Sender Reputation in a Large Webmail Service

Bradley Taylor
Google Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
bat+ceas@google.com

ABSTRACT

In this paper, we describe how a large webmail service uses reputation to classify authenticated sending domains as either spammy or not spammy. Both SPF and DomainKey authentication are used to identify who the sender of the mail is. We describe a simple formula for how we calculate the reputation and how it is used to classify incoming mail. We show in general how domains, both good and bad, get distributed among the various reputation values, and also show the reputation values for a few specific domains. We describe some of the problems associated with this reputation system, and propose some recommendations for senders to avoid those problems.

1. INTRODUCTION

Google's *Gmail* is a free email service, built from the ground up, that launched as a beta in April 2004. In the beta period, which we are still in, we have worked diligently to improve Gmail's spam-fighting abilities. Though we are by no means done (can you ever be done with spam-fighting?), we've made significant progress to the point that today we have very low false positive and false negative rates. There are a lot of different components that are responsible for our progress, and this paper is not a comprehensive discussion of Gmail's spam filter. However, a lot of the progress came from something we call the Gmail reputation service, and we'd like to share our experiences with it.

2. SENDER REPUTATION IN A LARGE WEBMAIL SERVICE

The basic idea of a reputation system is to try and classify mail based upon *who* is sending the email rather than on *what* the content of the email is. It turns out that most sites implement a rudimentary form of reputation today. We'll discuss that briefly, then we'll talk about a more advanced system that we use with Gmail. Throughout this document we will use the terms *spam* to refer to unsolicited or unwanted mail and *nonspam* to refer to solicited or wanted mail.

2.1 Rudimentary Reputation

This is the system used at many, if not most, sites today. Spam filters make a judgement of the message, but they can be pretty cavalier about what they choose to put in the spam folder, no matter how good the sender may be. To correct these false positives, whitelists are installed for some good senders to avoid spam filter mistakes. Typically, these whitelists are lists of IP addresses. In addition, many services use block lists ahead of the spam filter that result in all mail from an IP address either getting rejected or going to a spam folder.

When you think about this in general, you have a system that works as follows:

1. Use the connecting IP address to represent the sender. The IP address is a crude form of authentication.

2. If the sender is in the whitelist, put all messages from that sender in the inbox.

3. If the sender is on a block list, either reject all messages from that sender, or put them in a spam folder.

4. Otherwise, send the message to a statistical spam filter that makes a final judgement.

One problem with the rudimentary system above is that it is too manual. Block lists can be automated with spam traps to acquire new entries, but removing false positives is often a manual process. Whitelists are even harder to manage. It is not possible to trap good mail, after all, since good mail is often very personal and doesn't run wild into random inboxes as spam does.

Also, if you claim to support manual whitelists, you will have no end of bulk senders calling you and asking to be whitelisted. Not all of them are good senders, though they all think they are, so you can't just add the whitelists without first understanding the impact on the users.

A second problem with the rudimentary system is that IP addresses are just not a very good way to authenticate. The same sender may not always have the same IP address. Also, the IP address that connects to send the mail is not always the true sender because of forwarding. It is a hard problem to figure out who the true sender is because any IPs in the "Received" header cannot be authenticated reliably. Finally, some domains may share a set of IP addresses, but only one domain is actually spamming. That can ruin the reputation for all the other domains.

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

2.2 Gmail's Reputation System

Now, we'll go on to describe what we think is a better system, using more advanced authentication and automated reputation lists. But before going into that, it's good to have a definition of what it is we're trying to get rid of first.

2.2.1 Defining spam for automation

Not everyone agrees what spam is, so there end up being many definitions of it. Some will say it is any form of unsolicited mail. Some will say only unsolicited bulk mail. Or some will say it is any mail they didn't want, even if they were subscribed. For Gmail, we wanted a definition that could be automated, because the volume of our mail is high and anything manual would be too expensive. But detecting that an email was solicited in an automated way is next to impossible. Even detecting that something is bulk is difficult. Spam may look all the same to the users, but the spams often come from a large zombie network, with small variations in the content to confuse spam filters.

The easiest definition for us to use is simply unwanted email. In the Gmail inbox, there's a button called "Report Spam" that users can use to tell us they didn't want this email sent to them. There's also a button in the spam folder called "Not Spam", where they can tell us they actually wanted this email. These buttons provide input to our spam system so that we can make a better decision next time. This isn't a perfect way to define spam and nonspam, but it is one that can be made to work easily in this context. We will discuss some of the problems this definition causes in a later section.

2.2.2 Authenticating the domain

Web-based email services, such as Gmail, make it easy to solve the problems with block lists and whitelists that were mentioned earlier. We have nearly perfect information about how spammy various IP addresses are based upon how often users mark and unmark spam. So, that's what we use. We don't need manual whitelists. If you want your mail to get through, just authenticate and behave yourself, and we'll take good care of you. And if you misbehave, we'll know that too, of course, and take "care" of you also.

But, as mentioned already, IP addresses don't really represent the sender in the way we'd like. Domain-based authentication systems, such as SPF [11] and DomainKeys [2], help with that. SPF lets us know which senders are tied to which IP addresses, so we don't have to worry about IP address changes by senders. A lot of senders use SPF, and for the ones that don't, we use *best-guess* SPF [8]. Best-guess SPF is a simple default record that assumes that a domain is authenticated if the sending IP comes from the same range of IPs as the A or MX records, or if the sending IP's reverse DNS name matches the domain claimed in the email. To get even more mail authenticated, we'll try one more thing if both plain and best-guess SPF fail: if the sender is a subdomain of the DNS PTR's zone, we'll authenticate it as if the sender claimed to come from the zone itself. For example, assume a message with the envelope sender *subdomain.example.com* comes from an IP with a DNS PTR of *host.example.com*. It won't authenticate with best-guess SPF. However, if we pretend the envelope sender is *example.com*, it will authenticate with best-guess SPF.

Table 1: SPF Authentication Method Breakdown

Method	Frequency
Plain SPF	52%
Best-guess SPF	38%
PTR zone	10%

Table 2: Authentication Breakdown: Nonspam

Method	Frequency
Both SPF and DomainKeys	20%
SPF only	53%
DomainKeys only	1%
Not authenticated	26%

Table 1 shows the breakdown of each method used to SPF-authenticate incoming email. You can see from it that roughly half (52%) of the SPF authenticated messages arriving into Gmail are authenticated by using plain SPF. Best-guess and the PTR zone trick lets us authenticate almost twice as much email as otherwise. These methods are only temporary measures until more domains get authenticated explicitly. We know it isn't perfect. Through the rest of this document, when we refer to SPF, it refers to our modified form of it.

But we should mention forwarding. A significant number of emails sent to Gmail are forwarded, since many of our users had existing email accounts before and want to continue receiving at their old email address. SPF doesn't work with forwarding very well, so without it, we're left with a lot of unauthenticated mail. DomainKeys helps with this problem, and Gmail was one of the first large email service providers to implement it. Unfortunately, there is no "best guess" DomainKeys. The only thing we can do is encourage people to sign their mail, and we hope we have helped a bit in that regard by being an early adopter of email signing.

SPF and DomainKeys are both good techniques: one does not replace the other. We encourage senders to implement both. That way, if for some reason one or the other breaks due to forwarding or message-munging, we can still authenticate with the other one. And if both are present, it is that much stronger of an authentication signal.

Table 2 shows the breakdown of how much of Gmail's incoming mail is authenticated, and by which methods. You'll notice that most of our nonspam is authenticated, about 75%.

Table 3 shows the same breakdown, but for spam this time instead. You'll see here that most of our spam is *not* authenticated, only about 40% is.

2.2.3 Calculating the reputation

As email arrives, it is classified by Gmail as either spam or nonspam. We log an event indicating what the classification was, along with any authentication associated with the

Table 3: Authentication Breakdown: Spam

Method	Frequency
Both SPF and DomainKeys	1.5%
SPF only	39%
DomainKeys only	0.5%
Not authenticated	59%

message: IP address, SPF domain or DomainKey domain. And any time a message is reclassified manually, that is the user hits "Report Spam" or "Not Spam", we also log a similar message.

The event logs are processed to update the reputations for each of the items logged. There are four variables involved:

autospam: How many times mail from this sender went into the spam folder automatically.

autononspam: How many times mail from this sender went into the inbox automatically.

manualspam: How many times a user marked a message from this sender as spam.

manualnonspam: How many times a user marked a message from this sender as nonspam.

As each entry is processed, the above variables are updated for the item in question, for example, the SPF domain reputation of *weliketospam.com*. The reputation is calculated as a number between 0 and 100, where 0 is the most spammy, and 100 is the most nonspammy. You can think of the reputation as the probability that a given sender's mail is not spam.

A simplified version of the formula is the following:

$$good = autononspam + manualnonspam - manualspam \quad (1)$$

$$total = autospam + autononspam \quad (2)$$

$$reputation = (100 * good)/total \quad (3)$$

The formula is a little more complicated in reality. You have to avoid dividing by zero and out of range issues. If *total* is zero, then there's no reputation. Otherwise, the more complicated version is this:

$$manualnonspam2 = min(autospam, manualnonspam) \quad (4)$$

$$manualspam2 = min(autononspam, manualspam) \quad (5)$$

$$good2 = autononspam + manualnonspam2 - manualspam2 \quad (6)$$

$$reputation = (100 * good2)/total \quad (7)$$

To make it clear with an example, let's say *weliketospam.com* sends 100 spams to different users in Gmail and 60 end up in the spam folder automatically, the other 40 in the inbox as false negatives. At that point, the reputation of *weliketospam.com* is 40. Later on, let's say 30 users mark the missed spams manually. Now, the reputation drops to 10.

For a positive example, let's say *weneverspam.com* sends 100 messages and 5 unfortunately end up with the spam folder as false positives. The reputation at this point is 95. Later on, let's say 3 users unmark the false positives. Now, the reputation is 98.

Another way to think of the reputation is as a potential false-positive rate. For example, if a particular sender has a reputation of 2 and you decide to threshold senders below a reputation of 5 as deserving to go to the spam folder, then about 2% of the time you'll have false positives on that sender's mail.

The reputations are calculated over many days. A domain which doesn't spam can build up a very solid reputation, so that a little blip of spam that may come from it from time to time is absorbed as noise by the generally good reputation. The flip-side of this is that a domain that does spam for a long time, but cleans up its act later, will take a few days to recover.

There are some refinements that can be made to improve the system. One problem is that some users don't log in very often or at all, so not everything gets voted on or in a timely enough manner. To fix that, the reputation system only uses a subset of the users: the ones we think will provide the best information. For example, users who never report spam or nonspam are not providing any feedback and their information should not be used. For reputations to self-correct, it is important that user input in both directions is fed into the formula. Only feeding in user spam reports, for example, provides no way for the reputation to adjust upwards in case of spam filter mistakes.

Some users get a lot of mail and some get very little. It wouldn't really be fair to the light users if the heavy users dominated the results. To fix that, we limit how much say any user has on a given item. For example, if a heavy user reports 100 spams against *weliketospam.com* in a given hour, we limit them to just one spam report against the domain for that hour. But they will get another vote the next hour, so they get up to 24 votes per day.

2.2.4 Using the reputation

We use the reputation in a similar way to how we explained rudimentary reputation earlier.

1. We authenticate the sender with SPF and DomainKeys, and using the domain we authenticated, calculate the reputation. For reputations above a threshold, the sender is considered good and reputations below a certain threshold, the sender is considered bad.

2. If the sender is considered good by reputation, put all messages from that sender in the inbox.

3. If the sender is considered bad by reputation, put all messages from that sender in the spam folder.

4. Otherwise, if the reputation is unknown or somewhere

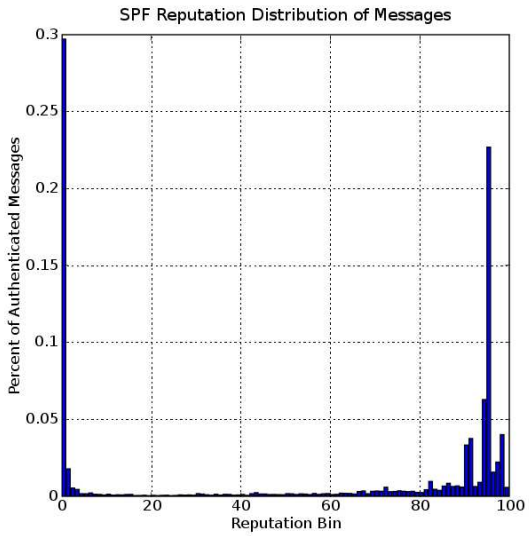

Figure 1: Distribution of SPF reputations

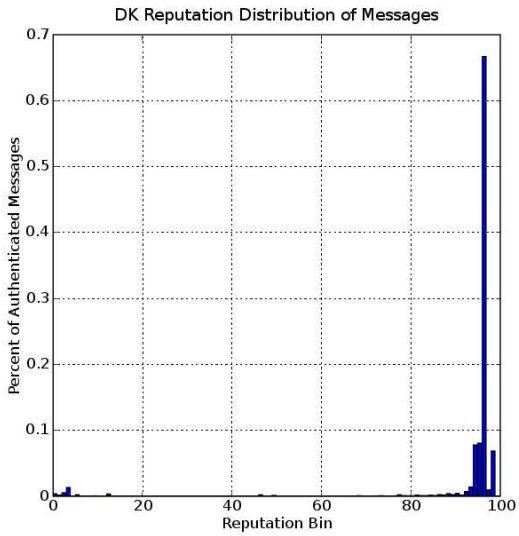

Figure 2: Distribution of DomainKey reputations

in the middle, send the message along with the reputation value to a statistical spam filter that makes a final judgement.

Once again, things are not quite as simple as that. Some domains may be very spammy for most Gmail users, but individual users may disagree about certain senders from that domain. So, we allow the user to override the general spam policy for individual email addresses from that domain.

2.3 Results

Figures 1 and 2 show the distribution of reputations for SPF and DomainKeys, respectively. You'll see from the figures that spammy domains get nicely clustered around a reputation of zero. We know these domains are spammy because if our users disagreed with us, they would unmark spam on them and the reputation would no longer be zero. Nonspam domains are more loosely clustered in the 90-100 range. There ends up being a smattering of domains in between. Many of the messages in the in-between category are from legitimate bulk senders, and the lower than 90 percent reputation is a reflection of less-than-ideal sending practices. But not all bulk senders are the same. Some have very high SPF and DomainKey reputations. For example, *eBay's* DomainKey reputation is 98.2, enough to be whitelisted by Gmail. This is the "holy grail" of bulk sending, and it is all done without requiring any payment or extra effort on the part of the sender other than just having good mail hygiene.

Tables 4 and 5 show a few selected senders and what their SPF and DomainKey reputations are. We are not identifying the actual names of spam domains here for fear that they'll change their ways once they know how easy it is for us to detect them. In general terms though, the spammers listed are trying to sell things like pre-approved credit cards and "free" iPods. For this table, we chose three spam domains that authenticate with both SPF and DomainKeys, so they are consistent between the tables.

One thing to note from this is that non-free ISPs, like *AOL* and *Earthlink* are able to keep spammers off of their network pretty effectively, and thus end up with very high reputations. Free services, such as *Yahoo Mail* and *Hotmail* have a harder time with this, but still do a pretty good job.

Notice also the high reputation of *ebay.com* here. This is mainly a bulk sender, yet they maintain a high reputation due to good sending practices. Many bulk senders unfortunately do not observe the same good practices as *eBay*, and as a result, have lower reputations.

There are some differences between the SPF and DomainKey reputations for a few of these domains. Speaking of *ebay.com*, it has an SPF reputation that is a bit lower than its DomainKey reputation. At the time of this writing, it appears that *eBay* is only signing their transactional email. The transactional mail would likely be more wanted by users than other mail from *ebay.com*, and this probably explains the difference.

Another example of a disparity is *earthlink.net*, with an SPF reputation about 5 points lower than its DomainKey reputation. Some of this may be due to forwarded spam from *Earthlink* users. Occasionally it will pass best-guess SPF authentication because the spammer forged an *earthlink.net* domain in the email.

2.4 Problems

No system is perfect, and this is no exception. One problem is some users forward their mail, including spam, to Gmail using tools which modify the envelope sender. *procmail* is an example of one of these tools. The forwarded mail gets authenticated, and since some of it is spam, it hurts the reputation of the forwarding domain unfairly.

Another problem is mailing lists, such as *Yahoo Groups*.

Table 4: Selected SPF domains and their reputations

Domain	Reputation
aol.com	98.5
earthlink.net	93.3
ebay.com	95.2
hotmail.com	95.4
yahoo.com	95.6
spamdomain1	1.7
spamdomain2	3.3
spamdomain3	2.5

Table 5: Selected DomainKey domains and their reputations

Domain	Reputation
earthlink.net	98.0
ebay.com	98.2
yahoo.com	95.0
spamdomain1	1.6
spamdomain2	3.1
spamdomain3	2.4

Yahoo Groups is signed with *yahoogroups.com* as the domain responsible, and generally speaking, it has a good reputation and is not a source of spam. However, some groups within it are spammy, and a blanket whitelist of *Yahoo Groups* will allow some spam through. There's a need to detect the spamminess of the individual groups more specifically.

A similar issue is that some corporate bulk senders may generally have a good reputation and be whitelisted, but once in a while they may get away with a bulk sending campaign that is unwanted by many recipients. However, at least these messages are authenticated and the recipients know where to complain. And if a domain does this too much, it runs the risk of establishing a long-lived bad reputation, or at least a less-than-good one.

Some users are lazy and find that reporting spam on a mailing list is easier than unsubscribing. In Gmail, this usually works for the user in that they will no longer get the emails for that list in their inbox anymore. But for the sender it means their reputation is hurt worse than it needs to be. There are ways for senders to work around this, by confirming with the user periodically that they are still interested in the list. A new standard to automate list unsubscription might be good also. At first glance, the URLs specified in the *List-Unsubscribe* [1] header seems promising, but they are pretty free-form and manual, and can't generally be automated. Plus, some people may not want spammers to know they exist, and unsubscribing would reveal that.

The problem with forwarded spam and best-guess SPF was alluded to earlier in the case of *earthlink.net*. A domain can make a policy decision not to publish SPF, yet we authenticate them anyway with best-guess SPF. This may unfairly give the forwarding domain a slightly lowered SPF reputation due to forwarded spam, which sometimes passes best-guess SPF. In the case of *earthlink.net*, it is not really a problem since their DomainKey reputation is still accurate and high.

It is theoretically possible to defeat a reputation system like this with mass registration of accounts and simulated email activity to earn a good reputation for an outside domain. However, if a spammer can mass register accounts, it is probably simpler for them to just send spam from those accounts instead. Given that free services already have to control the automatic abuse of accounts to prevent spamming, there may not be an actual problem here.

Reputation may not work for everybody. Having both "Report Spam" and "Not Spam" buttons is critical for the success of it, because it provides a way to self-correct to the right value. While most webmail providers have the capability to provide these buttons, most standalone email clients do not. Possibly the whitelists from large webmail providers could be shared in the public domain for these clients to use, but they would have no reporting ability.

2.5 Recommended Policies

To ensure that domains get the reputation they deserve, and in light of the above problems, we propose a few simple rules for senders.

1. Authenticate mail you send with both SPF and DomainKeys. Note that you may be implicitly implementing SPF already due to best-guess SPF.

2. Do not authenticate mail you are only forwarding, unless you have filtered out spam first. In other words, if you authenticate the mail as coming from you, you are taking responsibility for the quality of the resulting stream of mail.

3. Keep spammers and zombies off of your network, and observe good bulk sending practices yourself.

The second point may not necessarily be the best thing to do, but there aren't a whole lot of other choices and it does have the benefit of being simple. Gmail does not forward spam, so we can provide authentication information on all of our outgoing mail. This should make it easy for others to determine what our sending reputation is. This may not be an option for other sites though.

A possible better solution to the forwarding problem is that a standard agreed-upon header field could be added when forwarding mail indicating that the email looked like spam but is being sent anyway. Mail with that header does not need to enter into the reputation system. And since the receiving system knows it is spam based upon the standard header, it will know how to classify the mail without the help of a reputation system.

Gmail publishes guidelines for bulk senders [3], as do many other large email providers. We've found that senders who follow them generally end up with good reputations. The enforcement of the guidelines is by our users: if a sender is doing something bad such as using single opt-in instead of double opt-in, it simply translates into a poor reputation and a greater likelihood of ending up in the spam folder.

2.6 Other Reputation Systems

There are other reputation systems out there, but Gmail is the only one we know of that operates on the authenticated domain, rather than the IP address. The data that Gmail has is also more comprehensive, since we have almost complete information on who sends us mail and who is marking it spam or nonspam. It is difficult for third-parties to get that kind of complete information because receivers are not usually willing to share it. Also, the other systems typically only offer a binary spam or nonspam (whitelisting) value, rather than the continuous value from 0 to 100 that Gmail provides. But Gmail's data is private and, because of that, useful only for Gmail itself. It would be nice if a third-party service could provide something similar that everyone could use.

As an an example of a DNSBL reputation system, we can mention *SpamCop* [9]. It is fairly typical of a DNSBL using IP addresses to identify senders. *SpamCop*'s data is based partially on complaint data from users, along with trap hit data. This is balanced against a guess to the sender's total volume based upon lookups on the DNSBL for that sender. *SpamCop* only returns a binary value though indicating if they think the sender is a spammer. They do not tell you *how* spammy that IP address may happen to be.

There are some whitelisting services worth mentioning here that not really true reputation systems, but are rather accredidation or certification services. The difference here is that reputation is driven more by the receiver, whereas accredidation is driven more by the sender. In these cases, the sender has to apply to get on the "good" list, rather than this being determined automatically by the receivers, and often payment is involved. Services in this category include Goodmail's *CertifiedEmail* [10], Habeas's *Safelist* [4] and Return Path's *Sender Score Certified* (formerly known as Bonded Sender) [6]. These systems are not very comprehensive and only cover the few senders willing to apply for certification.

One system that comes reasonably close to Gmail's system though is Return Path's *Sender Score* [7]. This is not to be confused with the *Sender Score Certified* mentioned previously, though the names are unfortunately similar. *Sender Score* was not fully available to receivers at the time of this writing, but we've had some early access to it. If and when it becomes available, it will return scores for senders, represented by IP addresses, in the the 0-100 range, with 0 being the most spammy and 100 being the most nonspammy. These scores are partially based upon complaint data, and are percentile rankings rather than the independent signal that Gmail provides. Because *Sender Score* offers a continuous value, you can know just how spammy a sender is. This makes it useful for both blocking and whitelisting, and lets receivers decide their own thresholds or input the score into a statistical filter. Habeas has a system called *SenderIndex* [5] which may be similar, though not much was known about it at the time of this writing. It is possible that these systems could approximate the system that Gmail provides, but we haven't evaluated the data to know for sure. Using the authenticated domain, rather than the IP address though, would be a welcome improvement to these systems.

3. CONCLUSIONS

Reputation based upon the authenticated sender's domain works well for us. Spammers are easily identified, as are good senders. Only a few senders end up in the grey area in between, and for these we still have the traditional statistical filtering methods for classification. It is not without a few problems though, but we think they are small enough that they can be worked out eventually. We think reputation systems can work well for others too, and we hope to see more senders authenticating and observing good sending practices because of it.

4. ACKNOWLEDGMENTS

The author would like to thank the following people for contributing to this paper: Keith Coleman, Rob Siemborski, Rich Bragg, Misha Zatsman, Vijay Eranti and Yu Liao. The author would also like to thank the CEAS program committee for accepting this paper and for contributing valuable feedback to it.

5. REFERENCES

[1] J. Baer and G. Neufeld. The use of URLs as meta-syntax for core mail list commands and their transport through message header fields. RFC 2369, Internet Engineering Task Force, July 1998.

[2] Mark Delany. Domain-based Email Authentication Using Public-Keys Advertised in the DNS (DomainKeys). `www.ietf.org/internet-drafts/draft-delany-domainkeys-base-03.txt`.

[3] Google. Bulk Email Senders. `www.google.com/mail/help/bulk_mail.html`.

[4] Habeas. Habeas Safe List. `www.habeas.com/en-US/FAQ_Safelist.php`.

[5] Habeas. Habeas SenderIndex. `www.habeas.com/en-US/Receivers_SenderIndex`.

[6] Return Path. Sender Score Certified. `www.senderscorecertified.com`.

[7] Return Path. Sender Score Email Reputation Management. `www.returnpath.com/delivery/senderscore`.

[8] The SPF Project. FAQ/Best guess record. `new.openspf.org/FAQ/Best_guess_record`.

[9] SpamCop. What is the SpamCop Blocking List (SCBL)? `www.spamcop.net/bl.html`.

[10] Goodmail Systems. Certified Email. `www.goodmailsystems.com/certifiedmail`.

[11] M. Wong and W. Schlitt. Sender Policy Framework (SPF) for Authorizing Use of Domains in E-MAIL, version 1. `www.ietf.org/internet-drafts/draft-schlitt-spf-classic-02.txt`.