless inference causes paths to change on the same timescale as packet forwarding).

The control plane. The control-plane protocol used in the Internet today is the Border Gateway Protocol (BGP) [92]. BGP allows ASes to discover paths to each destination in the Internet. In BGP, an AS discovers a path to a destination via an *announcement* message that it receives from each of its neighboring ASes. Each announcement contains the AS-level path that the neighbor AS uses to reach that destination. In this thesis, we make the simplifying assumption that each AS selects a *single* path for *all* its traffic to each destination. (In fact, each individual *router* inside the AS selects a single path for each destination, but we ignore this complication in our work.) Path selection is guided by the AS's *routing policies*; these routing policies may depend arbitrarily on commercial, performance, or even security considerations [22].

(In Figure 1.1, we show how BGP announcements allow Comcast, AT&T, Local ISP, and Princeton to discover paths to destination IBM. Routing policies for Local ISP and Princeton are shown inside scrolls. Here, Local ISP prefers the path through Comcast over the path through AT&T, perhaps because Comcast provides service to Local ISP at a lower cost than AT&T. As such, Local ISP routes all traffic destined for IBM over the path through Comcast. As a result, Princeton's most preferred path (Local, AT&T, IBM) is not available, and so Princeton chooses to send traffic over its second favorite path through AT&T.)

The data plane. Once an AS establishes a path to a destination using BGP, the routers inside the AS forward packets along these paths. Because each AS uses BGP to choose a *single* AS-level path to each destination, it follows that packet forwarding from a source to a destination on the Internet typically occurs on a single AS-level path. Even so, packet forwarding can be a non-trivial task; at the core of the Internet, packets must be processed at extremely high speeds (about 2 nsec per packet). To ensure that packet-processing is extremely fast, the data-plane was designed to be quite simple; for instance, it was *not* designed to guarantee that packets will arrive unmodified at their correct destination. As packets travel through the network, congestion at links or nodes can cause packets to be dropped before they arrive at their destination; there is no mechanism that detects or prevents packet loss[1] . Furthermore, packet modification may occur as a result of device malfunction, link failure, or even malicious attack; because packets in the Internet are usually not authenticated cryptographically, the data-plane does *not* guarantee that packet modification is always detected and/or prevented.

1.1.1 Protocols for improving availability

This thesis studies protocols that can be used to improve availability on the Internet; namely, to improve the chances that the network delivers packets correctly. In this work, we will use the term "secure" to mean that a protocol operates correctly even in the presence of certain misbehaviors by parties on the network. We emphasize that our focus is exclusively on protocols that can be used to improve availability; we do not concern ourselves with protecting confidentiality, privacy, or any other issues traditionally associated with "security".

Ultimately, one of our goals will be to understand whether improved guarantees on availability should be architected into the control-plane, the data-plane, or both. As such, we start by surveying a small sampling of security research proposals that deal with availability on the control-plane and the data-plane.

Securing the control plane. BGP was designed under the assumption that all nodes in the network can trust each other. As such, BGP does not have any mechanisms to validate that a path announced by an AS in BGP is actually used for forwarding traffic, or even *exists* in the Internet topology! The networking research community has put together a number of research proposals to remedy this (see [21] for a comprehensive survey).

The most important of these research proposals is "Secure BGP" (S-BGP) [66]. S-BGP guarantees that ASes can only announce paths that actually exist in the Internet by using digital signatures to cryptographically authenticate each BGP announcement message. This ensures that no AS can announce a path to its neighbors unless that path was announced to it by one of its own neighbors. While S-BGP provides the strongest control-plane security guarantees known to date [21], there are still many hurdles that must be overcome before the protocol can be deployed in the Internet. The most significant of these is probably the fact the security properties of the protocol only take effect *after* it has been adopted by a large number of autonomous systems; however, independently operated ASes will only undertake a costly upgrade to S-BGP once its security benefits have taken effect. In spite of this, practitioners are currently working towards a large-scale deployment of S-BGP [2].

Even though S-BGP defends against announcement of paths that do not exist in the Internet topology, S-BGP does *not* guarantee that a path that appears in a BGP announcement message (*i.e.,* in the control plane) is actually being used for forwarding traffic (in the data plane)! To see how, consider Local ISP in Figure 1.1. Because Local ISP learns two different paths from its two neighbors, AT&T and Comcast, Local ISP can easily send an S-BGP announcement to Princeton containing the AT&T path , while actually forwarding all its traffic over the Comcast path!

Securing the data plane. While most of the security efforts of the networking community have focused on the control plane, earlier studies of routing security focused instead on the data-plane mechanisms. These early works (*e.g.,* Radia Perlman's thesis [89] and the work on Secure Message Transmission [32]) focused on designing protocols that *prevent* packet loss and corruption, even in the presence of adversarial nodes in a network. To do this, these protocols encode and transmit message over multiple paths, such that only a some subset of these paths is

[1] Detecting and preventing packet loss is handled by the transport and application layers; this thesis focuses on the network layer.

controlled by the adversary. However, because these protocols require a source and destination to communicate over *multiple* paths, they are unsuitable for today's interdomain routing system where the source and destination communicate over only a *single* path.

When a source and destination may communicate over only a single path, data-plane mechanisms alone *cannot* guarantee that packets arrive correctly at their destination. (To see why, suppose that an adversary on the path decides to drop all traffic from the source. Then, the source has no way of guaranteeing that his traffic arrives at the destination, unless the source switches to a different path. However, here we shall consider path-switching mechanisms to be part of the control-plane, not the data-plane. We make this distinction because we think of data-plane mechanisms as operating at the level of individual packets; path-switching mechanisms typically operate on an aggregate stream of packets, rather than on individual packets themselves.) For this reason, instead of attempting to *prevent* packet loss, many recent works [28,12,33,59,60,95,96,82,33,76,99,86,13,11,81,10] have focused on developing techniques for *detecting* when packet loss occurs on a path. Some works [11,13,86,109,81,10] take this one step further by also localizing the link that is responsible for packet loss. These protocols can then be used to mitigate packet loss if they are used in conjunction with modern control-plane protocols [55,105] that react to packet loss (and other performance issues) by switching to better paths through the network.

1.2 Our Goals

In this thesis, we take a principled approach to analyze the types of security guarantees that are possible within the engineering and economic constraints of the Internet's interdomain routing system. Our ultimate goal is to inform and advance practitioners' efforts to deploy new security protocols in the system. We do this by analyzing two broad themes:

1. **Which part of the system should be secured?** Should we be designing secure protocols for the control plane, the data plane or both?

2. **What is the right tradeoff between security and efficiency?** Ideally, we would like to design protocols that operate correctly even in the presence of very strong adversarial behavior. However, protocols with strong security guarantees can sometimes come with a cost that makes them impractical for real deployment in the interdomain routing system. As in most traditional settings, one important cost that we consider in this work is *system overhead*; namely, the increase in computation, storage and communication incurred by a network device running the security protocol. A less traditional issue that is extremely important in our setting, is the cost of *participation*; namely, the number of parties in the system that must deploy and participate in a protocol before its security guarantee can take effect. Because the Internet lacks a centralized authority that can force ASes on the Internet to adopt a new security protocol, deploying new protocols in the network requires each AS to independently decide upgrade to the protocol. Thus, we a protocol that requires participation from multiple parties comes at a higher cost than one that requires participation from only a small number parties.

1.2.1 Security guarantees and threat models

We formally characterize the types of security guarantees that can be achieved by different parts of the interdomain routing system. We focus exclusively on protocols that can be used to improve availability in the Internet. We shall consider control-plane protocols separately from data-plane

Misbehavior	Description	Technique
Honest	Follows the protocol	-
Benign	Average-case failure	-
Rational	Strategically deviates from protocol to maximize utility	Game Theory
Adversarial	Worst-case failure	Cryptography

Figure 1.2: Misbehaviors considered in this thesis.

protocols, in order to understand the types of security guarantees that can be built into each part of the system. For a given security guarantee, we shall study the conditions (*e.g.,* system overhead, participation, etc.) that are necessary in order to achieve that security guarantee. In many cases, we shall also design new protocols that achieve the security guarantee.

When we say that a protocol provides a certain "security guarantee", we really mean that protocol should function *correctly* in the face of certain *behaviors* or *threats* in the system. We consider well-defined security guarantees: for each, we will specify the notion of correctness (*e.g.,* "detect if more than 1% of all traffic on a path is being dropped"), and clearly define the behaviors of the parties that participate in the protocol (*e.g.,* "the sender and receiver are honest, and there is a single adversarial party on the path between them that can add/drop/modify packets at will"). At this point, we defer explicit statements of each of the security guarantee to individual chapters in this thesis. Instead, we overview, below and in Table 1.2, the general "threat models" or misbehaviors considered in this thesis.

Because the inventors of the Internet assumed that devices inside the network can be trusted, Internet protocols are typically designed to deal with for honest parties and benign failures:

> **Honest behavior.** An honest party always correctly follows the protocol.
>
> **Benign failure.** We will use benign failure as an umbrella terms for "average-case" deviation from the correct behavior of a protocol. Benign failures can include random link cuts or node failures that case parties to stop responding to protocols. Another example of benign failure is when a router randomly drops packets as a result of congestion. Benign failures are caused by parties that are not strategic or malicious.

Because the Internet is now a federated system consisting of multiple ASes owned by independent profit-seeking businesses, there is a high potential for parties to act selfishly/strategically in order to maximize profits or derive benefits for themselves:

> **Rational (selfish) behavior.** A rational party will strategically deviate from a network protocol in order to derive some well-defined benefit for itself. When we think of rational parties, we first define their utility function. Then, we assume that these parties will attempt to maximize their utility, potentially at the expense of deviating for the correct behavior prescribed by a network protocol. In this thesis, we will use emerging game-theoretic techniques, namely, distributed algorithmic mechanism design, to analyze protocol correctness in the presence of rational behavior (see Chapter 2).

Devices on the Internet are also subject to misconfiguration, or malfunction; they can be commandeered by malicious outsiders or be subverted by disgruntled network operators. The most general way to model these types of misbehaviors is to assume that the device is controlled by a malicious adversary.

> **Adversarial (malicious) behavior.** Unlike the rational party, the adversarial party is not characterized by a utility function. This is models of worst-case behavior;

adversarial parties do anything in their power to break the correct operation of the protocol.[2] In this thesis, we will leverage techniques from cryptography to analyze protocol correctness in the presence of adversarial behavior.

The reader might wonder why we bother with the rational model of behavior, when the more general adversarial models are available. We do this for two reasons:

1. While we always prefer protocols that operate correctly even in the presence of very strong adversaries, these protocols often incur unacceptably high costs (*e.g.,* system overhead, participation). Thus, it sometimes makes sense to design protocols that operate correctly in the presence of realistic models of rational behavior, even if we know that these protocols fail in the presence of adversarial behavior.

2. In this work, we shall prove statements of the form: "Security guarantee X is impossible without (system overhead or participation) cost Y". These statements are actually more convincing if we prove them under the assumption that parties in the network are rational, rather than adversarial! To see why, notice that arbitrarily malicious behavior is a superset of rational behavior. As such, if a security guarantee X requires some (system overhead or participation) cost Y even when parties are rational, then cost Y is also required when parties are arbitrarily malicious.

1.3 Our Contributions

Each chapter of this thesis is completely self-contained, with its own introduction, motivation, notation, and conclusion. We now overview the contents and connections between these chapters, and discuss how they relate to the goals of this thesis, as discussed in Section 1.2.

1.3.1 Securing the control plane (Chapter 2)

Our goal is to study network security protocols that can be used improve availability on the Internet's interdomain routing system. With this goal in mind, there are many reasons why it is natural to consider the security of the control-plane protocols (*i.e.,* BGP) that are used to establish paths through the network. Firstly, recall that in BGP, ASes announce the (AS-level) paths that they use to reach each destination in the Internet. Thus, the design of BGP seems to encourage ASes to rely on path announcements as an accurate indication of the paths that packets take through the network. If BGP announcements did indeed accurately reflect the paths that packets take in the data plane, then an AS could rely on BGP announcements to choose a high-performance AS path for its traffic, or to avoid ASes that it perceives to be unreliable or adversarial. Secondly, as we discussed in Section 1.1.1, control-plane protocols operate at a much smaller timescale than data-plane protocols. As such, the system overhead (*i.e.,* communication, computation, storage) incurred by control-plane protocols is typically less costly than that incurred by data-plane protocols.

Thus, in Chapter 2 we explicitly focus on control-plane protocols, and consider the security requirement of ensuring that the paths announced in the control plane protocol (*i.e.,* BGP, S-BGP, etc.) match the AS-level forwarding paths that are used in the data plane. Because this security requirement is quite strong, we investigate whether it can be met in a weaker, but still realistic, 'threat model' where all ASes in the network are assumed to be rational, rather

[2] Of course, in order to formally model adversarial parties, we must define their adversarial powers. See Section 3.2 for one example.

than arbitrarily malicious (see Section 1.2.1). Assuming that ASes are rational allows us to use game-theoretic tools to reason about when ASes have an incentive to send BGP messages that *deliberately misrepresent* the AS-level paths that their traffic takes through the Internet. We use tools from distributed algorithmic mechanism design (DAMD) to look for conditions under which we could prove that ASes have no incentive to send BGP announcements that misrepresent the forwarding paths they use in the data-plane. Earlier attempts within the DAMD framework [39, 73, 35, 36, 37, 38, 84, 87]assumed that the utility of an AS is completely determined by the *outgoing path* its traffic takes to the destination. However, this model of utility fails to capture the fact that many ASes are paid by their customers to carry incoming traffic (*e.g.,* In Figure 1.1, Princeton pays AT&T to carry its traffic.) Thus, for the first time, our work considers ASes with utility functions that also depend on the *incoming traffic* that they attract to their networks.

Our analysis yields some surprising results. We first show that even if we assume that ASes are rational, and even if they all use S-BGP, the strongest known secure routing protocol (Section 1.1.1), then some ASes may still benefit from sending BGP messages that misrepresent the paths that they use for forwarding traffic. We then prove that there do exist certain conditions under which ASes have no incentive to misrepresent their about forwarding paths; however, these conditions require unrealistically strong assumptions on the routing policies of every AS in the Internet.

Thus, the results in Chapter 2 suggest that ASes should not rely on traditional secure routing protocols (like S-BGP [66]) to improve availability by choosing high-performance/trusted paths for their traffic.

1.3.2 Data-plane path-quality monitoring (PQM) (Chapters 3-4)

In Chapters 3-4, we move away from control-plane mechanisms, and focus instead on data-plane mechanisms that can be used to improve availability. Here, instead of taking the more traditional approach of *preventing* packet loss by sending traffic over *multiple* paths, we instead focus on the more realistic *single* path setting. (Recall that with BGP, routers chose a single path for all their traffic to a destination.) We study *path-quality monitoring (PQM)* protocols that run in the data-plane and *monitor* packet loss and corruption on a single path through the Internet. Then, packet loss and corruption can be *prevented* by combining these PQM protocols with control-plane techniques (*e.g.,* intelligent route control, source routing, overlay routing [55]) that give source networks greater flexibility when selecting a path to a destination; if the monitoring protocol indicates that packet loss or corruption on a path is too high, the source can switch to another (better) path. Because we want path-quality monitoring protocols that can be used to inform routing decisions, our goal is to design protocols that can run in high-speed routers. Furthermore, we require these protocols to return correct information, even when adversarial nodes on the path interfere with the monitoring process.

Because our goal is to design PQM protocols that run in the data-plane of high-speed Internet routers, our protocols need to be able to keep up with the high packet-processing speeds and traffic volumes at the core of the Internet. Thus, the question of security *v.s.,* efficiency becomes paramount. Indeed, we argue (informally) that if data-plane monitoring protocols are required to return correct information even when adversarial nodes on the path try to bias monitoring results, then these protocols incur high overheads, related to the amount of traffic sent in the data-plane. To see why, notice that if traffic pertaining to the monitoring protocol can be distinguished from regular data-plane traffic, then the adversary can bias the outcome of monitoring protocol by selectively dropping the regular traffic, while providing good performance for the monitoring traffic. Thus, ensuring that the outputs of the protocol cannot be biased requires us to make PQM-related traffic indistinguishable from regular traffic send on the path. Providing this indistinguishablity introduces system overheads that are roughly proportional to the amount of traffic sent in the data plane.

In Chapter 3 we consider protocols that can *detect* high rates of packet loss/corruption, even in the presence of adversaries. In Chapter 4, we take this security requirement one step further by considering protocols that can also *localize* the (possibly adversarial) link responsible for dropped/corrupted packets. Our major objectives in each of these chapters is to understand the cost (in terms of system overhead and participation, see Section 1.2) of each type of security requirement. Along the way, we also design some interesting detection and localization protocols.

Detecting the adversary.

In Chapter 3 we consider protocols that allow a source to detect high rates of packet loss and corruption on data-plane path.

We start by using simple cryptographic proof techniques (*i.e.,* reductions [50]) to prove that *any* protocol that robustly detects high rates of packet loss and corruption in the presence of adversaries requires that the sender and receiver share secret keys and perform cryptographic operations. We then use cryptographic and data-streaming approaches to design a number of highly-efficient detection protocols. One of our protocols, the "secure sketch", can monitor up to a billion packets without marking normal data-plane traffic, and using only two control messages and 200-600 bytes of storage at the source and destination only. (Asymptotically, monitoring T packets requires $O(\log T)$-storage, and two control messages.) We prove that all our protocols satisfy a precise definition of security, and derive analytic expressions for the tradeoff between statistical (measurement) accuracy and storage overhead for each protocol.

The results of Chapter 3 are encouraging; by focusing on the modest security requirement of detecting packet loss/corruption, we are able to design highly-efficient protocols that can withstand very strong adversaries. Furthermore, all of our protocols require the participation of the source and destination only; no other node on the path is required to participate.

Localizing the adversary.

While it is useful to enable sources to detect packet loss and corruption on path, it is even more useful to be able to localize the adversarial node responsible for tampering with packets. Thus, in Chapter 4, we use a similar adversarial model to study a *stronger* security requirement; namely that a source can localize the link that is responsible for high packet loss or corruption.

We start by developing a formal cryptographic model of security for the localization problem, and use this formal model to find security vulnerabilities in previously published works [86,13,10]. We then present a number of localization protocols. One of our protocols can monitor T packets using $O(\log T)$-storage per node, two additional control messages, and shared keys between the source node and every other node on the path. While the detection protocols of Chapter 3 require participation from the source and destination only, all known localization protocols *e.g.,* [11,13,86,109,81,10], including the ones we design in Chapter 4, require participation from *every* node on the path. It is natural to ask if these high levels of participation are necessary. We answer this question in the affirmative by leveraging cryptographic proof techniques (black box separations [62]) to argue that *any* protocol that correctly localizes links responsible for packet loss and corruption in the presence of adversaries, requires *every node on the path* to share secret keys with the source, and perform cryptographic operations.

Thus, the results of Chapter 4 suggest that security requirement of localizing links responsible for packet loss/corruption might be too ambitious for the interdomain routing system; we may be better off with the more efficient protocols that only *detect* packet loss/corruption, as designed in Chapter 3.

1.4 Conclusions, Implications and Future Directions

In Section 1.2, we mentioned that the goals of this work are to understand which parts of the interdomain routing system should be secured, and to study the tradeoffs between security and efficiency. We now discuss how our results and several new research directions can begin to address these goals. We also overview the implications of our work on the design of network architectures that guarantee availability in the presence of selfish or adversarial behavior.

1.4.1 Which part of the system should be secured?

Should we be designing secure protocols for the control plane, the data plane, or both?

Securing the control plane is not a panacea. Our results in Chapter 2 suggest that availability will still be a challenge even if the strongest known secure routing protocol (S-BGP) is fully deployed in the Internet. We showed that, even if we assume that all ASes in the network use S-BGP, and are *rational* (rather than *adversarial*), ASes still have an incentive to announce AS-level paths in the control plane that do not match the paths actually used in the data-plane. Thus, our analysis shows that it is unreasonable to assume that an AS can rely on BGP messages to choose paths that circumvent routing traffic through untrusted or adversarial ASes.

It is interesting to note that our analysis in Chapter 2 is a *worst-case* analysis. We show that if ASes are rational and use S-BGP, then *there exist network topologies* where at least one AS has an incentive to send a BGP announcement that misrepresents the path he uses in the data plane. To better understand the practical relevance of the results in Chapter 2, we would also like to answer the following questions: Firstly, how often do such network topologies (where ASes have an incentive to lie) appear in practice – do they only exist in the obscure corners of the Internet, or are they extremely prevalent? Secondly, how effective is S-BGP in reducing the number of ASes with an incentive to misrepresent their paths – how many more ASes can get away with lying if we assume that ASes use plain BGP, as compared to S-BGP or some other secure routing protocol [21]?

We are in the process of conducting an empirical study of the Internet's topology that seeks to answer some of these questions. Preliminary results suggest that even if all ASes in the Internet use S-BGP, many ASes will still have an incentive to lie in their BGP announcements.

Strong security guarantees are possible in the data-plane. Our results suggest that it is feasible to design secure protocols that are efficient enough to run in the data-plane of high speed routers, especially if we consider protocols that only require the participation of a source and destination. Indeed, we were able to design highly efficient path-quality monitoring (PQM) protocols that operate correctly even in the presence of a very strong adversary that knows the details of the monitoring protocol and can add/drop/modify traffic at will.

While this thesis focused on PQM protocols that monitor packet loss and corruption, there are many other metrics that can determine path quality, including traffic latency (delay), jitter (delay variance), and packet lag (the number of packets that arrive out-of-order at the destination). We believe that designing efficient and secure PQM protocols for these metrics is a worthwhile direction for future work.

Hop-by-hop protocols vs. end-to-end protocols. On one hand, we can design 'end-to-end' security protocols do not require knowledge of the *identities* of the nodes on the path between the source and destination, like the detection protocols of Chapter 3. On the other hand, we can design 'hop-by-hop' protocols that require the sender to know the identities of the nodes on a path, like the localization protocols of Chapter 4. However, our results indicate that (a) control-plane protocols like BGP and S-BGP do not always accurately return information about the identities of the ASes on a data-plane path, and (b) data-plane protocols that localize an adversary are expensive, because each node on the path has to participate. Taken together, these results suggest that hop-by-hop protocols are impractical; indeed, such protocols are likely useful only in limited settings where the need for security is so strong that it overwhelms such practical concerns.

We believe that promising direction for future research is to analyze other security functionalities that can be realized in an end-to-end manner. For instance, certain control-plane path-switching protocols (*e.g.,* multipath routing, overlay routing [55]) can be realized in an end-to-end manner. However, more work is required to characterize the security guarantees that can be achieved by these path-switching protocols, especially when they are combined with end-to-end PQM protocols as discussed in Chapter 3.

1.4.2 Security versus efficiency

As we discussed in Section 1.2.1, our notion of a "security guarantee" for a protocol has two parts: a notion of "correctness", and a "threat model". We would like our protocol to operate correctly even in the presence of parties that behave (and misbehave) in the ways specified by the threat model. Ideally, we would like to design protocols that provide strong security guarantees; however, these protocols often come at high cost, either in the form of system overhead (*e.g.,* computation, storage, communication resources) or participation (*i.e.,* many nodes in the network must deploy the protocol, so that deploying these protocols in the Internet becomes a challenge, see Section 1.2). In order to understand which security guarantees are feasible within the engineering and economic constraints of the Internet's routing system, we studied ways to tradeoff between strong security guarantees and protocol cost. One way to do this is to consider weaker notions of protocol correctness; another is to consider weaker threat models. Indeed, this thesis takes both of these approaches:

Weaker notions of correctness. Our study of path-quality monitoring in both Chapters 3-4 considered the following 'threat model': a source and destination trust each other, while an adversary that drops and corrupts traffic occupies any subset of the nodes on the path between them. The ideal notion of correctness in this setting would be to empower the source to *localize* the adversarial nodes; however, in Chapter 4 we show that achieving this notion of correctness comes at the unacceptably high cost of requiring all the nodes on the path to participate in the protocol. Thus, in Chapter 3 we show that a weaker notion of correctness, *i.e.,* empowering the source to *detect* when packets are lost/corrupted, comes at a much more reasonable cost, *i.e.,* participation by the source and destination only.

Indeed, we believe that analyzing a spectrum of notions of correctness is a useful exercise, especially when architecting networks with practical and useful security guarantees. We believe that a number of other problems in network security could benefit from this approach.

Weaker threat models. Now consider the following notion of correctness: ensuring that ASes send BGP messages that accurately reflect that AS-level paths that they use in the data plane. Viewing our results in Chapter 2 in the broader context of the work on distributed algorithmic mechanism design and BGP, we see that this notion of correctness has been studied for a variety of different 'threat models'. For instance, Levin, Schapira and Zohar [73] show that full deployment of S-BGP is a sufficient condition for this notion of correctness, as long as ASes are modeled as rational with utility that depends only on the *outgoing path* that they use for their traffic. However, once we consider a stronger threat model, where ASes' utility also depends the *incoming traffic* routed through their network, our results in Chapter 2 show that S-BGP alone is no longer sufficient; we also need to (unrealistically) constrain the set of allowable routing policies. Finally, if we assume ASes are adversarial, even constraining the set of allowed routing protocols and requiring nodes to use S-BGP is insufficient for this notion of correctness.

While it is not surprising that this notion of correctness becomes increasingly difficult to achieve as the threat model becomes stronger, it is interesting to note that these results are extremely *sensitive* to the strength of threat model. This observation suggests that we must be very careful in extrapolating from positive results obtained in a weak threat model (*i.e.,* statements of the form: condition X guarantees Y notion of correctness for threat model Z) to the real world. Indeed, this is likely one of the reasons for the success of strong cryptographic threat models, in which parties are assumed to be arbitrarily malicious. On the other hand, we do view weak threat models as a useful tool for proving very convincing negative results (*i.e.,* statements of the form: notion of correctness Y for threat model Z cannot be achieved without condition X). For instance, our results in Chapter 2 show that S-BGP is not sufficient for matching the control- and data-plane *even if ASes obey a realistic, well-defined notion of rationality*; it immediately follows that S-BGP will not guarantee that the control- and data-plane match when ASes are adversarial.

1.4.3 Implications on network architecture

To summarize our discussion, we discuss the implications of our work on the design of networks that can withstand selfish or adversarial behavior, and present a number of other open questions.

Firstly, we believe that any solution that purports to improve availability must include some data-plane security component; indeed, our results in Chapter 2 suggest that even if we assume ASes are rational, control-plane security protocols are not sufficient to ensure ASes do not misbehave in the data plane.

Secondly, we believe that the most promising direction for improving availability in the setting of interdomain routing is focus on protocols that take an end-to-end view of the network; in particular, we advocate for combining intelligent route control protocols [55,105] with the end-to-end PQM protocols proposed in this thesis.

Thirdly, while this thesis suggests that securing the control plane is not a panacea, we do believe that control-plane security protocols have an important role to play in making the interdomain routing system more predictable and robust. However, it is unclear which of the many of proposed control-plane security protocols [21] are 'right' for the interdomain. As such, we believe that it would be valuable to have further studies *comparing* the deployability and security guarantees provided by each of these protocols.

Finally, another interesting direction (that we did not investigate here) is the question of accountability and contracts in the Internet. Because ASes are controlled by profit-seeking businesses, it may be possible to enforce 'good behavior' in the interdomain routing system by designing a system of contracts that penalizes ASes that perform poorly, *e.g.,* by dropping or corrupting packets. While there have been a number of interesting works in this direction [10, 69, 74, 40, 26], many of these results assume that the existence of hop-by-hop secure PQM protocols that we showed to be impractical (Chapter 4). As such, we believe that the question of designing a practical accountability system for the Internet, that uses only end-to-end security protocols, remains open for future research.