A Theoretical Taxonomy and Analysis of Anti-Spam Technologies

A Thesis presented

by

Charles Duan

to

Computer Science

in partial fulfillment of the honors requirements

for the degree of

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

April 6, 2004

A Theoretical Taxonomy and Analysis of Anti-Spam Technologies

Charles Duan
cduan@fas.harvard.edu

A Thesis Presented to the
Department of Computer Science
of Harvard College

April 6, 2004

Preface

This thesis began life just over a month from its final submission date of April 6, 2004, in the form of an abstract the size of an 80-column, 24-row terminal screen, written directly in an e-mail client to be sent to my potential faculty reviewers. Three days later the first draft was complete, weighing in at nineteen pages, or about six thousand words. Since then it has accumulated reference material, grown several theorems, and acquired a total of 19,040 words—an expansion of about 217%.

I have never written anything upwards of about twenty pages, so I cannot take personal credit for about 70% of the text that is present before your eyes. Both the reader and the author, then, might ask the same perplexing question: where did that 70% come from?

The answer, of course, is that it came from the multitude of people who have aided and supported me in this endeavor. **Prof. Mema Roussopoulos**, my advisor for this thesis, has by far contributed the most to this end: her assistance has included reading about five drafts of this paper, front to back, and taking a conference call with me *while she was on vacation.*

In addition to her, numerous people have kindly contributed many thoughtful and useful comments and ideas:

Christina Anderson proofread the entire document.

Daniel Chia made several suggestions on the introduction, including the idea of the "empirical" approach.

Chris Conlon and **Hassan Sultan** gave me a crash course in information economics, and Hassan read an early version of Chapter 4.

Jeff Fogel was asked to read the introduction, and he made several useful suggestions. Jeff Fogel's father was *not* asked to read the introduction, but he did so anyway and also made several useful suggestions.

Mai-Anh Huynh also read the introduction and gave me several useful ideas.

Aaron Lee read several drafts of Chapters 4 and 5; he encouraged me to essentially rewrite the proof in Section 4.1.

Lisa Leung suggested that I add many of the statistics in the introduction.

Yuen-Jong Liu was told to only read Chapters 3 and 4, but he read the entire thing anyway—twice.

Jacob Russell inadvertently suggested the concept of rules.

Prof. Stuart Shieber provided some useful thoughts and ideas, one of which is actually discussed in Section 4.5.

Prof. Alison Simmons and **Thomas Teufel** were my instructor and teaching fellow for an introductory philosophy course; that course has influenced both the style and, in some cases, the substance of my arguments.

Mildred Yuan proofread every section of text that I handed her, totaling about half of the thesis.

I am indebted to all of them for their numerous contributions.

Although they did not contribute to the ostensible writing, I am grateful to my family and friends who have been supportive of this endeavor, which has several times put my health, alertness, and sanity at slight risk; their presence allowed this work to come about, and their presence allowed this work to come to fruition.

Finally, in the tradition of my father, I acknowledge *Homo Sapiens*, whose continuing success has caused me to exist.

Contents

List of Figures

Don’t send large amounts of unsolicited information to people.

—S. Hambridge, *Netiquette Guidelines*, RFC 1855 (Oct. 1995)

Abstract

The problem of unsolicited, unwanted e-mail messages, otherwise known as spam, is growing to uncontrollable proportions, and there is a clear need for directed research in spam prevention and control. Current research has identified many different schemes for controlling spam, but the general direction of this research has been in improving individual schemes without consideration for the various strengths and weaknesses of those schemes as compared to others. The goal of this paper is to identify and categorize the various anti-spam technologies. This categorization will differ from more common ad-hoc anti-spam taxonomies in that it will attempt to cover the entire theoretical space of anti-spam solutions rather than merely those that are implemented or those that are popular. Using such a taxonomy, then, we can investigate the relative strengths and merits of each category. We will demonstrate that many commonly used solutions, such as content filtering, public-key authentication, and computational-cost based prevention, are vulnerable to various forms of attacks and thus ineffective in this problem space. Some of the key conclusions are that a system successful in preventing spam requires sender intervention, and that intervention must be in the form of either identity verification or payment of some cost. This research will aid future studies in anti-spam technology, steering them to greater efficacy.

Chapter 1

Introduction

In a wired world increasingly driven by computer technology, electronic messaging has become the norm for speedy, effective communication, especially in the form of e-mail. Advances in networking have only improved speed, but e-mail users are increasingly entangled in a battle of effectiveness, a battle waged against an army of those who see a different use for e-mail and electronic messaging: sending large volumes of unsolicited bulk mail in hopes of duping a few of the recipients into aiding the sender with some ulterior motive. These unsolicited, unwanted e-mails, otherwise known as “spam,” have saturated the volume of electronic mail with get-rich-quick schemes, advertising, and incomprehensibility to various ends, such as making money and carrying out distributed network attacks.

The voluminous presence of spam in electronic messaging is much more harmful to e-mail than its namesake canned meat is to one’s health. The proliferation of spam threatens to do away with the effectiveness of e-mail, forcing readers to wade through piles of junk to find the important and relevant messages. Recent studies show that spam cost about $9 billion to US corporations in 2003, and the volume of spam being sent is growing at a rate of 20% each month, with millions of spam messages being delivered *every day* [5]. Current popular solutions are ineffective: a 2003 report reveals that 37% of individuals use spam filters, but they receive practically as much spam as those without filters [14]. Economist George Akerlof's seminal paper on information economics reveals another reason why the proliferation of spam is harmful:

> The presence of people in the market who are willing to offer inferior goods tends to drive the market out of existence. . . . It is this possibility that represents the major costs of dishonesty—for dishonest dealings tend to drive honest dealings out of the market. [2]

If spam continues to grow at the current rates, then it is inevitable that dishonest e-mails will soon drive honest ones out of the system, rendering e-mail and other forms of electronic messaging unusable.

It is not surprising that many have taken major initiatives to subdue spam. Some have called for legal restrictions; others are searching for, rooting out, and punishing the senders. The majority, however, believes that this problem can best be—or only be—solved using advances in computing technology and innovation [41]. It is these technological approaches that form the substantive material of this paper.

The field of spam research is interesting due to the diverse, multidisciplinary nature of the technological approaches being researched. Artificial intelligence has provided message filters based on content analysis. Systems research has contributed new protocols and distributed system analyses. Cryptographic algorithms are yet another subfield proposed for dealing with spam. And certainly other fields have contributed or will contribute new ideas. All sorts of solutions have been proposed, ranging from simple filters to re-engineering the entire e-mail system, but these solutions have generally been developed and evaluated independently, improving on proposals within a subfield without much consideration for unrelated solutions and the comparative advantages of each. There is a clear need for a comprehensive look at these different solutions and a comparative analysis of them.

1.1 The Empirical Approach and its Problems

There have been a number of attempts to look at the range of anti-spam technologies and their comparative strengths. One comprehensive published taxonomy was made by Paul Judge for the opening meeting of the Anti-Spam Research Group, or ASRG [3, 25]. Other compendia of anti-spam technologies include a paper by Shane Hird and a presentation by John Levine to the Federal Trade Commission [20, 26]. All three of these are similar in content, and we will focus on Hird's research only because it is the only one presented in the form of a complete paper (the other two are only available in presentation slides).

Hird's paper essentially presents an overview of the state of the art in anti-spam technology. He discusses systems that are currently in use and those that have been proposed but not used, many of which will be discussed throughout this paper as well. Hird takes an *empirical* approach to analyzing anti-spam technology: the analysis is conducted by considering existing solutions, grouping similar ones into categories, and analyzing the features of those categories based on the features of the existing solutions.

There are a number of problems with the empirical approach, all of which are exhibited by Hird's paper. First, the analysis of existing systems tends to emphasize the difficulty of implementing systems. For example, Hird's discussion of the Internet Mail 2000 system focuses on the fact that it would be difficult for the system to be globally adopted and implemented. Section 1.3 presents an argument for why the focus should *not* be on implementation difficulty. Second, the empirical approach fails to address advances in technology. For example, Hird dismisses "computational stamp" systems because computers differ greatly in CPU speed.[1] However, less than a year after Hird's paper was published, two research groups presented a form of computational stamp that is only slightly affected by CPU speed [1, 10]. Finally, empirical analysis cannot address systems that have not been proposed or implemented, so it can draw no broad conclusions on anti-spam research, and the scope of the analysis is inherently limited.

1.2 The Theoretical Approach

Because of the problems inherent in the empirical approach to understanding anti-spam systems, it seems reasonable to take a *theoretical* approach, one that analyzes the design space of *all possible* anti-spam technologies, independent of specific solutions within that space that have been implemented or proposed. This is the approach taken in this paper. We will not see comparisons of products of implementations; rather, we will explore the world of possibilities and the comparative advantages of each possibility. We will ensure that all possible solutions are covered by employing strong proofs to divide the solution space into categories, and then we will consider the properties inherent to each category.

This theoretical approach is valuable for three reasons. First, because the space of all possible solutions is considered, the resulting conclusions are generalizable even to solutions that have not yet been proposed. Second, a theoretical analysis of systems allows us to

[1]Computational stamps are discussed in Section 4.2. At this point, though, the reader unfamiliar with this anti-spam technology needs to know only that such systems require the sender to perform a difficult computation in order to send a message.

deduce properties that must be true for any implementation of those systems, regardless of how advanced that implementation might be. Third, a theoretical approach can provide a measure of the maximal efficacy of each type of system, so we have a foundation for making comparisons of *types* of systems rather than of individual systems themselves and their relative maturity levels.

1.3 A Theoretical Taxonomy and Analysis

This paper will present two distinct but related pieces of work. First, it will present a taxonomy of anti-spam technologies. An overview of that taxonomy is given in Figure 1.1 (at the end of the chapter). This taxonomy will consider the entire design space of anti-spam technologies and divide it into a hierarchy of distinct categories. Second, this paper will present an analysis of each category, identifying the comparative advantages of each.

To construct the taxonomy, the following approach is employed. Given a category of solutions, a *separating test* is used to divide every possible solution within that category into one of two subcategories. For example, the super-category of all possible anti-spam systems is divided into systems that can function entirely without the sender performing some intervening action and those that cannot function without intervention.

Several factors influenced the selection of these tests, some of which are listed below:

- The categories produced by the test should have analytical properties that are as different as possible. The analysis is the most useful when the different categories have different properties.

- The test should *not* be influenced by the number of *existing* solutions within a category. For example, there are many different forms of spam detection based on e-mail content, but they are all placed into one category because of their similar properties.

- The tests should tend to single out unique systems. Such systems represent unexplored categories with distinctive properties and so they should be analyzed in the context of those properties.

In general, the separating test is chosen by the analytical power of the resulting categories. This is not surprising, as the purpose of the taxonomy is to produce an interesting analysis.

The purpose of the analysis is to identify features of the taxonomic categories of solutions. For each category, a number of questions will be considered:

- What is the underlying assumed definition of spam? How closely does that definition match what we intuitively think of as spam?

- What types of legitimate messages would this system mark as spam (Type I errors)? What types of spam messages would this system mark as legitimate (Type II errors)?

- In what areas does this system succeed? A system succeeds when legitimate messages are correctly marked as legitimate and similar-looking spam is correctly marked as spam; for what sorts of legitimate messages is this the case with the given system?

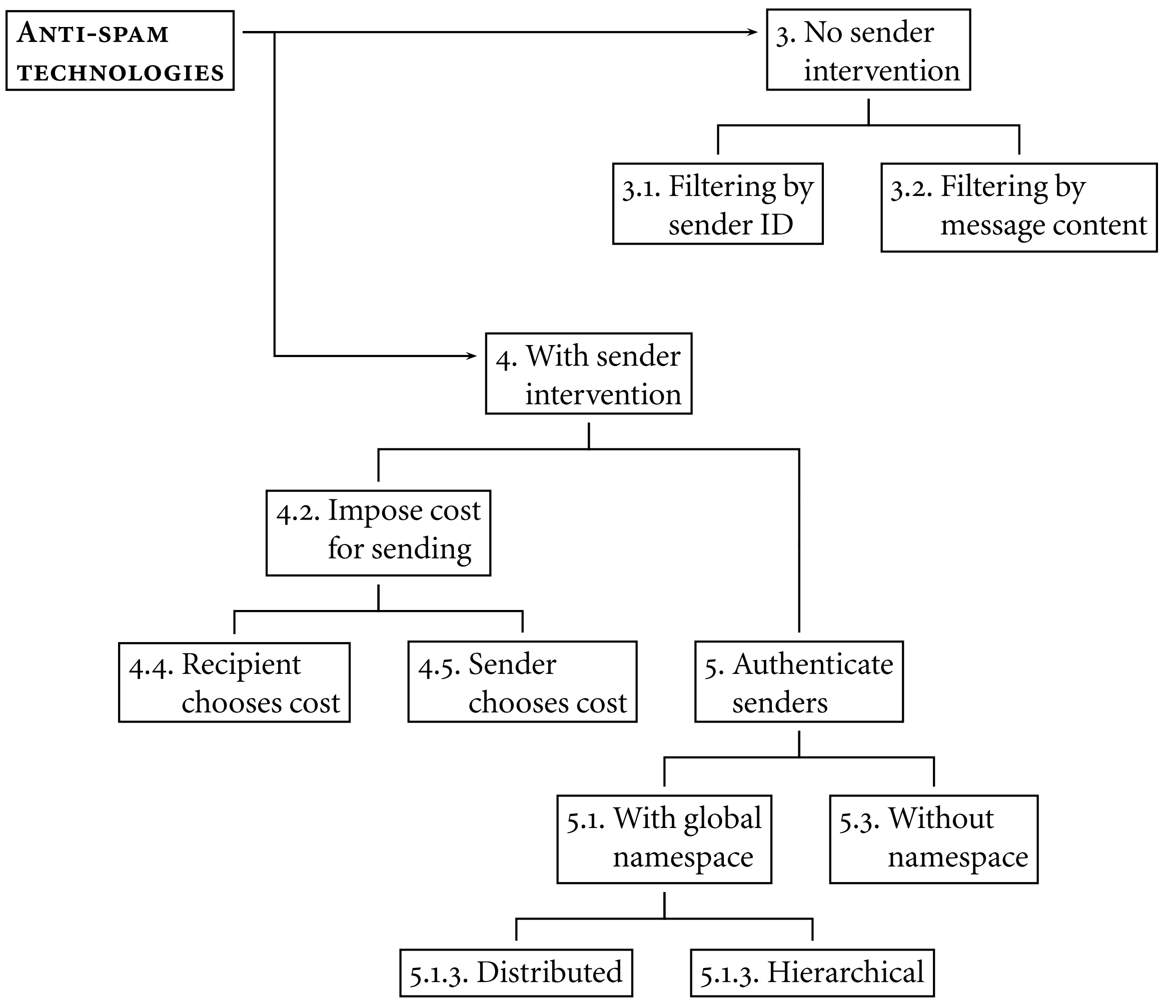
- How would an adversary thwart the system in order to deliver spam? Is there an attack that would allow someone to send spam unrestrictedly, and what resources would be needed to carry out such an attack?

Answering these questions provides insight into the usefulness and the limits of a given system. These questions also provide a uniform basis for comparing categories of systems.

For each category of systems that we analyze, the results of that analysis are summarized in a box in the appropriate section.

Because of the theoretical nature of these analyses, certain questions will *not* be addressed, primarily those concerning the initial implementation of systems. In particular, many of these systems may require large-scale deployment efforts; the required work in deployment is not considered here. The rationale behind not considering implementation difficulties is that such difficulties are irrelevant when considering the effectiveness of the system once implemented. If the world is to adopt a standard system of preventing spam, then that decision should be primarily based on the effectiveness of the system when it will be in use, and the difficulty of implementing that system does not affect that effectiveness.

FIGURE 1.1 Diagram of the taxonomy of anti-spam technologies. Section numbers precede each category, denoting the section discussing that category.


ANTI-SPAM TECHNOLOGIES
3. No sender intervention
3.1. Filtering by sender ID
3.2. Filtering by message content
4. With sender intervention
4.2. Impose cost for sending
4.4. Recipient chooses cost
4.5. Sender chooses cost
5. Authenticate senders
5.1. With global namespace
5.3. Without namespace
5.1.3. Distributed
5.1.3. Hierarchical

Chapter 2

Characterizing the Situation

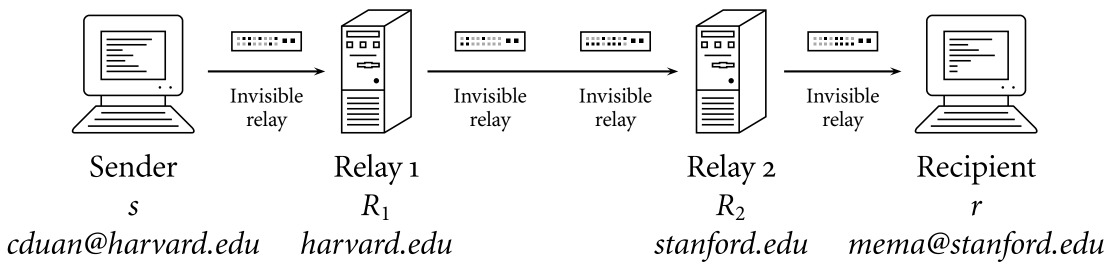
In order to begin analyzing anti-spam solutions from a theoretical context, we must first provide a generalized framework for conducting this analysis. We first propose a model for an electronic messaging system, and using that model we provide a definition of an anti-spam technology. Additionally, since the analyses will focus on the efficacy of systems in separating spam from legitimate mail, a number of guidelines are provided outlining qualities of messages, both legitimate and spam.

2.1 A Generalized Distributed Messaging System Model

In order to form a generalized model of an anti-spam system, we first present a generalized model of a distributed messaging system on which that technology will operate. While this model is based on the e-mail model, it should also apply to other distributed messaging systems such as Internet Relay Chat.

The world of e-mail consists of *entities*, which may be of two types: humans and computers. At any given time, an entity may be immediately able to communicate with others on

FIGURE 2.1 Example of how a messaging system might use visible and invisible relays.



the network; in that case the entity is considered *available*. Entities, in particular humans, are most likely not always available. The process of sending a message is called a *transaction*; the entity sending the message is called the *sender* and the entity receiving the message the *recipient*. The computer from which the message is initially sent is called the *sending computer*; since the sender may be a human, the sender is distinct from—and possibly unrelated to—the sending computer.

When a sending computer wishes to send a message, it might not make a direct connection to the recipient's computer; rather, the message may jump through a number of computers called *relay computers*. There are two possible types of relays: invisible relays and visible relays. Invisible relays simply route network traffic; they will be treated as just part of the infrastructure of the network and ignored. A visible relay is one that accepts messages on behalf of the recipient, possibly performs some processing on the message such as modifying the message header or batching messages with the same destination together, and then forwards the message to the recipient (or another visible relay).

Figure 2.1 gives an example of how visible and invisible relays might be used on a messaging network. The example is modeled after the current e-mail protocols. Say the sender s is the author, *cduan@eecs.harvard.edu*, and the recipient r is a distinguished professor of computer science, *mema@cs.stanford.edu*. The sender s would first send the message to an appropriate SMTP mail server, *smtp.eecs.harvard.edu*, or R_1 in the diagram. This mail server would then queue and send the message to the recipient's mail server, *cs.stanford.edu*, denoted by R_2. Finally, R_2 would store the message and eventually deliver it to the recipient. In transmitting the message from s to R_1 to R_2 to r, the network transmission may have involved intermediate invisible relays such as routers and switches, but those invisible relays only treated the messages as network traffic. The visible relays R_1 and R_2, however, inspected the message's content and performed processing based on that content. In terms of the OSI network model, invisible relays operate on the data, network, and transport layers; visible relays operate on the application and presentation layers [24].

Visible relays complicate the messaging system. For example, since a message may pass through and be altered by one or more relay computers, it is difficult to identify the actual origin of a message. Nevertheless, visible relays are useful components of a messaging system for several reasons. First, relay computers may have faster network links and more processing power than individuals' computers, so messages sent through relay computers can be sent much more efficiently than messages sent directly between individuals with slower computers. Also, relay computers can batch several messages bound for the same location into one transaction, saving network bandwidth. Additionally, the recipient's computer may not be available at sending time and the sender's computer may not be constantly available either. A relay computer can retain the message until the recipient becomes available.

Because of these beneficial attributes of relay computers in a messaging system, it seems unreasonable for an anti-spam technology to eliminate them entirely, even though they may exacerbate the spam problem by introducing complexity into the system.

2.2 Definition of an Anti-Spam Technology

Given our formal model of a distributed messaging system, we define an anti-spam technology, or AST, in this paper as follows:

> An AST is any system that, when overlaid on a messaging system, provides recipients with a mapping of received messages to values, such that those values have semantic meaning to the recipient with regard to whether or not the corresponding messages are spam.

The simplest AST might mark messages with one of two values, "spam" or "not-spam." A more complex system might offer a value between zero and one indicating the probability that a message is spam. A third system might divide messages into several discrete categories based on the trustworthiness of the sender, such as "trusted sender," "partially trusted sender," "untrusted sender," or "unknown sender." This last set of values does not explicitly mark the message as spam or legitimate, but the values are useful to a recipient in making that determination (e.g., mail from trusted senders is most likely legitimate).

It is worth considering the goal of an optimal AST so that we can have a perspective on how to judge the types of systems that we will encounter. There are two possible goals: complete elimination and control of spam. Since spam is just a type of e-mail message, a system that allows messages to be sent *must* allow spam to be sent to some degree, so complete elimination is clearly an impossible task. The goal of an optimal AST, then, must be *control* of spam: it may be possible for spam to be sent, and some may even be read by recipients, but the system should halt the *unrestricted* proliferation of spam, so that the majority of messages that consume the time of e-mail readers are legitimate. It should not be *impossible* for spam to be sent, but the system should make it substantially difficult.

Finally, a current "solution" to the spam problem that does not fit into the definition of an AST is the practice of keeping e-mail addresses secret or obfuscating them in public listings, or the use of a disposable address service to hide one's actual address [31]. Solutions like these deal with the spam problem by "hiding" from it: recipients are forced to obscure their identities so that spammers cannot find them. However, a key advantage of e-mail is the ability to easily contact anyone; obfuscating or hiding addresses diminishes this advantage. So solving the spam problem by hiding e-mail addresses would eliminate a disadvantage by eliminating an advantage. And even so, reliance on long-term secrets is not a sustainable practice: one's address *must* be given out to others if that address is ever used, and once the address reaches the eyes of a spammer, it is no longer protected thereafter. A successful AST should limit the ability of spammers regardless of the knowledge or resources they have.

2.3 Common Uses of Messaging Systems

E-mail and other messaging systems today are commonly used in many different ways. It is worth considering some of those uses so that we can determine which legitimate uses of e-mail would be limited or prevented by a given AST.

The most common use of e-mail is personal communication: individuals send messages to one or a few recipients. It is clear that no AST should block such messages, and it would seem that no reasonable AST *would* block them, but there is an unusual aspect of personal messages: they may originate from essentially any computer on the network. Consequently, filtering methods based on network addresses may inhibit this use of e-mail.

A second common use of e-mail is to distribute information to a large body of recipients. Mailing lists are a common example of this: hundreds or thousands of people, randomly dispersed by geography, e-mail provider, and other factors, all receive messages from one sender. The recipients need not all see the exact same message. For instance, a credit card company may issue statements to its customers by e-mail; each of the messages should be personalized to the recipient. The ability to send large volumes of messages via electronic messaging is valuable, and an AST should not block such legitimate bulk messages.

A third common use of e-mail is "sender delegation," in which some entity, usually a computer, is authorized to send one or more messages on behalf of another entity, usually a human. One example is a mailing list, in which a sender provides the mailing list server with a message to distribute to the list recipients. Another example is online greeting card services, in which the sender actually never encounters the messaging system itself but rather uses an external interface, in this case the service's web page. In both cases, the service delivers the message (the list posting or the greeting card) to the recipient on behalf of the sender. If an AST requires the sender of a message to perform some action, then in cases of sender delegation it is necessary to consider who should perform that action.

2.4 The Nature of Spam

The final component of our messaging system to understand is the nature of spam on that system. In this section we will consider first the defining characteristics of spam messages and then the powers of an adversary who sends spam.

There are two commonly used definitions of spam. The first is that spam is *Unsolicited Commercial E-mail*, or UCE; this is the definition the Federal Trade Commission uses [16]. Second, spam is defined as *Unsolicited Bulk E-mail*, or UBE; this term is preferred by academic research [20]. Generally one of these two terms is used in formal dialogue rather than the colloquial term "spam."

Neither of these terms is used in this paper, because the first is inaccurate and the second insufficient. There are clear uses for sending spam other than commercial advertising, such as distributing a virus [13]. There are legitimate uses for bulk mail, and there are situations in which it may be unsolicited, the classic example being the long-lost high school buddy trying to contact friends. But, most importantly, underlying both of these definitions of spam—and underlying *any* definition of spam—is the assumption that there is an absolute definition of spam. This is simply not the case, as every person has a different concept of what mail is spam and what mail is legitimate. As a result, the definition of spam employed in this paper is as follows:

> If a message is sent to a recipient *r*, then that message is *spam* if and only if *r* deems that message to be spam. A message is *legitimate* if and only if it is not spam; that is, it is legitimate if and only if *r* deems it to be not spam.

Spam, then, cannot exist unless there is a recipient to read it, and one recipient's spam may be another's multi-million dollar treasure in an offshore bank.

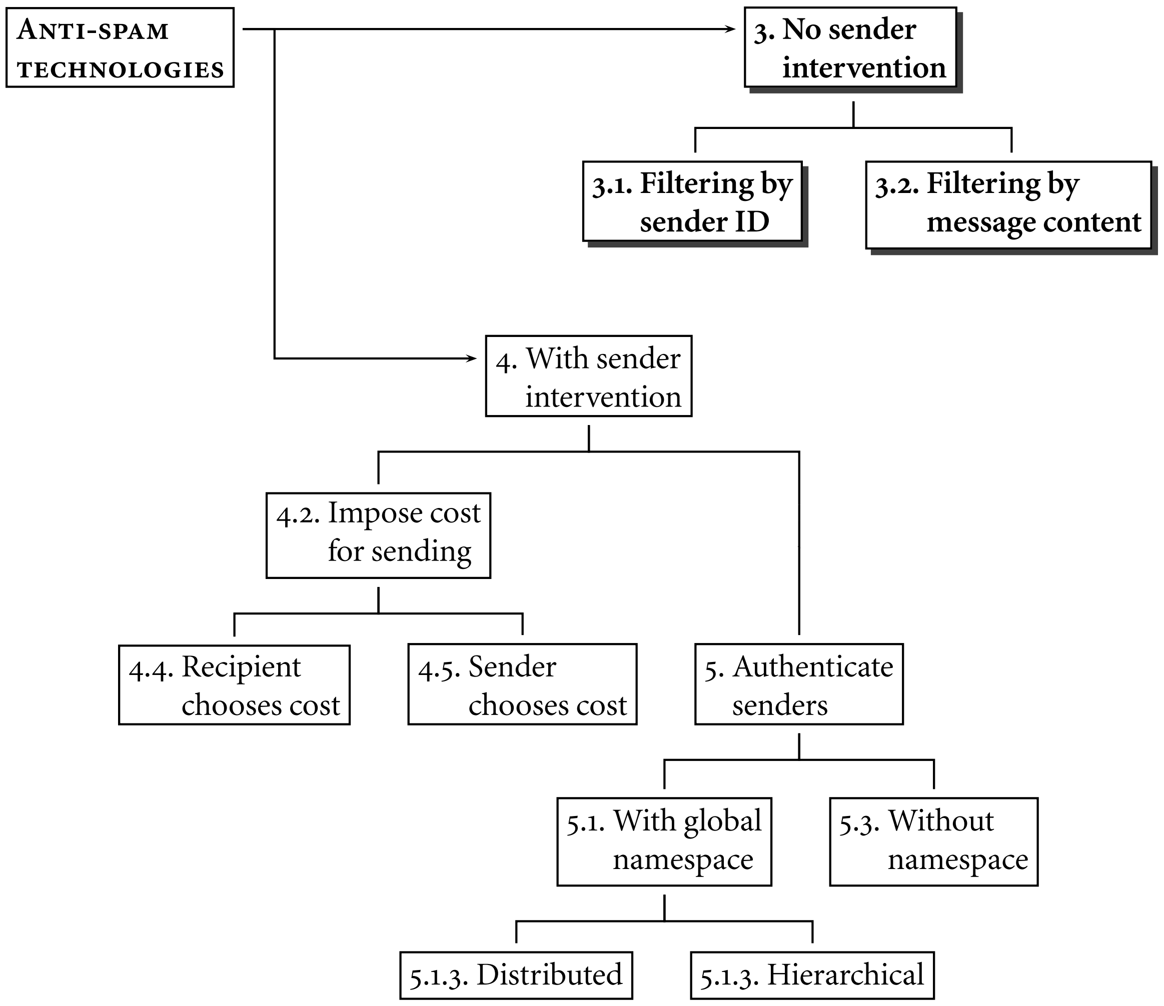
Even with this vague, circular definition it is still possible to draw conclusions about the nature of spam. It is reasonable to assume that, in general, if one recipient determines a message to be spam, then *most* people would consider it spam, and so it is reasonable to expect that the sender knows that the message will be viewed as spam. The sender would only send the message if it were in some way useful to send it, and if most people would discard the message as spam, then it is necessary that the message be sent to a large number of people. As a result, it is necessary that spam be sent in large volumes.

In addition to understanding the nature of spam, we need to understand the nature of those who send spam, commonly known as *spammers*. This paper takes the position that spammers are powerful, knowledgeable, and resourceful. If there exists some vulnerability in an AST that will allow a spammer to send messages, then that spammer will take advantage of the vulnerability.

This is a reasonable assumption. Interviews with self-proclaimed spammers and analysis of spam messages confirm this: distributed network attacks, intricate business bargaining, and innovative contortions of spam messages reveal their intelligence [18, 23, 33]. Experience with recent computer viruses—many of which used the infected computers to send spam—show that spammers can acquire networks of hundreds of thousands of computers to use in carrying out distributed attacks [37]. Solutions to the spam problem that rely on the assumption that spammers are predictable, unknowledgeable, or without resources are bound to fail. A successful anti-spam system must be able to sustain a severe, well-conceived attack, as those sending spam will utilize such attacks.

The following notational conventions for variables will be used in this paper. Roman letters will represent normal entities on the system and their related items, and Greek letters will represent adversaries and adversarial items. Capital letters will refer to either sets of entities or large server computers; lowercase letters will refer to individuals. Finally, the sender of a message will always be denoted as s, and the recipient always as r.

Systems Without Sender Intervention


Anti-spam technologies
3. No sender intervention
3.1. Filtering by sender ID
3.2. Filtering by message content
4. With sender intervention
4.2. Impose cost for sending
4.4. Recipient chooses cost
4.5. Sender chooses cost
5. Authenticate senders
5.1. With global namespace
5.3. Without namespace
5.1.3. Distributed
5.1.3. Hierarchical

Chapter 3

Systems Without Sender Intervention

At the top of our hierarchy of anti-spam technology is the question, does the sender need to perform some sort of intervening action, provide some sort of proof, in order for the AST to work? The general advantage of systems that make no requirements of the sender is that they are simple to implement and require no long-term persistence on the part of the sender. The disadvantage is that such systems by nature lack sufficient spam prevention capabilities due to the limited "knowledge" present in the system, as we shall see.

If the sender provides no aid to the mail system, then the AST has two pieces of information to work with for each message: the message's content and the network identity of the sending computer. Since this is the entirety of the communication with the recipient, that is all the information that this form of AST has to work with.

Thus we may break down such "non-intervention" systems into two types: those that evaluate the message based on the identity of the sending computer and those that evaluate the message based on its content. In Section 3.1 we deal with the former of these systems; in Section 3.2 we deal with the latter.

3.1 Filtering based on Sending Computer Identity

Using the network identity of the sending computer to identify spam is one of the simplest AST's in use today. It is most commonly implemented as "whitelisting" or "blacklisting," in which a message is either accepted or rejected based on whether or not the sending computer is present on a list of accepted or blocked network addresses. Organizations such as MAPS and SPEWS provide public IP address blacklists for use in such filtering [29, 40].

Whitelists and blacklists are essentially the theoretical extent of this filtering method. The most general network-identity-based filter would take a network address and produce some sort of value indicating whether the likelihood of that address sending spam. If we could freeze time for an instant and run this function on all possible network addresses, then we could create a timestamped table that maps computer addresses to spam indication values. That is essentially what anti-spam data warehouses such as MAPS and SPEWS do: they provide these tables as a service, using various secret mechanisms to stay up-to-date, such as setting up spam honeypots or testing networks for open relays [30].

In terms of the systems in use today, one problem with this filtering method is that network addresses can be forged by various network attacks, so it is possible for a message to appear to originate from one location when it actually came from another. Systems like DNSSEC have been proposed to solve this problem by using cryptographic techniques to provably verify network addresses, but they have yet to be implemented [4, 12].

Even if that were solved, though, there is still a critical problem with this approach. Underlying this type of system is the assumption that e-mail is spam when it originates from particular computers, but there are many ways in which both spam and legitimate mail can appear to come from the same sending computer.

The first of these is the simple case in which both the spam sender and a regular person use the same computer for sending mail. Consider, for instance, the following situation. A sender, s, sends an e-mail and then leaves to get a cup of coffee. During the five minutes when s is away, an attacker σ uses that same computer (by compromising it or just walking up to it) and sends out spam messages. Then s returns and sends out another legitimate e-mail. A recipient, aware only of the network address from which the mail was sent, is unable to distinguish between mail from s and mail from σ, since the messages were sent at practically the same time. Making the situation more complex is the prevalence of public computers and publicly available networks; in those cases it is common for many different senders—including spammers—to use the same sending computer.

The transience of network addresses only worsens this problem. If addresses are constantly changing, then an address cannot assuredly identify a certain *computer*, much less the sender using that computer. The approach taken to this situation today is simply to block large address ranges: for example, a few months ago all mail from Harvard University was rejected by the Internet service provider America Online because of a single compromised computer that was acting as a spam relay [19].

Finally, the use of relay computers makes determination by network address virtually impossible. When the recipient receives the message, it may have gone through several relay computers, and it is possible that one of those relay computers actually generated the message and simply claimed to be relaying it from another location. Based on the above observations, it is clear that merely using the sending computer's network address is insufficient to make any valuable determinations with regard to spam.

Filtering messages by network identity today seems to work acceptably well. This is only

FIGURE 3.1 Summary of the analysis of the network identity filtering AST.

Assumed definition of spam Given a time t, there is some set of computers C_t such that spam is defined as messages coming from any computer in C_t.

Potential errors If both a legitimate message and spam are sent from the same computer at about the same time, then either both will be accepted or both will be marked as spam.

Successful areas If only a small, unchanging set of computers is sending spam, then the system can be highly successful in blocking those messages.

Potential attacks Possible attacks include compromising a large number of computers, changing network addresses often, spoofing network addresses where possible, and sending mail through relay computers.

because many spammers tend to use a few easily-identifiable computers to send spam. As more and more of them start using more complex tactics such as network spoofing, compromising computers, and sending messages through open relays—and many spammers today currently employ these tactics—spam identification by sender network identity will fail to control the spam problem, blocking legitimate e-mails through accidental or invalid blacklistings along the way.

3.2 Filtering by Message Content

Spam identification based on e-mail content is both one of the most popular ways of identifying spam and one of the hottest research topics in the anti-spam field. It is worthwhile considering whether this area has a reasonable shot at solving the spam problem, given the amount of energy that is being put into the area now.

Over time, content filters have improved considerably. Originally these filters just looked for and blocked certain keywords; later they developed scoring and phrase-identification methods [42]; now adaptive Bayesian models based on statistical theory are growing in popularity [35, 38]. Claims have been made to "99.9% effectiveness" in eliminating spam, and some believe that content filtering will solve the spam problem within the next decade [44].

This claim is not valid, as this section will show. First we consider why currently implemented solutions have been thus far ineffective, and second we examine a theoretically optimal content-filtering AST and its spam-filtering properties.

While major efforts in anti-spam research have focused on content-based filtering, implementations have been ineffective, evidenced by the proliferation of spam despite popular use of these filters [14]. The reason for this is that spammers have found ways to confuse, manipulate, and outrightly deceive the filters. Intentional misspellings ("Fr33 M0n@y") are easily understood by humans but not by filters; random words from a dictionary cause Bayesian filters to adapt incorrectly. John Graham-Cumming has developed a "Spammer's Compendium" of tricks for circumventing even the best filters, using among other techniques unusual conformations of HTML code [18].[1] Spam delivery technology, or SDT, has developed and is developing just as quickly as content filtering (and other) AST develops.

What would the optimal content-filtering AST look like? It would act much like a virtual secretary, using artificial intelligence techniques to evaluate the semantic content of each e-mail message. Unfortunately, though, consider the ultimate SDT: an automated mass-mailer that uses artificial intelligence techniques to create a message with semantic content that looks sufficiently "human" that even the best filter should accept it. If artificial intelligence research advanced to the point that we could build a perfect content-filtering AST, then it most likely would have advanced to the point of building the perfect SDT as well.

[1]Some of the techniques described by Graham-Cumming are so spectacularly innovative that the author strongly encourages the reader to look up this work.

An unusual type of system that falls into this category of content filtering is the *collaborative filtering* system, one commercial implementation being Brightmail [9]. In these systems, messages are compared against a known database of spam messages, and the database is maintained by a large group of people (company employees, in the case of Brightmail). These systems tend to be more accurate today, as there are mature algorithms for finding similarities between two texts, but the algorithms for identifying "spam-like" content are much less developed. However, collaborative filtering systems are theoretically vulnerable to the same sorts of attacks as any other content filtering system: a sufficiently advanced SDT could produce content that defeats the filtering algorithm.

Content filters operate under the assumption that spam has a certain type of semantic content, such as marketing a product. This is a flawed definition for two reasons: first, spam is sent for many different purposes, and second, spam can be made semantically identical to a legitimate message. Because of these two flawed inherent assumptions of content filters, they are insufficient for preventing the delivery of spam.

It is a common mistake to believe that all spam is sent for marketing or money-making schemes. Fundamentally, spamming is nothing more than the ability to send out a large number of messages to a wide audience, and there are many uses for that ability. Viruses and worms, for instance, send out spam messages in order to infect the recipients' computers and carry out distributed network attacks; most spam filters let these virus-laden messages pass because they appear to be unrelated to marketing. Filters cannot predict future uses of the ability to send large volumes of messages; the best they can do is react to existing spam.

The other fundamental problem is that semantically meaningful message content can be duplicated *identically* by a computer, and consequently it is possible for an independent

FIGURE 3.2 A sample spam e-mail generated by the Bagle virus (with slight modifications), and then a sample legitimate message, both of which have semantically similar content.

From: administration@harvard.edu
Subject: Warning about your e-mail account.

Dear user of Harvard.edu:

Some of our clients have complained about the spam (negative e-mail content) outgoing from your e-mail account. You have probably been infected by a proxy-relay trojan server. In order to keep your computer safe, follow the instructions.

For more information see the attached file.

From: compusec@fas.harvard.edu
Subject: Network Task Disabled: Computer Virus

The FAS Computer Security Group has received a report that your computer may be infected with a virus and is attempting to infect other machines.

A single infected computer can quickly infect many additional computers on the FAS Network. To ensure network security and integrity, we must immediately disable network connections of infected computers.

Your network connection is now scheduled to be terminated.

computer to generate a message that a human might have sent. Figure 3.2 provides an example of an actual spam message active around February of 2004 and an actual network deactivation notice; the two messages contain similar content and tone. The fact that, over a twelve hour sampling period, a similar virus infected 80,000 computers, indicating that 80,000 people believed the message and opened the attached file, shows the difficulty of distinguishing between legitimate messages and spam, even by humans [13].[2]

A potential counterargument is that an independent computer is not always able to forge

[2]It is noteworthy that the virus discussed here was *not* successful because of a software flaw (e.g., the mail client automatically executing the virus). The virus was transmitted in a compressed file, which the victim actually had to decompress prior to executing, and later versions of the virus in fact *encrypted* the file as well, requiring the recipient to enter a password provided in the message!

FIGURE 3.3 Summary of the analysis of the content filtering AST.

Assumed definition of spam A message is spam if and only if it has a certain semantic content, that content defined by the content filter.

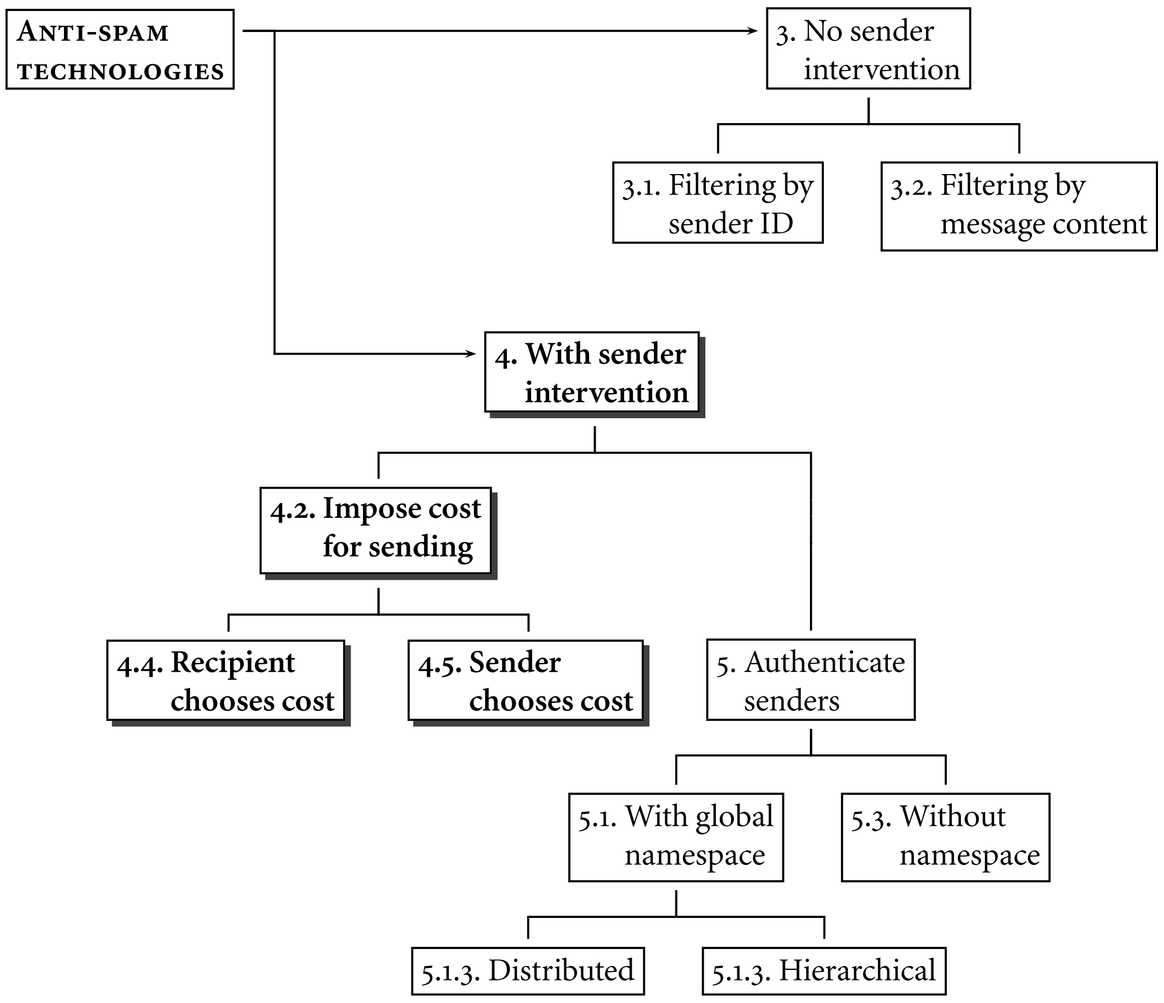
Potential errors If a spam message is written to sound much like a legitimate message, then the filter will fail to mark it as spam.

Successful areas Spam messages that are blatantly "spam-like" in content, such as marketing advertisements, could be easily detected.

Potential attacks An adversary with a sufficiently advanced spam generation AI could create messages similar in tone and semantics to legitimate messages, thus defeating the system.

a message from some given sender. The sender may include a cryptographic signature or personally identifying information in a message, for instance, and the independent computer would not have the necessary resources (the private key or the personal information) to include those. However, if a system *requires* such information in messages to thwart spam, then that system requires sender intervention, so it does not fall under this category of AST. Content filters on their own, then, can never be fully effective because they rely on the differentiation of content between spam and non-spam e-mail, and spam need not be differentiable from legitimate e-mail in terms of content.

Proof of Intent Systems



Anti-spam technologies
3. No sender intervention
3.1. Filtering by sender ID
3.2. Filtering by message content
4. With sender intervention
4.2. Impose cost for sending
4.4. Recipient chooses cost
4.5. Sender chooses cost
5. Authenticate senders
5.1. With global namespace
5.3. Without namespace
5.1.3. Distributed
5.1.3. Hierarchical

Chapter 4

Systems with Sender Intervention: Proof of Intent Systems

In order to understand how sender intervention can help prevent spam, we can consider the non-digital analogue of e-mail: postal mail. For some reason, we do not experience an overwhelming spam problem with our physical mailboxes. What are the differences between postal mail and electronic mail that allow spam to flourish in one but not the other?

The first difference is obvious: cost, or sending effort. There is the monetary cost of the postage, the cost of the paper and envelopes, and the effort of delivering the letter to the post office. In contrast, there is no monetary cost to sending e-mail; the computing resources required are next to nothing, and the entire process can be scripted so the cost of human effort is independent of the volume of mail being sent. If we implemented a cost to sending e-mail, with any form of cost (monetary, computing effort, human effort, etc.), we might be able to prevent spam, at least in its bulk-mail form.

The second difference is more subtle. When two people correspond by postal mail, they have the paper quality, the handwriting, the textual style, and other elements to ensure the recipient of the sender's identity. In the case of bulk mail, the sender generally has to actually bring the letters to the post office (presorted, to avoid high cost!) and pay the postage by check or credit card, so the sender's identity is verified to some extent by the postal worker. In contrast, the digital nature of e-mail makes it possible to duplicate a message millions of times; sender identities can be indistinguishably forged by nothing more than typing in a fake name. These examples all indicate that identity verification differentiates postal and electronic mail.

We can analytically prove that requiring the sender to pay some cost and verifying the identity of the sender are the only two ways in which sender intervention can help in controlling spam. In Section 4.1 we show that proof of intent and identity verification are the only two forms of intervention that allow the recipient to distinguish the sender of a message from other entities, and we show that this distinction is a necessary condition of a successful sender-intervention AST. In Section 4.2 we show that a proof of intent is equivalent to a cost, so a system that uses proof of intent is equivalent to a cost-based system.

4.1 Proof of Intent and Identity Verification

The following proof will demonstrate that identity verification and proof of intent are the only forms of sender intervention that might be useful in identifying or preventing spam. The proof is conducted rather indirectly, and can be outlined in the following way:

1. If a system is to prevent spam, then message recipients using that system must be able to make valuable inferences about that sender.

2. If a recipient can make valuable inferences about the sender, then that recipient must be able to distinguish the sender of a message from other entities.

3. If a recipient can distinguish the sender of a message from other entities, then that recipient must both (1) identify a distinctive quality of some entity (one that other entities do not have) and (2) have some knowledge that the sender of the message possesses that distinctive quality.

4. One way the recipient could have such knowledge is by possessing some prior *rule* specifying that the sender possesses some distinctive quality.

5. Otherwise, the recipient has no prior knowledge of the sender, and the only distinctive quality of the sender is that sender's intent to send the message.

6. Therefore, if a sender-intervention AST is to prevent spam, then it must either provide the recipient with a rule (identity verification) or verify the sender's sending intent.

The proof is conducted in reverse order, so we begin by showing that, without prior knowledge, a recipient can only determine the sender of a message by identifying the sender's intent to send the message. Let s, t, and r be entities such that exactly one of s and t wishes to send a message m to r. The goal of r is to determine which of them wishes to send m.

The work of David Hume shows that an observer can only detect *qualities* of entities, so, to an observer, *an entity is represented entirely as a set of qualities* [22]. Then r can use nothing more than observed qualities of s and t to determine the sender of the message.

Identity verification can be understood in the following way:

> A *rule* is a provable assertion that the sender of a message has some quality. Given a message m, the rule asserting that the sender of m has quality q is denoted as $\langle m, q \rangle$. *Identity verification* is defined as the use of a rule to determine the sender of a message.

Say that r possesses the rule $\langle m, q \rangle$. Then r somehow detects whether q is present in s and t. If $q \in s$ but $q \notin t$, then s must be the sender. If both of them have quality q, then the rule is not useful in determining which is the sender. As an example, consider the use of signatures on a message (either cryptographic or handwritten). The recipient uses the rule, "the sender of the message has the quality of possessing that signature," thus allowing the recipient to determine the identity of the sender. Note the loose usage of the term *prior*: r may acquire the rule only after the message has been sent.

It should be intuitively clear that any useful information available to r about the sender of a message can be reformulated into a rule. A proof of this is omitted.

Now consider the case in which r does *not* have any rule and thus lacks prior knowledge about the recipients. We prove that the only quality useful in determining the sender of the message is the sender's intent to have that message delivered. Note that, in this proof, it is irrelevant whether or not the message is spam; the question is how r can determine the sender of that message, whatever its content may be.

THEOREM 1. *If r has no prior knowledge of the actual sender, then the only quality that allows r to determine whether s or t is the sender is the quality of intent to send m.*

Proof. The intent to send m is a quality of an entity; we name that quality i_m.[1] The qualities of s are the set Q_s, and those of t are Q_t. Neither Q_s nor Q_t contains i_m.

Assume two possible worlds W_s and W_t, in which s and t are the potential senders, respectively. In W_s, sender s has intent to send the message and t does not, and the opposite is

[1]It is possible that the quality of sending intent would induce other qualities (nervousness, pride, etc.). However, assessing those qualities would be indirectly assessing sending intent, so such qualities are essentially equivalent to sending intent.

true for W_t. In other words, for each of the worlds, the qualities of s and t are as follows:

$$W_s \implies s = Q_s \cup i_m,\ t = Q_t$$

$$W_t \implies s = Q_s, \qquad t = Q_t \cup i_m$$

Determining whether s or t is the sender is equivalent to determining which of W_s and W_t is the case. But based on the above relations, the only difference between the two worlds is which entity possesses i_m. Since r has no prior knowledge of the sender, the above is all of the information available to r. Therefore, other than i_m, r has no way to distinguish between W_s and W_t, so the only distinction between s and t that is useful to r in determining which of them wishes to send the message is i_m. ■

We can now show that identity verification and proof of intent form the only possible types of AST with sender intervention. We assume a "challenge-response" protocol: the recipient may issue challenges to the sender, and the sender can helpfully intervene by responding to those challenges. A challenge-response protocol can be formalized as follows: r sends a challenge c to s, and then s provides a response $f(c, s)$. It should be clear that any intervention that the sender can provide can be generalized into this challenge-response protocol.

THEOREM 2. *The only two types of challenges that would be useful in preventing spam are a proof of intent challenge and an identity verification challenge.*

Proof. Without loss of generality, say that s is the actual sender of the message.

Imagine that r sends challenge c to both s and t and receives responses $f(c, s)$ and $f(c, t)$. Those responses are functions of *qualities* of s and t, and r can use those responses to deduce some of those qualities.

Now say that r uses a rule to determine which of s and t is the sender, and that rule states that the sender has quality q. Then r must be able to deduce from the responses the following two relations:

$$f(c,s) \implies q \in s,\ f(c,t) \implies q \notin t$$

The response therefore is a proof of identity based on the already known rule, and the system performs identity verification.

The other case is that r does not use a rule and thus has no prior knowledge. Then, by our above theorem, it is necessary for r to identify i_m in the sender, so the challenge must allow r to deduce the following:

$$f(c,s) \implies i_m \in s,\ f(c,t) \implies i_m \notin t$$

The response therefore is a proof of intent. Therefore, the only challenges that allow r to distinguish the sender of a message from other entities are challenges that demand a proof of identity and challenges that demand a proof of intent.

If the response to a challenge does not distinguish the sender from other entities, then that response cannot be used to make any valid inferences about the sender, since any inferences drawn from that response would be the same for all entities on the system. Determining whether a sender of a message is sending spam is an inference that an AST must draw from the response to the challenge. Therefore, the only challenge-response systems that could make a useful AST are systems that verify the identity of the sender and systems that verify the sender's intention of sending the message. ■

4.2 Proof of Intent Systems

Systems based on proof of intent, labeled "cost systems" from here on, are a lesser-known technique in the spam wars, but new ideas in this subfield are propelling it into mainstream research. In particular, the employment of non-monetary costs such as computing resource costs have made this option much more attractive. We can divide types of "costs" incurred into those that are permanent, such as monetary costs (since you never get your money back), and those that are transient, such as resource time costs.

We now formalize the concepts of "intent" and "proof of intent." Let s be a sender—either a spammer or a sender of legitimate mail—who wishes to send a message to recipient r. Then saying that s has intent to deliver m means that s would be happier if r were to read the message. Since the metric of happiness in economics is utility, intent is the positive utility u that s would gain from r reading the message. If s is to provide a proof of intent, then s must make a demonstration that $u > 0$.

The fundamental problem is that r is unaware of u. This creates an "asymmetric" situation where one party is lacking in full information. As Loder *et al.* observe, this means that the spam problem can be explained in terms of information economics [27].

Information economics is the field of research in market systems in which there exist information asymmetries, situations in which some parties may be better informed than others. This work was pioneered by George Akerlof's seminal 1970 paper "The Market for 'Lemons': Quality Uncertainty and the Market Mechanism" [2]. The key problem is that one party knows of the utility of an item, while another party does not know that utility; Akerlof shows that, other factors not present, low-utility products will dominate the market. In the context of e-mail, the sender knows the utility of the message, but the recipient does not, so low-utility spam will predominate on the e-mail system.

Fundamentally, the solution to this problem involves s somehow indirectly proving the existence of u to r, who cannot observe u directly. Disregarding identity, which we discuss in Chapter 5, the only quality that distinguishes s from other entities on the system is u. So the only proof that s can offer of $u > 0$ is the performance of some action that has cost $c \leq u$. Thus a proof of intent must be the payment of that cost c or some sort of insurance that c will be paid upon demand. Note that we do not specify that $c > 0$, but it seems intuitive that an action with no cost would not produce a useful proof, and we will show this to be true.

There are two categories of solutions to this problem, called signaling and screening. Consider a transaction between a sender s sending a message m to a recipient r. In screening, r would give s a test; if s passes the test, then r has some confidence that m is valuable. This is essentially a system in which r determines the cost that s is to pay; the cost is the test, and payment of the cost is passing the test. In contrast, signaling involves s taking the initiative to prove the utility of m. In this sort of system, then, the sender offers to pay c. Michael Spence and Joseph Stiglitz identified the concepts of signaling and screening; for their work they shared the 2001 Nobel Prize in Economics [28].

Although the word "cost" carries the connotation of a monetary cost, cost can actually take on many forms. There have been several proposals of using computation time as a cost function in AST's [7, 11], with more recent research focusing on using memory-bound functions so that computers with greatly varying CPU powers can perform reasonably similar challenges [1, 10]. The idea behind those systems is that only senders who have a strong intent to have their messages read will be willing to perform the difficult computations, thus proving their intent of sending the message. Also considered a cost function is a challenge-response system like CAPTCHA, which presents an automated Turing test challenge that should only be solvable by humans [43]. In this case, only senders who have a strong intent should be willing to spend their own time doing the test.

The first question to answer is whether the cost should be permanent or transient. A permanent cost is one in which, after payment of the cost, the sender has given up something permanently, such as money. A transient cost is one in which the sender, after payment of the cost, is in the same state as prior to payment. Consequently, the only change between the pre-payment and post-payment states of the sender is time, so a transient cost is a payment of time. In the next section we will discuss the problems with permanent costs, and thereafter the focus will be on transient cost systems.

The second question to answer is what the the appropriate cost to demand. First, the recipient or some external body could dictate the cost, so cost is determined prior to the transaction and we have a screening system. Second, the sender can negotiate the cost, so cost is determined during the transaction and we have a signaling system. We will discuss first predetermined cost systems and then transaction-determined cost systems.

4.3 Problems with Permanent Costs

Imposing a permanent cost on sending e-mail would be the cost-based system most similar to the postal model, as sending a letter requires a monetary payment. However, permanent costs are an unsatisfactory solution, not because they would be ineffective in preventing spam, but because they conflict with the nature of e-mail itself and they present severe difficulties of implementation and maintenance.

It is worth noting that any form of permanent cost is essentially equivalent to a monetary cost. If sender *s* makes some sort of permanent payment to recipient *r*, be it in hard disk storage space, digital currency, or pack llamas, *s* could simply send *r* the monetary value of that item. In the case of digital currency, this may not be immediately obvious, as the currency arbiter may not allow the direct purchase of points, but one could imagine a digital currency market developing on the side; if there is money and a demand for virtual currency, then people will find a way to exchange.[2] At any rate, permanent cost systems can be reduced to monetary cost systems, so we will restrict the following discussion to monetary cost systems.

The first difficulty of implementing a monetary cost to e-mail is that it introduces the possibility of theft. Since sending e-mail is a common occurrence, the system must make it easy for funds to be transferred, and that introduces the possibility of funds being transferred into the wrong hands. One might argue that electronic fund transfers already exist on the Internet, but the prevalence of Internet fraud shows that theft already exists as well: the Federal Trade Commission recently reported that in the 2003 there were 166,617 reports of Internet-related fraud, costing $199,355,357 [15]. Forcing e-mail users to put their money into this risky arena seems to be an unreasonable demand.

Second, a key advantage of e-mail is the fact that it is free. Day-to-day use of e-mail reflects this. People writing large research papers often e-mail the documents to themselves as a form of network backup. Testing a new e-mail account often involves sending a test message; troubleshooting a broken account often involves sending many. Some forgetful people (author included) often send messages to themselves as reminders. Students doing research

[2]As an unrelated example, consider the United States' purchase of pollution credits in order to comply with the Kyoto Protocol on air pollutant emissions.

can send many e-mails to professors and other knowledgeable sources while living on a student's budget. Imposing a monetary cost for sending messages could invariably eliminate these and other equally reasonable and common uses of e-mail.

Third, even if a monetary cost were imposed on e-mail, the Internet community would find a way around it. Imagine that a monetary cost system were successfully implemented. Certainly organizations would relax the cost on intra-organizational e-mail; messages sent within a university, office, or ISP would be delivered for free. Then, organizations would strike agreements between each other: Harvard, Berkeley, Stanford, and MIT might agree to deliver mail to each other without charge, and these agreements might then grow to encompass more and more of the Internet. Correspondents at locations without such agreements might turn to alternate messaging systems such as instant messaging or Internet Relay Chat. Without imposing some sort of monetary cost on *all* Internet traffic, it seems unlikely that a monetary cost on just e-mail would ever take hold.

Fourth, a monetary cost would require a stronger sense of identity on the Internet. The nature of e-mail addresses is that they are not strongly tied to individual entities (humans or computers). Since only actual entities can pay permanent costs, there would need to be a stronger tie between entities and e-mail addresses. However, the loose association between e-mail addresses and entities is valuable; the way we use e-mail today reflects this value. E-mail accounts are generally associated with both an individual and an institution (e.g., the author's address *cduan@eecs.harvard.edu* is associated with both the author and the EECS division of Harvard University), and since individuals may move among different institutions, e-mail addresses can frequently change and be revoked. Creating a stronger bond between entities and addresses would potentially require complex tracking and revocation schemes.

In summary, a monetary cost for e-mail would be difficult to implement and might end up ineffective anyway, and even if it were implemented, it would take away inherent key advantages of e-mail. Solving the spam problem by imposing a monetary cost on e-mail would thus solve one major disadvantage by removing one major advantage, and it seems difficult to justify such a system in light of that.

4.4 Cost Determined by the Recipient

Most proof of intent systems involve a well-known fixed cost, using some technology to implement that cost. The cost may be a function of two values: the number of messages that the sender wishes to deliver and the network identity of the sender. In the general case, we can assume that the sender will have to pay the maximum cost of all network addresses, so the cost becomes just a function of the number of recipients.

This last point may seem contentious. A common suggestion is to use whitelists to permit messages from certain known senders and require all others to pay a cost, so in the general case no cost is paid by the sender. This is a useful but incomplete solution. Network identities are volatile, and people change e-mail addresses, so the whitelist would need to be updated constantly. Unless a strong identity verification system is implemented (which we discuss in Chapter 5), addresses can be forged and thus the whitelist might erroneously accept messages. Finally, if whitelists become a general practice, lawsuits might ensue over, say, whether refusal to add someone to the list is discrimination. It seems safest to assume that senders will generally need to pay the maximum cost.

Since the cost is simply a function of the number of recipients, we can investigate the boundaries of that function.

THEOREM 3. *Given a number of messages n to be sent, let $f(n)$ be the predetermined cost for sending those messages to a given computer. Then $f(n)$ must be, in general, monotonically increasing; that is, if $n_0 \ll n_1$ and n_0 is a reasonable number of messages, then $f(n_0) < f(n_1)$.*

Proof. Assume that $f(n)$ is not generally monotonically increasing. Then there exist two values n_0 and n_1 where n_0 is reasonably small, $n_0 \ll n_1$ and $f(n_0) \geq f(n_1)$. Then if any sender wished to send n_0 messages at a cost of $f(n_0)$, then that sender could just as easily send n_1 messages for the same cost, so $n_1 - n_0$ messages would essentially be delivered by the system for free. Since n_0 is a reasonable number of messages, this could become a common occurrence, and then the system would fail since many messages would be delivered properly without any proof of intent. ■

One interesting corollary to this theorem is that, for all $n > 0, f(n) > 0$; that is, the cost must be positive. This theorem provides a lower bound for the cost function; there is also an upper bound.

THEOREM 4. *Given a number of messages n to be sent, let $f(n)$ be the predetermined cost for sending those messages to a given computer. Then $f(n)$ must be at most linearly proportional to n; in other words, given integers x and n:*

$$f(xn) \leq xf(n)$$

Proof. Say a sender wishes to send xn messages. The sender can then send groups of n messages in separate transactions, making x transactions each with cost $f(n)$, so the total cost is $xf(n)$. Alternately, the sender can make one transaction sending all the messages, with total cost $f(xn)$. Naturally, the sender will take the less costly of the two options, so it is ridiculous for a system to impose a cost $f(n)$ greater than $nf(1)$ since the ultimate cost to the sender will be at most $nf(1)$. ■

The observation that the cost function must be monotonically increasing indicates that proof of intent systems essentially are trying to block large volumes of mail from being sent. In other words, spam is implicitly defined as any mail that is to be sent to a large number of recipients. But, as discussed at the beginning of this paper, there are legitimate reasons to send bulk mail, so proof of intent systems inherently prevent such uses of e-mail. We now show that, if reasonable bulk mail such as mailing list messages can be sent, then a simple distributed attack is sufficient to allow an adversary to send spam through the AST.

We make the following numerical assumptions. A mailing list may consist of up to n_m members, so it should be possible for an individual to send n_m messages through the AST without difficulty. Second, the goal of the adversary is to send n_s messages. Finally, a spammer can get access to n_c computers.

These values are set as follows:

$$n_m = 1{,}000 \qquad n_s = 10{,}000{,}000 \qquad n_c = 10{,}000$$

These are generous assumptions. Mailing lists often have many more than 1,000 recipients; the Linux kernel mailing list, for example, has 3,611 recipients [32]. The probability of spam being effective is well more than one in 10,000,000; interviews with actual spammers report effectiveness rates between 0.013% and 1%, and a spammer can be profitable with a response rate of one in 100,000 [23, 33]. And it is no difficult task for an adversary to control thousands of computers: computer viruses can compromise 80,000 computers in a day, and two years after the initial attack by the Code Red virus, 20,000 hosts still remained available [37].

THEOREM 5. *Given those assumptions, the spammer can use a distributed attack to meet the message goal of sending n_s messages.*

Proof. Again, let the cost of sending n messages be $f(n)$. Then, by our theorem:

$$f(n_s) \leq \frac{n_s}{n_m} f(n_m)$$

Since the adversary has access to n_c computers, the computational effort per computer is:

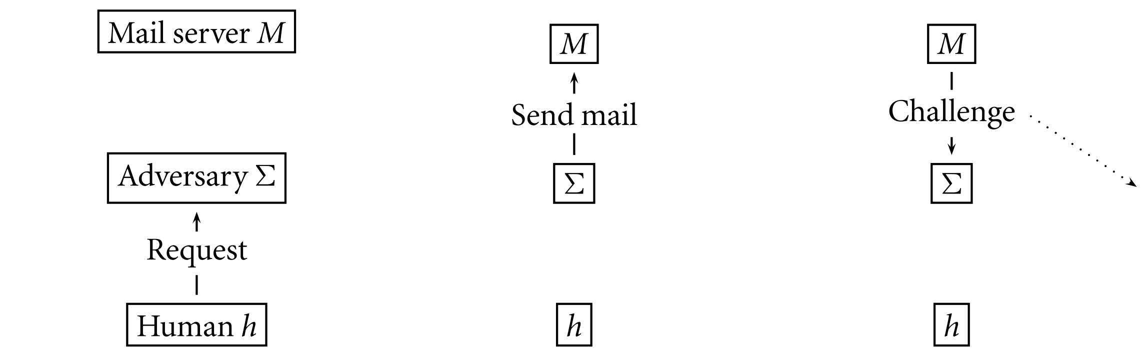
$$\frac{f(n_s)}{n_c} \leq f(n_m)\frac{n_s}{n_c n_m} \leq f(n_m)$$

The last part is true because $n_s \leq n_c n_m$ by our numerical assumptions. Therefore, the adversary's task is computationally feasible. ■

While the previous example refers to a proof of intent by computing resources, proof of intent by human resources, such as an automated Turing test, is no less susceptible to a "distributed attack." Figure 4.1 demonstrates one theoretical method by which an adversary could distribute a human-resource challenge such as an automated Turing test to many unsuspecting humans. Simply put, the adversary sets up a service that humans may find interesting or useful; the service requires that the human answer some sort of challenge first. The adversary then delegates challenges received from the e-mail system to those humans interested in the service. Rosenthal *et al.* remark that spammers may even be using this attack already [37]. Consequently, even in the case of human resource costs, an adversary can employ a distributed attack to distribute more messages than the average sender could.

In summary, AST's based on predetermined costs are reducible to systems that are intended to block large volumes of e-mail, which can cause problems because there are legitimate reasons to send large volumes of e-mail, such as mailing lists. Additionally, such

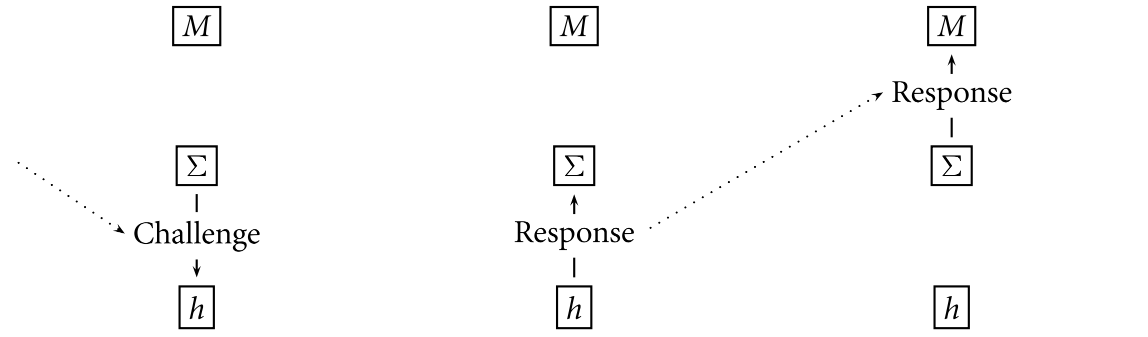
FIGURE 4.1 A method by which an adversary can distribute a human-resource cost such as an automated Turing test. The adversary, Σ, wishes to send spam to a mail server *M*. To do so, Σ runs a website providing some service to members who have signed up.



(A) A human *h* requests to sign up for the service run by Σ.

(B) Σ makes a request to send messages to *M*.

(C) *M* presents a challenge to Σ as a condition for delivering the mail.



(D) Σ presents that same challenge to *h* as a condition for Σ's service.

(E) The challenge is solved by *h*.

(F) Σ passes the same response to *M*, thus allowing the mail to be sent.

FIGURE 4.2 Summary of the analysis of the cost-based AST with the cost chosen by the recipient.

Assumed definition of spam Any message that is to be delivered to a sufficiently large number of recipients is considered spam.

Potential errors Legitimate uses of bulk mail, such as mailing lists, might be considered spam. A spammer with sufficient resources could send spam accepted by the system as legitimate.

Successful areas Personal messages delivered to a small number of recipients would be easily sent. Additionally, because in general spammers rely on the ability to send large volumes of messages, this system would at least make the spammers' jobs more difficult.

Potential attacks An adversary who is able to compromise a reasonably large number of computers could use a distributed attack to send spam.

systems are vulnerable to simple distributed attacks. As a result, an AST based on a predetermined cost will either block legitimate bulk mail or it will allow spam to be sent via a distributed attack.

4.5 Cost Determined by the Sender

Systems in which the cost is predetermined are essentially systems in which the recipient dictates the cost of sending the e-mail. The alternative, then, is to have the *sender* determine the cost or at least take part in determining the cost. This is the type of system that Loder *et al.* propose in their Attention Bond Mechanism [27].

A nice feature of this sort of system is its implicit definition of spam: a legitimate message is any message on which the sender is willing to stake a high cost, and spam is any message for which the sender is unwilling to do so. As a result, the system's definition of spam is equivalent to the sender's idea of spam, and it is reasonably safe to assume that the sender's idea of spam is similar to the recipient's. So if there is a viable solution in this category, then its chance of success in solving the spam problem would be high.

The simplest form of a sender-based cost system would be one in which the sender pays up front whatever cost seems appropriate, and the recipient deems messages where the sender has paid a sufficiently high cost to be legitimate. Naturally, this system falls into the same problems as predetermined cost systems: either legitimate bulk messages cannot be sent or a distributed attack will allow spam to be delivered.

Instead, Loder *et al.* propose a system in which the sender does not immediately pay the cost but rather sets up a warranty on the message. The hope of that system is that if a recipient deems a message legitimate, then the warranty will be forgiven; otherwise the recipient will collect on the warranty. Consequently, legitimate mail senders will be willing to put up high warranties so their messages will be easily identifiable from spam without fear of having to pay those warranties.

Unfortunately, warranties may be difficult, if not impossible, to implement with transient costs. If there is no third-party broker, then it is clearly impossible to implement such a system, since the sender could simply send spam with a high warranty, default on the warranty, and disappear. One could solve this with a reputation scheme, in which a sender who defaults on a warranty is given a bad rating. But if we can implement a successful rating scheme for e-mail senders, then there is no reason to use the cost system in the first place! Furthermore, a reputation scheme would require strongly verifiable identities for every e-mail user; as we will show in Section 5.1, such a system is not feasible.

Instead, we can use a broker who holds the cost put up under warranty until the recipient decides to collect on the warranty or forgive it. But how is it possible for a broker to temporarily hold onto a transient cost? Since a transient cost is a payment of time, and time

FIGURE 4.3 Summary of the analysis of the cost-based AST with the cost chosen by the sender.

Assumed definition of spam If the sender of a message is unwilling to provide sufficient proof of intent, such as a sending cost or a warranty, then that message is considered spam.

Potential errors Unknown, but potentially the same as a recipient-chosen cost system.

Successful areas Unknown, but potentially the same as a recipient-chosen cost system. However, a properly implemented warranty-based system may be successful in allowing legitimate bulk mail to be sent (for example, if proper etiquette develops so mailing list recipients do not collect on the warranty).

Potential attacks Like the recipient-chosen cost system, an adversary may be able to employ a distributed attack in order to send a large volume of mail.

Note The only apparent solution in this category, that proposed by Loder *et al.*, appears to not be implementable without permanent costs. It is thus difficult to identify analytical qualities of this sort of system, since a successful system in this category would be the product of an innovative discovery.

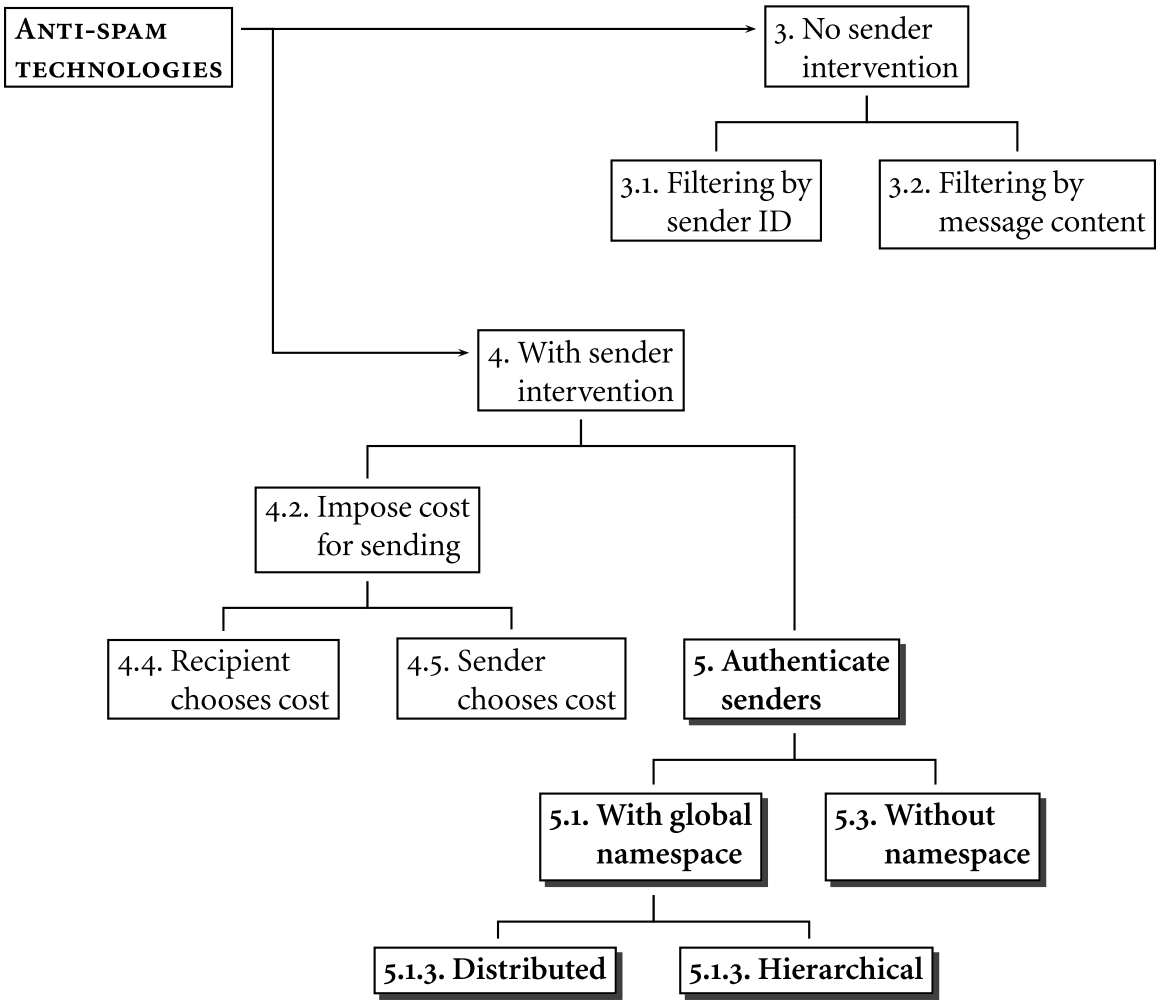
always moves forward, it seems difficult to develop a scheme in which a transient cost can be held by a broker and later forgiven or charged.

The only other option, then, is to use a permanent cost rather than a transient cost. But for the reasons outlined previously, an AST based on permanent costs is an unsatisfactory solution. So it seems that a warranty-based system will not suffice.

One suggestion by Stuart Shieber is to use a "computational stamp" [39]. The sender performs some difficult computation for a "stamp broker," and in return the broker gives the sender a stamp. The sender then attaches the stamp to the message and sends them to the recipient. If the recipient believes that the message is spam, then the broker is notified and the stamp is consumed; otherwise the recipient does nothing and the sender can reuse the stamp. One problem with this solution is that the stamps begin looking much like a digital currency, and so they begin to have some of the problems of permanent cost systems. For example, one might compromise computers, steal stamps from them, and use those stamps to send spam.

This is not to say that there is no viable solution in this category of AST's. The research of Loder *et al*. represents, to the best of the author's knowledge, the only formal recognition of the applicability of information economics to solving the spam problem, and the valuable lesson we learn from it is that proof of intent systems in which the sender chooses the proof may work better because senders are able to adjust the proof to the strength of their intent. There is no reason that a pre-paid cost or a warranty are the only possible ways of implementing signaling. Research on the part of computer science in this subfield of economics could prove worthwhile in the effort against spam.

Identity Verification Systems



Anti-spam technologies
3. No sender intervention
3.1. Filtering by sender ID
3.2. Filtering by message content
4. With sender intervention
4.2. Impose cost for sending
4.4. Recipient chooses cost
4.5. Sender chooses cost
5. Authenticate senders
5.1. With global namespace
5.3. Without namespace
5.1.3. Distributed
5.1.3. Hierarchical

Chapter 5

Systems with Sender Intervention: Identity Verification Systems

The class of AST's based on identity verification approaches the problem of spam rather indirectly, for knowing the sender of a message does not indicate whether the message is spam. The recipient verifies the sender's identity so that, if the message is spam, retributive actions can be taken against that sender.

The nice property of identity verification systems is that they employ a reliable definition of spam: a legitimate message is one that the sender is willing to sign; a spam message is one for which the sender is unwilling to make that guarantee. One hurdle to making an authentication system effective, then, is that *all* legitimate messages be signed, so that the system has this property that any unsigned message is spam.

Recall from Section 4.1 that identity verification is the recipient's use of a *rule*, an assertion of some quality of the sender, to determine the sender of a message. There are two possible types of rules: direct rules and indirect rules. A *direct rule* is one in which the rule specifies that the sender has some quality that the recipient can immediately observe during the transaction. Otherwise, the rule must specify a quality of the sender that is not observ-

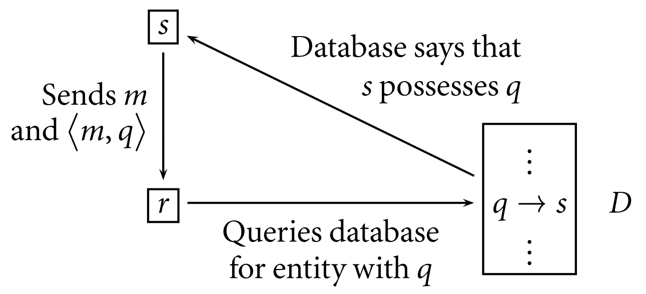

s
Database says that s possesses q
Sends m and $\langle m, q \rangle$
r
Queries database for entity with q
$q \to s$
D

FIGURE 5.1 Authentication via an indirect rule. The message m is sent along with a rule $\langle m, q \rangle$. The recipient then queries the database D, searching for q, and the database shows to r that s provably possesses q.

able during the transaction, so the recipient must use some external "database of qualities" to determine the sender's identity, through the mechanism specified in Figure 5.1. Since the recipient can only verify the sender's identity through this external database, this sort of rule is an *indirect rule*. Note that the term *database* is used loosely, referring to any sort of lookup system that allows someone to identify an entity given a quality.

A public key system is essentially an indirect rule system: the recipient must use a database of public keys to identify the sender. These systems are called *authentication* systems in this paper, and they are discussed in Section 5.1. Systems that employ direct rules allow the recipient to immediately determine the sender, so those systems are called *immediate responsibility* systems. They are considered in Section 5.3.

5.1 Sender Verification by Authentication

Members of the cryptography community have often cried, "public key message authentication can solve spam problems!" Can they really provide the solution? In this section we will see that public key cryptography, and in general any system of authentication, cannot serve the needs of users of e-mail, much less serve their needs for spam reduction.

The essential concept in the proof of this inherent inadequacy of authentication systems is that the user base is constantly in flux. Qualities are not intrinsically linked to identities; they are created, revoked, and transferred constantly, and any identity system that wishes to allow for this flexibility cannot provide the necessary guarantees of identity verification.

The following discussion will employ terminology from public-key authentication systems, although the scope of the proof is any system of authentication with indirect rules. In a public key system, the sender's private key is used to encrypt the message, so the recipient can decrypt the message with the appropriate public key. Since the recipient now knows that the sender must possess that key pair, a public database of keys should now reveal the entity owning it. In the terminology of Figure 5.1, the sender's encrypting the message provides the rule $\langle m, q\rangle$; ownership of the public key is the quality q, and the key database is D. The term *key* will thus refer to the quality of the sender specified by the rule.

It should be noted that there already exist systems like PGP and X.509 certificates that suffer from—and successfully deal with—the problems of a changing user base [21, 45]. However, these systems have a relatively small population of users: use of PGP is generally limited to academicians and secret government agents; use of certificates is generally limited to online merchants and organizations. Neither has a substantially large number of users. The proof to be presented will rely on the fact that there are many people using e-mail—and it is a reasonable assumption, because there are many people using e-mail.

5.1.1 Assumptions About our Authentication Infrastructure

What is the nature of the world in which we are trying to establish an authentication infrastructure? In this section we will discuss the assumptions that will be employed in our analysis, and we will consider a possible attack that would render the system ineffective.

We begin with a trivial but important proof, that qualities can inherently be forged by others with sufficient information.

THEOREM 6. *Let s and σ be entities and C be a computer. For any quality q, if s can prove to C that $q \in s$, and σ has sufficient information about s, then σ can prove to C that $q \in \sigma$.*

Proof. By our challenge-response protocol, C sends a challenge c to s and s returns $f(c, s)$ to C (where $f(c, s)$ proves that $q \in s$). If σ has sufficient information about s, then for all c, σ can construct $f(c, s)$, and so C would deduce from the responses of σ that $q \in \sigma$. ■

A corollary to this theorem is that keys, being nothing more than qualities of entities, must be kept secret, or else an adversary will be able to forge identities.

This proof may seem at first glance to contradict Theorem 1. There we argued that only s can have the quality of intending to send a message, and here we state that σ could imitate that quality and thus appear to have it. But although σ could *appear* to have intent, σ might or might not actually *have* that intent. True, σ could pay the sending cost for a message sent by s, but σ would only do so if there were some value to σ in having that message delivered—in which case σ actually has intent to send that message.

The following are assumptions about e-mail users:

A1. Given two arbitrary people a and b, it is unreasonable to assume that a has any prior knowledge of any validating aspects of b (e.g., facial appearance, physical features). Note that this assumption is only true if the number of entities is large; if there are few enough people on the system, then it is likely that many of them know each other.

A2. E-mail users are generally not technically savvy, and there is no reason to believe that this will change. Even if the computer savvy of the general public rises so that keys are generally not stolen, people are still forgetful; they might accidentally delete their key or forget to keep their address or phone number up-to-date.

A3. People also are generally immobile; they like to stay in one location, and it is unreasonable for a cryptosystem to force people to travel to designated places.

A4. It is often necessary to revoke a key, possibly even without the current holder's consent. For example, if person a works for a company and has key address k_a, it must be possible to quickly dissociate a from k_a if a is fired.

This last assumption requires a bit of consideration. One might suggest, for example, that we adopt a single, complete, universal namespace of keys, such that every person receives a single, immutable key. Some even go further, suggesting that there already exist universal namespaces, in the form of Social Security numbers and other such identifiers. However, recall from our above theorem that an authentication system relies on the secrecy of qualities of individuals. The prevalence of identity fraud and computer security compromises shows that such global identifiers cannot be kept secret [13, 15, 37].

Possibly, then, revocations of keys could be permitted, but only with the consent of the keyholder. For example, in a password-based authentication system, a user can generally only change passwords by presenting appropriate identification to the system administrators. However, there are situations in which a key should be revoked even without the consent of the keyholder. Consider, for example, the case of a user's key being compromised while the user is unreachable due to vacation, illness, or some other reason. If the system has no mechanism for revoking that user's key without the user's consent, then the compromised key could be used to send indefinite quantities of spam under a forged identity.

So it is *necessary* for the authentication system to have some mechanism for revoking a key without the keyholder's consent.

Throughout this discussion, we will entertain a generic attack of a conspiracy on an e-mail user u. The user's own key is p; the goal of the conspirators is to associate u with their key π. The conspirators possess the following powers, all of which are reasonable for any well-connected conspiracy:

C1. The conspirators can forge any form of physical identification, such as a driver's license, a passport, or a Social Security number. This is a corollary to A1.

C2. They are otherwise normal people. The conspirators are regular users of the e-mail systems; they are otherwise trustworthy, and other people trust them.

C3. They have whatever resources (financial, computing, etc.) they might need.

C4. They *cannot* compromise any computers or computer-stored data belonging to u. An unprotected private key can obviously be compromised; it is interesting to prove that one's identity can be compromised regardless of the precautions taken. By the previous assumption, though, they have access to systems beyond the control of u.

The conspiracy attack may seem improbable, and it is probably true that it would not occur to any large extent. It is worth discussing, though, because it reveals a mechanism by which one could send spam without fear of responsibility. If the conspiracy attack is possible, then people could send spam with their own keys and later deny responsibility for those messages on the grounds of a conspiracy attack—spammers might even launch a conspiracy attack *on their own keys*, so that they are entirely free of responsibility.

5.1.2 Two Forms of Authentication: Distributed and Hierarchical

If a database tells a recipient r that q is possessed by some sender s, then r has no reason to believe that s actually possesses q. The database must provide, in addition to the entity possessing the key, some justification or proof that the entity is in fact the owner of that key.

If the recipient is unwilling to trust *anyone*, then the only way to prove to r that entity s possesses the quality q is for r to actually observe that quality (or its effects). However, if r is willing to trust some set of entities T_r, then any of those entities can verify that $q \in s$, and then r will accept that $q \in s$. And entities in T_r may trust a set of entities T_{T_r}, and so forth. So the database can provably show r that s possesses q only if either (1) r can observe that s possesses q or (2) the database can provably show that some other trusted entities have observed that s possesses q. By A1 and A3 the first of these is not always feasible.

When an entity e observes a quality q in s, e can provide the database with a documented proof of having verified that quality, so that others can observe that e has vouched for $q \in s$. This documented proof is called a *signature* in this paper.

Let the set of users of the system be P. One possibility is any member of P can provide signatures for members of P. If this is not the case, then some subset of "super-entities," S, must sign for members of P. In the first case we have a *distributed authentication* system, in which peers sign for each other; in the second case we have a *hierarchical authentication* system, in which higher authorities sign for people.

5.1.3 Distributed Authentication

Distributed authentication is epitomized by the PGP Web of Trust scheme, in which keyholders rely on signatures of other keyholders [45]. The lack of central authorities gives distributed key infrastructures the advantage of not having central points of failure.

In any distributed key infrastructure, there is one central question: given a key k, who will sign it? There are two possibilities: first, the system can stipulate some set of people to sign k; second, the system can allow the the holder of k to choose signers at will.

In the first case, given some entity e, the system chooses a set of people S_e who may sign the key k_e belonging to e. Then e must allow members of S_e to observe that $k_e \in e$, and those members produce the appropriate signatures for k_e. There are two problems with this system. First, e has no reason to trust any member of S_e, and none of the members of S_e have any reason to trust e, based on A1. Second, unless the rule is based on geography, members of S_e may be far away from e, and by A3 it is unreasonable to expect e to travel to visit members of S_e, or vice versa. And if the rule is geographic, then if the conspirators happen to live near our hypothetical user u, then they could easily produce false but accepted signatures that key π belongs to u.

The second case of a distributed system allows any entity to sign any other entity's key. In this system, the key p belonging to our user u is trusted because u has other users sign p, and those users are trusted by other members of the system. But by C2, the conspirators are *also* trusted by other members of the system, so their false signatures on π must be accepted if the signatures on p are accepted. In this way, the conspirators can cause π to be falsely associated with u.

Since these two cases comprise the entire space of distributed authentication, the above demonstration shows that such infrastructures either make the system impossible to use or vulnerable to the conspiracy attack, so distributed authentication is insufficient for the needs of an e-mail system.

5.1.4 *Hierarchical Authentication*

The paradigm of hierarchical authentication is the x.509 certification authority architecture [21]. In general the system provides a guarantee of identity to domain name servers, and some have suggested simply extending the hierarchy to users within a domain, thus equipping every user with a signed key pair [17].

By A4 the system must be able to revoke the key p belonging to our user u without the assistance or consent of u. This means, by C1 and C3, that the conspirators can force a revocation of p and then replace it with π. One might object on the grounds that only certain officials would be permitted to perform such an unassisted revocation. However, by C3, the conspirators may have access to the resources of those officials, thus allowing them to revoke the key. Even worse, the conspirators might *be* those officials. Possibly these seem highly unlikely, but recall that the conspiracy attack is only proposed because a spammer could use it to deny responsibility. The officials need not be corrupt or compromised; the spammer merely needs to *accuse* them of being such.

Hierarchical authentication is successful because it is *static*: keys are generally not compromised, so few revocations are necessary. Current implementations of certificates reveal this rarity of revocations: in order for a certificate to actually be revoked on the system, one must resort to inelegant solutions such as certificate revocation lists. In contrast, with its large number of users, e-mail is a *fluid* system: entities frequently change addresses, lose passwords or other identifying qualities, and become victims of identity theft. In this sort of system, revocations would be frequent. It is this disparity between the static nature of hierarchical authentication and the fluid nature of e-mail and messaging systems that makes the former system unsuitable for the latter.

FIGURE 5.2 Summary of the analysis of the authentication-based AST.

Assumed definition of spam If a sender is unwilling to take responsibility for sending a message, then the message is spam.

Potential errors An entity's identity may be forged or compromised, causing legitimate mail from that entity to be considered spam and/or spam from that entity to be considered legitimate.

Successful areas If it were possible to implement a provably secure authentication system, then this system could theoretically eliminate all spam, since appropriate retributive actions could be taken against spammers.

Potential attacks An adversarial conspiracy could compromise the identity of some entity and use that identity to send spam. Alternatively, an adversary could simply send spam and dismiss responsibility on the grounds of a conspiracy attack.

5.2 Conclusions on Identity Verification by Authentication

As demonstrated above, authentication infrastructures are not useful for performing identity verification of e-mails. This is not surprising; the problem of identity in philosophy is prominent and still unsolved, and identity on computer systems is also a complicated subject of research [34]. The SDSI authentication architecture by Rivest and Lampson, for example, acknowledges that a global namespace of entities (e.g., a database of keys) is a practical impossibility, so their architecture uses keys only to give access to resources, not to identify entities [36]. In a system as blind as distributed messaging it is difficult, if not impossible, to identify members on the system with any strong guarantee of assuredness.

Since the fundamental problem with authentication systems is that they cannot handle the large, volatile number of users, one might suggest providing keys only to a small subset of entities not likely to be compromised, such as large corporations, allowing them to send out authenticated messages. Naturally, this would not be a solution for everyone, but it would suffice for at least those individuals.

If an identity verification AST is to be useful for everyone, though, it must be able to guarantee the identity of any sender. Authentication methods, which require a publicly accessible, volatile database of keys, cannot provide this guarantee.

5.3 Immediate Responsibility Identity Verification

The problem with authentication systems was the reliance on an external database and the need to maintain it. As a result, a system of identity verification that did not rely on external data would be ideal. What exactly would this sort of system look like?

We defined immediate responsibility systems as systems that employ direct rules, rules specifying a sender quality that the recipient can observe entirely during the sending transaction. This definition is sufficient to prove two aspects of immediate responsibility systems.

THEOREM 7. *In an immediate responsibility system, the rule must be that the sender exists and is present to the recipient during the transaction of sending the message.*

Proof. The recipient r must have some rule $\langle m, q \rangle$ and then observe the quality q in the sender s. By our definition, q must be observed entirely during the sending transaction. But r can only observe two types of qualities in s during the transaction. First, s could perform some action, and then r would have observed the quality of s having performed that action. In that case, the system is a proof of intent system, by the definitions provided in Section 4.2, not an immediate responsibility system. So then s cannot perform any sort of action, and the only quality that r can observe is the *existence* of s and the fact that s is present and conversing with r during the transaction. Since existence is the only quality of s that r can observe, q must be the quality of existence, and $\langle m, q \rangle$ must be the rule stated in the theorem. ■

We need not stop here. The definitions let us infer not only the rule employed by immediate responsibility systems but also the behavior of the recipient.

THEOREM 8. *In order for an immediate responsibility system to be effective, the recipient must read the message during the transmission of the message (the transaction).*

Proof. By the previous theorem, the only quality of s that r can observe is the quality that s is present during the transaction. This is a temporal quality, and once the transaction has completed, s fails to have that quality thereafter. So r can only identify s during the transaction. But if the message sent is spam, then in order for r to take retributive action against s, r must be able to identify s. So r must know whether or not the message is spam before the transaction terminates, so r must read the message during the transaction. ∎

Compare this situation to an open marketplace or bazaar, in which a buyer wishes to purchase a fruit from a street vendor. If the buyer determines that the fruit is spoiled while the seller is watching, then the buyer can hold the seller responsible and return the fruit immediately, even if the buyer doesn't know the seller's name, address, or other identifying information. In contrast, if the buyer checks the fruit days later, then the seller may be long gone by the time the buyer tries to establish responsibility for it. In an immediate responsibility AST, if the recipient reads the message while the sender is delivering it, then the recipient can immediately hold the sender responsible if that message is spam.

This idea of immediate responsibility for message content is the basis for both the Internet Mail 2000 protocol and the Message Distribution Protocol [6, 8]. if s wishes to send mail to r, the mail is retained on the computer of s rather than being immediately delivered to r, and r retrieves the message by contacting s and retrieving the message. If r believes that the message is spam, then the message can be forwarded to an impartial arbiter who can investigate *s*, inspect the computer that retained the message, and so forth.

One disadvantage of this sort of system is that, in order for the recipient to retrieve the message, the sender must be available. Mailing lists and large organizations most likely have an always-available server, but an individual sending a personal e-mail would be forced into the difficult situation of having to remain constantly available until the recipient retrieves the message. Additionally, individual people are less likely to have static network addresses, making it difficult for the recipient to locate the sender to retrieve the message. One possible workaround is for individuals to delegate their messages to a high-uptime relay computer. The computer administrators would determine that the message is not spam and then make it available to the recipients. The recipients would then hold the computer administrators responsible if the message were spam, and in this case the responsibility would be legitimate because the administrators should have verified the message content already.

The second flaw with this sort of system is the same as the flaw we saw in the first type of system we analyzed: the verification identifies the sending computer, but not the sender.[1] It is conceivable that an adversary could use compromised computers to send spam, causing the computer's owner to be held responsible. However, this is not a problematic flaw with the system. Since the compromised computer is identifiable, any remaining spam on it can be easily identified and removed. As a result, only a few spam messages will actually be successfully delivered, so an adversary wishing to send a million spam messages would need to compromise on the order of a million computers, which is a fairly unreasonable goal even

[1]The author is reminded of the penultimate scene in the film *Monty Python and the Holy Grail* (1975), in which the protagonists find, after their arduous quest, the castle containing the Holy Grail, only to discover that it is owned by the insulting Frenchman from the beginning of the movie.

FIGURE 5.3 Summary of the analysis of the immediate-responsibility AST.

Assumed definition of spam If a sender is unwilling to take responsibility for sending a message, then the message is spam.

Potential errors A spammer could compromise another person's computer and have that computer send spam, although in the case of this sort of system, once the spam is detected the "sender" can be quickly identified and the remaining spam can be eliminated.

Successful areas All sorts of messages, in particular bulk mail, can be successfully sent on this system, and spam can be easily eliminated by the method described above.

Potential attacks Attacks would be difficult; the spammer would have to compromise a large number of computers and hope that enough messages are read before the message is detected and removed.

Note Since the sender has to be available for the recipient to receive the message, then individuals who are not consistently online may find this sort of system inconvenient.

in light of the success of recent viruses [37].

A third flaw with this system deals with what the recipient can actually do upon receiving a spam message. If the recipient reports the message to the authorities *after* the transaction has completed, then by the logic in the above theorems, the authorities will be unable to identify the malicious computer! So it would seem necessary for the transaction to be kept alive *until the authorities can identify and inspect the sender*, which would present an inherent design difficulty for any immediate responsibility AST. It should be noted that although both the Internet Mail 2000 proposal and the Message Distribution Protocol proposal specify protocols for transmitting legitimate messages, neither provides a mechanism for reporting spam messages.

Chapter 6

Conclusions

We began by considering two types of AST: those that do not require the sender to perform some intervening action and those that do require some action. We then showed that, of those AST's that do not require sender intervention, there can only be two types, network identity filters and message content filters. Neither of these two systems can sufficiently control spam from being sent, as both of them are vulnerable to attack.

We proved that systems that require sender intervention must fall into one of two categories if they are to successfully deal with spam: either they must require the sender to provide a proof of sending intent or they must verify the identity of the sender. We showed that a proof of intent is equivalent to the sender's performing some costly action in order to demonstrate to the recipient the utility of the message being sent. Systems that demand the sender to pay a cost for sending seem like a promising solution if the cost is transient and non-monetary. However, if the cost is predetermined, then either mailing lists will be unable to pay the cost of sending messages or spammers will be able to send large volumes of spam using distributed network attacks. Cost-based systems in which the *sender* takes a part in choosing the cost, such as the Attention Bond Mechanism of Loder *et al.*, are especially promising solutions, but such systems might be difficult to implement without the use of monetary, or at least permanent, costs.

Finally we considered systems that deter spam through identity verification. Such systems come in two flavors: those that identify the sender through the use of an external database, and those that identify the sender immediately during the transaction. Systems that require an external database suffer from the inherent fact that the database must be constantly changing, so they are open to attack and thus cannot provide a strong guarantee of the sender's identity. Systems that verify the sender's identity during the transaction require the sender to be available and present while the recipient receives and reads the message, so a difficult burden is placed on the sender. Nevertheless, such systems would provide the necessary guarantee of identifying the sender and thus controlling spam.

6.1 A Proposed Solution

This paper would not be a paper on anti-spam technology if it did not present some proposed AST. Unlike other proposed solutions, though, this one will not be a new solution, but rather an innovative combination of two different types of solutions that we discussed in this paper, taking the strengths of each in the situations in which they are strong.

As we mentioned in Section 4.4, we could consider an analogue of the postal model in which small quantities of messages can be sent with a relatively high cost, but large quantities can be sent at a lower cost but with stronger verification of the sender identity. We can employ any of the standard cost functions for low-cost messages, such as number hashing. The cost per message is prohibitively high so that the maximum number of messages any single computer could send in a reasonable amount of time is on the order of 30; this means that millions of computers would need to be compromised to send a spam message to a reasonably large audience. Personal communications, on the other hand, would generally be easily sendable under this constraint.

Since sending a large quantity of messages is desirable sometimes, we can employ either of the identity verification schemes, as both of them thrive in the bulk mail environment of few potential senders each sending large quantities of messages. As discussed earlier, we could simply equip bulk mailers with signed keys, and as long as the keyholders are few in number and follow good security practices, their keys should rarely need to be revoked. Alternatively, bulk mailers could use an immediate responsibility scheme, in which they would store messages on their own servers for the recipients to retrieve. Chances are, anyone sending bulk mail probably has an always-available server to use for storing those messages.

How would a spammer send messages on this system? Since the spammer is willing to be identifiable, the spam must be sent using the cost-based alternative. However, the cost is prohibitively high, so the spammer would only be able to send a relatively small number of messages. The degree to which spam is controlled on the system, then, can be adjusted by simply raising or lowering the cost per message in the cost-based sending alternative; legitimate bulk mail would not be affected because it can be sent using the identity verification scheme.

This solution is an example of combining two AST's together to create a much more secure solution. With the taxonomy of AST's presented in this paper, we can investigate other possible combinations of solutions to come up not with ad-hoc combinations but rather reasoned solutions that actually take into account the strengths and weaknesses of each part.

6.2 Final Remarks

In this paper we have divided the problem space of anti-spam technologies, or AST's, into a taxonomy of solution categories, and we have considered the advantages and disadvantages of each solution. In doing so, we have demonstrated that some concepts, such as content filtering, depend on unreliable definitions of spam, and others, such as public-key authentication, are infeasible for the large scale of the e-mail community.

The approach taken in this paper is unique in two respects. First, the theoretical approach to the spam problem, considering the design space of all possible solutions rather than just those particular solutions that have been proposed or implemented, makes the conclusions drawn about systems more universal and independent of the state of the art. The use of proof-based logic and separating tests allowed for this complete exploration and total coverage of the solution space. Second, the analogies with non-computer systems, in this paper the postal system, provide insightful comparisons that can help us understand the nature of the problems with computer systems, which are less mature.

Successful systems come not out of flashes of insight or genius; they are rather the product of reasoning and analysis over time by many hard-working minds. By employing a theoretical approach to the area of anti-spam research, it is the aspiration of this research to provide a foundation for such reasoning and analysis, in hopes of developing systems that are successful in this field.

Bibliography

[1] Martin Abadi, Mike Burrows, Mark Manasse, and Ted Wobber. Moderately hard, memory-bound functions. In *Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2003.

[2] George Akerlof. The market for lemons: Qualitative uncertainty and the market mechanism. *Quarterly Journal of Economics*, 84:488–500, 1970.

[3] Anti-Spam Research Group, March 24, 2004. URL http://asrg.sp.am/.

[4] Giuseppe Ateniese and Stefan Mangard. A new approach to DNS security (DNSSEC). In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 86–95, Philadelphia, PA, USA, 2001. ISBN 1-58113-385-5.

[5] Steve Atkins. Size and cost of the [spam] problem. In *Proceedings of the Fifty-Sixth Internet Engineering Task Force*, March 2003. URL http://word-to-the-wise.com/asrg.htm.

[6] Steve Atkins. Giving congrol over email to the recipient: The Message Distribution Protocol, March 24, 2004. URL http://word-to-the-wise.com/message_distribution.

[7] Adam Back. Hashcash: A denial of service countermeasure. URL http://www.cypherspace.org/hashcash/. Only available electronically, August 2002.

[8] Daniel J. Bernstein. Internet mail 2000, March 24, 2004. URL http://cr.yp.to/im2000.html.

[9] Brightmail Inc.: The Anti-Spam Leader, March 24, 2004. URL http://www.brightmail.com.

[10] Cynthia Dwork, Andrew Goldberg, and Moni Naor. On memory-bound functions for fighting spam. In *Advances in Cryptology (CRYPTO 2003)*, Santa Barbara, CA, USA, October 2003.

[11] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology (CRYPTO 2003)*, Santa Barbara, CA, USA, August 1992.

[12] D. Eastlake. Domain name system security extensions, March 1999. RFC 2535.

[13] Gergely Erdelyi and Ero Carrera. F-secure virus descriptions: Bagle. *F-Secure Virus Information*, January 2004. URL http://www.f-secure.com/v-descs/bagle.shtml.

[14] Deborah Fallows. *Spam: How it is Hurting Email and Degrading Life on the Internet*. Pew Internet & American Life Project, Washington, D.C., USA, October 22, 2003.

[15] Federal Trade Commission. *National and State Trends in Fraud & Identity Theft*. January 22, 2004.

[16] FTC Division of Marketing Practices. *False Claims in Spam*. Federal Trade Commission, April 30, 2003.

[17] James M. Galvin. Public key distribution with secure DNS. In *Proceedings of the Sixth USENIX UNIX Security Symposium*, San Jose, California, USA, July 1996.

[18] John Graham-Cumming. The spammer's compendium. In *Proceedings of the 2003 MIT Spam Conference*, Cambridge, MA, USA, January 2003. URL http://www.jgc.org/tsc.

[19] Harvard University FAS Computer Services, March 2004. Personal communication.

[20] Shane Hird. Technical solutions for controlling spam. In *Proceedings of the Australina UNIX and Open Systems User Group*, Melbourne, Australia, September 2002.

[21] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, April 2002. RFC 3280.

[22] David Hume. *Enquiries Concerning Human Understanding and Concerning the Principles of Morals*. 3rd edition, 1975.

[23] Edward Hurley. In the spammer's lair. *TechTarget*, July 16, 2003. URL http://searchsecurity.techtarget.com/originalContent/0,289142,sid14_gci914837,00.html.

[24] International Organization for Standardization (ISO). *Information technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*, 1994. ISO 7498-1.

[25] Paul Judge. A taxonomy of anti-spam systems. In *Proceedings of the Fifty-Sixth Internet Engineering Task Force*, March 2003. URL http://www.ietf.org/proceedings/03mar/slides/asrg-6/.

[26] John R. Levine. Technical approaches to spam. In *Federal Trade Commission Spam Forum*, May 2003. URL http://www.ftc.gov/bcp/workshops/spam/Presentations/levine.pdf.

[27] Thede Loder, Marshall Van Alstyne, and Rick Wash. Information asymmetry and thwarting spam. In *Proceedings of the 2004 MIT Spam Conference*, Cambridge, MA, USA, January 2004.

[28] Karl-Gustaf Löfgren, Torsten Persson, and Jörgen W. Weibull. Markets with asymmetric information: The contributions of George Akerlof, Michael Spence and Joseph Stiglitz. *Scandinavian Journal of Economics*, 104:195–211, 2002.

[29] Mail Abuse Prevention System, LLC, March 24, 2004. URL http://mail-abuse.org.

[30] Mail Abuse Prevention System, LLC. Getting into the MAPS RBL, March 24, 2004. URL http://mail-abuse.org/rbl/candidacy.html.

[31] Mailshell™: The OEM Anti-Spam Leader, March 24, 2004. URL http://www.mailshell.com.

[32] Majordomo Lists at vger.kernel.org, March 2004, 2003. URL http://vger.kernel.org/vger-lists.html.

[33] Mylene Mangalindan. Spam queen: For bulk e-mailer, pestering millions offers path to profit. *The Wall Street Journal*, November 13, 2002.

[34] Petros Maniatis and Mary Baker. A historic name-trail service. In *Proceedings of the Fifth IEEE Workshop on Mobile Computing Systems and Applications*, Monterey, CA, USA, October 2003.

[35] Patrick Pantel and Dekang Lin. Spamcop: A spam classification & organization program. In *Learning for Text Categorization, Fifteenth National Conference on Artificial Intelligence*, Madison, WI, USA, July 1998.

[36] Ronald L. Rivest and Butler Lampson. SDSI–A simple distributed security infrastructure. URL http://theory.lcs.mit.edu/~rivest/sdsi10.html. August 1996.

[37] David S. H. Rosenthal, Petros Maniatis, Mema Roussopoulos, T. J. Giuli, and Mary Baker. Notes on the design of an internet adversary. *Adaptive and Resilient Computing Security Workshop*, November 2003.

[38] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. In *Learning for Text Categorization, Fifteenth National Conference on Artificial Intelligence*, Madison, WI, USA, July 1998.

[39] Stuart Shieber, March 2004. Personal communication.

[40] Spam Prevention Early Warning System, March 24, 2004. URL http://www.spews.org.

[41] SurfControl. Public opinion strategies survey. URL http://www.surfcontrol.com/resources/surveys/SpamSurveyOnePagerFinal.pdf. February 2003.

[42] Theo van Dinter, Duncan Findlay, Craig Hughes, Justin Mason, Dan Quinlan, Matt Sergeant, and Malte S. Stretz. Spamassassin, March 24, 2004. URL http://www.spamassassin.org.

[43] Luis von Ahn, Manuel Blum, and John Langford. Telling humans and computers apart automatically. *Communications of the ACM*, 47(2):56–60, 2004. ISSN 0001-0782.

[44] William S. Yerazunis. The spam-filtering accuracy plateau at 99.9% accuracy and how to get past it. In *Proceedings of the 2004 MIT Spam Conference*, Cambridge, MA, USA, January 2004.

[45] Philip Zimmermann. *The Official PGP User's Guide*. MIT Press, May 3, 1995.

Colophon

This document was set entirely in Adobe Minion. The text was typeset using the LaTeX document preparation system. The graphics were prepared using the *PSTricks* package by Timothy Van Zandt, with the exception of Figure 2.1, in which the diagrams were coded in the Adobe PostScript page description language.

The document was prepared on the Harvard University NICE servers, running Redhat Linux, and the terminal sessions were conducted on an Apple PowerBook G4.

No Microsoft products were used in the production of this thesis.