IPv6 Deployment and Spam Challenges

Spam has posed a serious problem for users of email since its infancy. Today, automated strategies are required to deal with the massive amount of spam traffic. IPv4 networks offer a variety of solutions to reduce spam, but IPv6 networks' large address space and use of temporary addresses — both of which are particularly vulnerable to spam attacks — makes dealing with spam and the use of automated approaches much more difficult. IPv6 thus poses a unique security issue for ISPs because it's more difficult for them to differentiate between good IP addresses and those that are known to originate spam messages.

Hosnieh Rafiee, Martin von Löwis, and Christoph Meinel
University of Potsdam

Since the Internet's beginning, spam has posed a serious problem to the IP network and to the Simple Mail Transfer Protocol (SMTP; RFC 5321). Spammers misuse considerable Internet resources to promote their products or fulfill their criminal objectives. Given this, governments have spent much time and many resources trying to resolve the spamming issue. Research in this area includes various approaches – such as Bayesian-based, content-based, DNS-based, or signature-based – and sometimes combines methods for a hybrid approach. Although such approaches are tried and tested in IPv4 networks, the feasibility of their use in IPv6 is questionable, particularly when it comes to DNS-based methods.

IPv6 was introduced to solve IPv4 network issues, focusing primarily on the lack of available addresses. Thus, IPv6 can support 2^{128} unique IP addresses, enough for each and every device or node attached to the Internet. The downside is that spammers can exploit this increased addressing scheme to send messages through many temporary addresses while doing denial-of-service (DoS) attacks on anti-spam firewalls or DNS systems.

Here, we briefly explain the history of spam and why it's such a serious issue. We also look at current spam-filtering techniques in use in the IPv4 environment and the open issues involved in using these techniques with IPv6.

What Is Spam?

Spam (for "spiced ham") is a precooked meat product made by Hormel Foods (www.spam.com). According to Wikipedia (http://en.wikipedia.org/wiki/Spam), the use of the word "spam" to indicate something both "ubiquitous and inescapable" originated in an old

1089-7801/12/$31.00 © 2012 IEEE

Monty Python sketch from the 1970s; the sketch took place in a restaurant where everything on the menu came with Spam. The term was later used to refer to unsolicited electronic messages sent to people who didn't choose to receive them. However, unsolicited messages sent to multiple accounts aren't always referred to as "spam"; see, for example, the occasional funny mass messages sent from friends to friends and back again. In such cases, the sender is known, whereas with spam, the sender is typically unknown.

In 1978, Gary Turk sent the first spam to about 400 people over Arpanet, the Internet's predecessor, to promote his line of new computers. Despite this early start, modern spam failed to take off until 1994, when two lawyers used spam to offer immigration services and ultimately netted US$100,000 in profit. This activity motivated others to jump on the spam bandwagon.

In 1997, the Nevada State Legislature passed the first antispam law because spam was overwhelmingly wasting expensive resources. The law required that

- mass mailers offer a procedure for recipients to remove themselves from mailing lists, and
- senders use their correct business names.

However, the law imposed no penalty for spammers who failed to comply.[1] In 2004, the US enacted the CAN-SPAM Act (for Controlling the Assault of Non-Solicited Pornography and Marketing). It is enforced by the US Federal Trade Commission (FTC), and the Department of Justice has authority to enforce its criminal sanctions. Other federal and state agencies can enforce the law in their jurisdictions, and Internet service providers can also sue violators.

What Spam Can Do

Spammers use forged messages, stolen identities, bogus cancellation addresses, and relay hijacking to hide their identities when sending their advertisements or bogus messages. This activity uses a lot of Internet bandwidth.

We can categorize spam issues from both the client and server perspectives. From the client perspective, if users receive more than 40 spams per day and need an average of 10 seconds each to decide what to do with them, they'll be wasting approximately 60 hours a year – more than seven work days – dealing with spam. Moreover, if users have limited Internet access on their mobile phones, they might have to pay extra each month just for downloading spam messages. Users clearly waste both money and time downloading spam. From a server perspective, processing numerous messages thrust onto the server, especially at the same time, can lead to a system crash or prevent users from sending or receiving messages (DoS attack).

Spammers also search open mail relays, SMTP servers that let anonymous Internet users send mail to deceive people with fraudulent messages. They can deceive inexperienced Internet users by using spammers' bluffs – that is, by misrepresenting themselves and their business enterprises. They can also send messages to a list of mail addresses obtained using software that crawls through Internet webpages.

Although outside our article's scope, the general definition of spam that we gave earlier can also be applied to the use of malicious programs, malware, or worms called *bots* that are attached to messages sent to people to infect other computers on the Internet. These infected computers, or *zombies*, give spammers full access to the computers' resources, letting them control the computers and either launch DoS and phishing attacks[2] on websites or disseminate additional malware.

As an example, consider a scenario in which attackers wish to misuse the DNS round-robin technique, which is used to balance the load among servers. To do this, attackers use a "fast-flux" technique to change an IP address and infect the client's computer and then use that computer's resources for further attacks. When a host resolves a domain name, the DNS server replies with a large list of IP addresses corresponding to that domain. According to the round-robin technique, the client DNS chooses one of these addresses (the attackers' computer) to gain access to one of the bots. The bot then acts like a proxy to direct a host to a malicious website the attacker controls.[3]

Spam in IPv6 Networks

The IPv6 network supports 2^{128-32} times more unique IP addresses than IPv4. Generally, service providers allocate prefix ranges to each home or small business network. Each of these networks would therefore have direct control over at least 2^{64}

unique IP addresses within their assigned subnet. Spammers might search for insecure SMTP-enabled computers in these networks. They might then start sending spam messages using different IP addresses accessible in that subnet. The sheer size of the addressable IPv6 address space threatens to render useless many antispam technologies that are based on IPv4 addresses, such as IP blacklisting.

IPv6 vs. IPv4

In 1998, the IETF IPv6 Working Group offered IPv6 to solve IPv4's security issues and the lack of addresses (see http://tools.ietf.org/html/rfc2460). On 3 February 2011, the Internet Assigned Numbers Authority allocated the final blocks of addresses to the regional registries, exhausting the central address pool. In addition, network address translation (NAT) technologies used in IPv4 networks to counter the lack of addresses hinder transparency and make problems for end-to-end communications such as voice over IP (VoIP) and IP Security (IPSec).

IPv6's main advantages over IPv4 are

- *larger address space*: offers potential support for 2^{128} unique IP addresses for each and every device or node attached to the Internet;
- *streamlined protocol header*: improves packet-forwarding efficiency;
- *stateless autoconfiguration:* lets nodes determine their own addresses;
- *multicast:* increases efficient one-to-many communications;
- *jumbograms*: provides very large packet payloads for greater efficiency;
- *network layer security*: offers encryption and authentication of communications;
- *quality of service (QoS) capabilities*: helps prioritize the traffic flow using packet marking;
- *anycast*: provides redundant services using nonunique addresses; and
- *mobility*: better supports mobility through the Mobile IPv6 (MIPv6) protocol's advanced features.

Address Generation and Allocation

Because IPv6 supports 2^{128} unique IP addresses, large networks use mechanisms other than manual address allocation to each computer because system administrators have difficulty keeping track of all the network's computers. IPv6 has two different addressing mechanisms.

Stateful address configuration is handled by the Dynamic Host Configuration Protocol for IPv6 (see www.ietf.org/rfc/rfc3315.txt). DHCPv6 lets IPv6 nodes assign their addresses and receive configuration information from DHCP servers on the network. It's similar to DHCP in IPv4 networks, but it requires manual intervention to configure each node.

Stateless Address Auto-Configuration (SLAAC) and Neighbor Discovery refer to the Neighbor Discovery Protocol (NDP; see tools.ietf.org/html/rfc4861), which is an essential protocol in the IPv6 suite. NDP's main functions in IPv6 are similar to the Address Resolution Protocol (ARP) in IPv4. NDP also supports router discovery and address assignment, replacing some of the DHCPv4 functionality. NDP greatly improves efficiency and network manageability. It's also heavily used in several critical functions such as generating a node's own IP address automatically, discovering other existing nodes on the same link, determining other nodes' link layer addresses, detecting duplicate addresses, finding routers, and maintaining reachability information about paths to active neighbors.

Why IPv6 Is Important

In IPv4 networks, the largest daily amount of traffic after file sharing is traffic related to SMTP usage; unfortunately, a large percentage of that SMTP traffic is spam. According to the spamcop blacklist report, spammers generate approximately 7.5 messages per second each month (see www.spamcop.net/spamgraph.shtml?spammonth). According to Google statistics (www.google.com/ipv6/statistics.html), the use of IPv6 has increased dramatically since 2010, as has the use of SMTP, which is a main protocol running within IPv6 networks.

According to AMS-IX (a mainline router), the average daily traffic of IPv6 is 2.7 Gbytes, which is roughly 4 percent of the Internet's total traffic (IPv4 and IPv6; see http://www.worldipv6launch.org/measurements). This means that more ISPs are supporting IPv6 and thus there's more IPv6-related traffic.

The IPv6 Large Address Space

Based on the current blacklisting and graylisting algorithms, when a mail firewall receives a large amount of SMTP traffic from an individual IP address, it adds that IP address to the blacklist or graylist database. In the IPv6 environment, for privacy or security reasons, node addresses are temporary. If spammers gain full access to the resources of one IPv6 node in a network, they can send spam through a different authorized IP address from the same node. Moreover, in a subnet, there are 2^{64} IPv6 usable addresses, making it easy for spammers to change their IPv6 addresses every second or to send out each spam message with a different address. This proves that the current blacklisting and graylisting algorithms will no longer work because traffic will be from different IP addresses and the mail firewall will be unable to detect a specific node's malicious behavior. This means the attacked network's system administrator must find the infected node and disinfect it, which is a daunting task. The reputation of that ISP or organization could very well be damaged by the spamming scenario described here.

Antispam Approaches and Technologies

When SMTP came into existence, no one thought about it needing a security mechanism. Thus, SMTP is a simple, text-based protocol that supports only a basic mechanism to avoid spam – that is, the receivers can check whether or not the sender's system meets the RFC standards. So, spammers and criminals have misused the SMTP system by flooding it with lots of spam.

According to some estimates, the cost of spam in terms of lost user productivity has reached more than US$1 billion annually. This is why governments have spent so much time and so many resources trying to resolve the spamming issue. Antispam approaches can be classified into three main categories: content-based, DNS-based, and collaborative spam filtering.

Content-Based Approaches

In content-based methods, the information contained in the mail body or header (such as the subject) is compared to different rules to automatically classify it as spam or legitimate. This approach uses different classifiers to label a message as spam, including rule- and Bayesian-based classifiers. This is the main reason that, in some classifications, Bayesian-based approaches are also listed under this category. Bayesian-based is the most popular technique because it's easy to implement. The algorithm's native version is the *naive Bayes* probabilistic, statistical classifier. In the learning stage, messages are classified as spam according to the probability of the frequency of a particular term used in them. Later, this data is used as a criterion to detect spam.[4]

Using content-based approaches that rely on existing classifiers has some drawbacks. In a rule-based classifier, working on the application layer leads to huge costs when applied to large-scale networks because the more rules the end user assigns, the more time it takes to analyze each message. Second, it takes a long time to establish the learning sample sets and define the rules. The Bayesian classifier calculates the next message's probability value by collecting as many samples as it can to increase its ability to filter out spam.[5] However, it's still vulnerable to poisoning attacks. For example, in the basic "good-word attack," the spammer chooses random words from a list of common English words and sends out emails with large amounts of legitimate text (or a combination of legitimate text and spam). In so doing, the spammer decreases the content-based approach's ability to correctly identify messages as spam.[6]

The cost of spam in terms of lost user productivity has reached more than US$1 billion annually.

DNS-Based Approaches

The emergence of DNS-based approaches dates back to 1997, when two software engineers began keeping a list of IP addresses or URLs that were known to send spam. The list became known as the *real-time blackhole list* (RBL). This was the first version of DNS blacklists (DNSBLs) that relied on static IP addresses. Today, it would be impossible for system administrators to manually create such a blacklist. Thus, real-time blacklist databases were created and are maintained on the Internet. When a mail server receives an incoming SMTP connection request, it typically checks a list of DNS blacklists and white lists (DNSxLs; see tools.ietf.org/html/rfc5782 and tools.ietf.org/html/rfc6471, respectively) using the existing DNS client and server protocols and utilities. If it finds one with a DNSBL entry, it will reject the connection. Today, in addition to mail servers using DNSxLs to avoid spam, antivirus, firewalls, and other security systems use them to prevent end users from accessing fake websites.

There are two different DNSxLs – IP address DNSxLs and domain DNSxLs. The former refers to the storage of an individual IP address or a range of IP addresses in lists. The latter refers to storage of individual domains (xx1.domain.com and xx2.domain.com) or to a root domain (as in domain.com).

Although DNSBL is a popular mechanism, IP addresses in DNSBLs must be updated continuously because spammers are continuously, dynamically changing their IP addresses. This updating process takes time, and spammers might complete their activities before the addresses are blacklisted. To deal with these dynamics and solve this problem, researchers have offered different behavioral blacklisting techniques. The basic idea is that spammers' applications – such as mass-mailing worms and malwares – also must rely on the DNS to resolve the mail servers' domain name into a valid IP address. Thus, they'll have to leave some traces in the DNS traffic.[7,8] For example, a spammer might use different "from:" addresses, but with the same source IP. Thus, a data mining approach would be able to recognize many mass-mailing hosts.

Collaborative Spam Filtering

Collaborative spam filtering seeks to involve users in labeling unwanted messages as spam, as well as conducting filtering in mail transfer agents (MTAs), SMTP clients, and servers that provide a mail transport service. Therefore, the antispam applications are fed by user input and employ the same technique as spammers to combat spam effectively. So, when a user classifies an email as spam, a signature is computed on the email and added to the collective knowledge base database. A signature is computed on every new email received and compared to the database of known spam. If the signature matches one in the database, it's considered spam. The main issue in this approach is that not all users consider the same email as spam. Also, most users don't label enough messages for an individual local classifier to be effective.

Several studies have been done in this area to try to improve this approach. One study used a peer-to-peer network to scale up this algorithm. In this method, the algorithm benefits from local network resources and establishes a distributed database in which to save information about spam messages and share it with other peers. A modified version of that algorithm sends information about filtering rules to other peers instead of sharing the information about each spam message.[9] So, when a new email is received, the system will activate the appropriate filters from the email users and then compute the message's final spam probability. To resolve the privacy and scalability issues, it can group users' filtering by their similar interests. Another possibility is to use social network approaches in which users can choose their "friends." In such cases, the collaborative exchange would occur using only filters from the social network.

Other Approaches

Although the stated antispam techniques are promising, they don't address the equally important security issues. Thus, cryptographic approaches such as Domain Keys Identified Mail (DKIM; www.ietf.org/rfc/rfc4870.txt), the Sender Policy Framework (SPF; www.ietf.org/rfc/rfc4408.txt), and Author Domain Signing Practices (ADSP; http://tools.ietf.org/html/rfc5617) are offered to permit authorization and verification of the senders or the senders' domain.

Internet email can be forged in several ways. To deal with this, the SPF email validation system lets domain owners' authorized hosts use their domain names in the address "from" or "HELO" identity (the SMTP command to start a new connection with a mail server). One advantage of this is that local policy decisions about the mail can be made based on the sender's domain rather than the host's IP address because domain name reputations are likely to be more accurate than those of the host IP addresses.

However, this technique is vulnerable to DoS attacks, and SPF-authorized email might contain other false identities. It's entirely possible for malicious senders to inject a message using their own domain into the SPF identities, to have that domain's SPF record authorized by the sending host, and yet still have the message easily list other identities in its header. This could occur unless the user or the mail user agent (MUA) notices that the authorized identity doesn't match that of the other, more commonly presented identities.

Another drawback to this technique is that network administrators must create SPF records for their domains so that others, not they themselves, will benefit from this configuration. However, the latter problem is resolved with the Dynamic Sender Policy Framework (DSPF).[10] The third-party thus collects legitimate servers' IP addresses that send emails automatically. Clients also don't need a DNS configuration to check the SPF records.

Another approach is DKIM, which describes a domain-level authentication framework for email using public-key cryptography and key server technology. It permits verification of a message's source and content using either MTAs or MUAs. It signs a domain to claim responsibility for a message, but, unlike the signature verifier, it doesn't depend on public/private key pairs being issued by trusted authorities. DKIM requests the public key from a repository in the claimed signer's domain directly rather than from a third party, and it doesn't need any new email infrastructure because it's compatible with the existing ones.

However, DKIM depends on DNS administration and the security of the DNSs. To reduce its dependency on DNS, it uses Trusted Core,[11] which is similar to Trusted Third Party. Trusted Core has a scheme embedded in the SMTP. The ISPs must register their authorized users in deposit agents or trusted cores one time. This works by first having the SMTP client and server establish a one-time shared key. The client stores this key and the receiver's addresses in one trusted core, and then supplies the server with that trusted core's URL. Finally, the server retrieves that URL – along with the sender and receiver addresses – from the SMTP command. These are then sent to the trusted core. If the trusted core can correctly verify these matching addresses, it returns the shared key to the server.

Another approach used to prove sender authenticity is ADSP, which uses DKIM to relay mail for the sender domain (author domain). However, if a sender message (address "from") doesn't have a DKIM signature, ADSP decides how the receiver should query this message and asks whether there should be a signature or whether the message should be discarded.

Anti-Spam Approaches and IPv6

The fight against spam will never end: as detection techniques continue to improve and new classifiers are developed, spammers will continue to come up with new techniques to circumvent them. In IPv6 networks, spammers have one extra powerful weapon at their disposal: the large address space available in each IPv6 subnet. This lets them change their IP addresses regularly for a certain number of messages. They can also combine attack techniques such as Bayesian poisoning and DNS DoS to bypass spam filtering and make the role of antispam systems less significant. However, some email providers combine the algorithms described above to detect spam. Google also uses an optical character recognition (OCR) algorithm to read and detect spam embedded in images and thus combat good-word attacks.[12]

The question now is, can current antispam systems tolerate the overhead needed to process a large amount of unexpected spam traffic? Can they identify spammers before the spammers can change their IP addresses and hide their identities again?

In IPv6 networks, spammers have one extra powerful weapon at their disposal: the large address space available.

We can classify antispam approaches into three main categories according to their feasibility in IPv6: *applicable*, *applicable with modification*, and *not applicable*. Because there's no proof that any of the approaches here won't work in IPv6, we'll focus on the first two classifications.

Applicable

All approaches in this category – which includes content-based and collaborative approaches – will continue working in IPv6 as they do now in IPv4 networks.

For example, content-based and behavioral-based approaches are tightly coupled to their classifiers. Although Bayesian filtering is the most popular technique here, as we discussed earlier, its statistical approach is vulnerable to good-word attacks. One solution is to frequently retrain the classifier, but the learning time increases with the number of featured words. Another solution is to combine DKIM with other classifiers to combat the attack.[13] In the preprocessing stage, the message is checked using one of the famous classifiers. Then, if the message isn't detected as spam in that stage, the DKIM signature header is constructed such that it's covered by the signature. The domain and identity are signed within the header using a private key. Later, the receiver will decrypt the hash value and verify the signature by the public key obtained through a DNS query.

Applicable with Modification

This category's approaches – which include DNS-based and non-DNS-based blacklist and graylist filtering – require modifications to be applicable in IPv6 due to its large address space, which hinders blacklisting based on individual IP addresses or domains.

We can divide blacklisting, whitelisting, and graylisting into two main categories:

- *Local network*. As in IPv4 networks, system administrators can add their router prefix to local blacklist databases regardless of which addressing mechanism they're using, be it NDP or DHCPv6. In IPv6, the prefix is the left-most 64-bit IPv6 address allocated to a legitimate router to identify its subdomain and thus enable system administrators to set their IPv6 addresses. They can thus add just the mailer's IP address to the local whitelist, which will prevent spammers from sending spam by accessing a network node.
- *Global network*. As in local networks, the solution here is router prefix blacklisting. Router prefixes don't change over time, but some countries, such as Germany, require ISPs to change their prefixes regularly due to privacy issues.

An example of an attack that can be foiled by these solutions is when spammers try to change the server's IP address to misuse the DNS round-robin technique. To do this, spammers use the fast-flux DNS attack technique: when attackers start their SMTP communication with a mail server, the mail server resolves its domain names and the DNS server replies with a large list of IP addresses from a different range corresponding to that domain. According to the round-robin technique, the mail server chooses one of these addresses and sends a query to DNSxLs to find any entries existing there. Spammers thus have a better chance to bypass this filter using this technique.

One solution would be to check DNS behavior when a mail server uses a Fully Qualified Name (FQDN; www.ietf.org/rfc/rfc4703.txt) to check for absolute domain names. (For example, if the mail server has the name "mailsrv" and the parent domain name is "mydomain.com," then the FQDN is "mailsrv.mydomain.com.) Spammers won't take the time to add a new DNS record for each subdomain, such as "xx.mydomain.com," and will instead misuse the wildcard DNS: if no match is found for a subdomain name, the DNS server replies only to queries within an authorized zone. So, by sending a query with a random string in the subdomain, if the replies are a similar address for all domains, then this is a wildcard DNS; otherwise, it's a normal DNS. It's possible to recognize spam by checking the time-to-live (TTL) value of the domain names because the attacker uses a smaller TTL to prevent clients from caching older IP addresses. However, this approach isn't very practical because it's not fast enough. During the query, these domain names can be taken by authorized groups such as countries, ISPs, and so on.[2]

However, prefix blacklisting can't prevent attacks against DNSxLs on the Internet where attackers can instigate DoS attacks against DNSxLs and thus block authorized systems from receiving any results to their query requests. Like other DNS-based approaches, DNSxLs is prone to various attacks such as DoS, cache poisoning, and so on.[14]

Another antispam technology, known as *Spamhaus*, focuses on IPv6 blocklists and addresses the problem wherein existing DNS caches can be filled to capacity because a large volume of spam can be generated due to IPv6's larger addressing scheme. Spamhaus uses a B-tree query system and the returned query result is like its existing IPv4 blocklist. This technology is still being tested; whether it will work well under high traffic loads is still questionable.[15]

Although IPv4 has inspired some spam-filtering approaches, the fight against spammers will be ever ongoing. Spammers can bypass the most popular approaches in IPv4 using various attacks, such as Bayesian poisoning, DNS DoS, and cache poisoning.

It's clear that in IPv6, network spammers have one extra, powerful weapon at their disposal – the large address space – so DNS-based, blacklisting, and graylisting solutions can't continue working as before without prefix blacklisting. Despite the fact that the problems we describe here are future Internet problems, we should change the blacklisting and whitelisting algorithm to support this solution before the IPv4 to IPv6 migration process is complete. The IPv4 to IPv6 migration can take a long time. The need to implement the deterrents against spam in IPv6 is immediate as the use of spam for ill gain continues to grow at an astronomical rate.

References

1. "Minutes of the Senate Committee on Judiciary," Nevada State Legislature, 14 Mar. 2003; www.leg.state.nv.us/Session/72nd2003/Minutes/Senate/JUD/Final/2197.html.
2. S. Suwa et al., "DNS Resource Record Analysis of URLs in Email Messages for Improving Spam Filtering," *Proc. IEEE/IPSJ Int'l Symp. Applications and the Internet*, 2011, IEEE CS Press, pp. 439–444.
3. F. Paget, "From Fast-Flux to RockPhish," McAfee Blog Central, 30 Nov. 2007; http://blogs.mcafee.com/mcafee-labs/from-fast-flux-to-rockphish-part-1.
4. T.A. Almeida and A. Yamakami, "Content-Based Spam Filtering," *Proc. 23rd IEEE Int'l Joint Conf. Neural Networks*, IEEE Press, 2010, pp. 1–7.
5. J. Wu and T. Deng, "Research in Anti-Spam Method Based on Bayesian Filtering," *Proc. Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, IEEE CS Press, 2008, pp. 887–891.
6. G.L. Wittel and S.F. Wu, "On Attacking Statistical Spam Filters," *Proc. 1st Conf. Email and Anti-Spam* (CEAS), 2004; http://ceas.cc/2004/170.pdf.
7. J. Zhang, Z.H. Du, and W. Liu, "A Behavior-Based Detection Approach to Mass-Mailing Host," *Proc. Int'l Conf. Machine Learning*, IEEE Press, 2007, pp. 2140–2144.
8. A. Ramachandran, N. Feamster, and S. Vempala, "Filtering Spam with Behavioral Blacklisting," *Proc. 14th ACM Conf. Computer and Communications Security*, ACM, 2007, pp. 342–351.
9. P. Sousa et al., "A Collaborative Approach for Spam Detection," *Proc. 2nd Int'l Conf. Evolving Internet*, IEEE Press, 2010, pp. 92–97.
10. N.T. Anh, T.Q. Anh, and N.X. Thang, "Spam Filter Based on Dynamic Sender Policy Framework," *Proc. IEEE Int'l Conf. Knowledge and Systems Eng.*, 2010; http://doi.ieeecomputersociety.org/10.1109/KSE.2010.11.
11. M. Takesue, "E-mail Sender Identification through Trusted Local Deposit-Agents," *Proc. 14th Int'l Conf. Network-Based Information Systems*, IEEE Press, 2011, pp. 84–91.
12. "So Much Time, So Little Spam," *About Gmail*, Google, https://mail.google.com/mail/help/intl/en/fightspam/spamexplained.html.
13. K.K. Kowser, K. Saruladha, and M. Packiavathy, "A DKIM-Based Architecture for Combating Good Word Attack in Statistical Spam Filters," *Int'l J. Scientific & Eng. Research*, vol. 2, no. 6, 2011, pp. 120–122.
14. S. Aryapperuma and C.J. Mitchell, "Security Vulnerabilities in DNS and DNSSEC," *Proc. 2nd Int'l Conf. Availability, Reliability, and Security*, IEEE CS Press, 2007, pp. 335–342.
15. *Spamhaus IPv6 Blocklists Strategy Statement*, the Spamhaus Project, 2012; www.spamhaus.org/organization/statement/012/spamhaus-ipv6-blocklists-strategy-statement.

Hosnieh Rafiee is a PhD student at Hasso-Plattner-Institut at the University of Potsdam. Her research interests are in network security – including spam-filtering approaches and DNS security, focused on composing cryptographic mechanisms in IPv6 networks – and deployment of Secure Neighbor Discovery (SEND). Rafiee has a master's degree in IT-computer networks engineering from Amirkabir University of Technology, Tehran. Contact her at hosnieh.rafiee@hpi.uni-potsdam.de.

Martin von Löwis is a lecturer at the University of Potsdam's Hasso-Plattner-Institut in the Operating Systems and Middleware Group. His research interests include compiler construction and embedded systems, as well as IPv6 and public-key infrastructure. Löwis has a PhD in computer science from the Humboldt-University in Berlin. Contact him at martin.vonloewis@hpi.uni-potsdam.de.

Christoph Meinel is a professor and director at the Hasso-Plattner-Institut at the University of Potsdam, where he leads the Internet Technologies and Systems research group. His research interests include security and trust engineering, Web 3.0, and eLearning. Meinel has a PhD in computer science from the Humboldt-University in Berlin. Contact him at christoph.meinel@hpi.uni-potsdam.de.