



The Case for S/MIME Digital Signatures as Solution for Email Sender Authentication

In addition to educational or policy-based solutions to the phishing problem, multiple technical solutions have been proposed. These solutions attempt to enable legitimate enterprises to send email that can be easily authenticated by their customers and partners. By deploying such technology, users who might otherwise be susceptible to phishing attacks would be able to protect themselves. After reviewing the popular technical proposals put forth to achieve this goal, S/MIME digital signatures stand out as a logical choice for authenticating email on the Internet to deter phishing attacks.

Overview

The fundamental problem that end-user victims of phishing attacks have is that they cannot trust an email's origin. The only visible elements of an email available in most email clients to make this trust decision are the address in the FROM field of the header and the message content itself. The content of a phishing email is typically very convincing in its alleged authenticity and provides no utility to discern fact from fraud.

Since phishers can spoof the FROM address without discovery, end-users are left to make a judgment call as to whether they should follow the instructions in the email and disclose personal information at the web site the embedded link takes them to. The growing numbers of successful phishing attacks suggests that a method for validating the FROM address in an email is needed. Once deployed, this validation method will provide Internet users with sufficient information to make educated decisions about the validity of a potential phishing email.

Of the several solutions to the email sender authentication problem discussed in the messaging industry today, only one meets the requirements of providing both immediate and lasting value to address phishing – S/MIME digital signatures. Because this standard provides the highest level of sender authentication, it can also be applied to the more general problem of spam. In addition, S/MIME's ability to provide message integrity, confidentiality through encryption, and non-repudiation services makes it the clear choice as the solution to deter today's phishing attacks and provide the infrastructure to protect tomorrow's business uses of email.

Why Education Alone Won't Stop Phishing Attacks

Phishing is a crime that uses social engineering techniques to fool users into taking ill-advised actions. Essentially, the phisher is counting on the fear, guilt, or general willingness of unsuspecting victims to trust the email's contents and follow its instructions. A standard reaction to combat this type of crime is to educate victims on how to identify the crime and avoid it. Virtually every major organization that has been a target for phishing attacks has taken the route of educating their users on how to avoid being phished. Analyzing this educational content produces some interesting contradictions and confusion for the user.



- ***“Don’t click on links in email”, combined with “Open your browser and type the URL you see in the email manually or go to the home site and navigate to the correct page”***
 - This advice may protect the user from being phished, but it also prevents a lot of business from being done on the Internet. Manually typing in long URLs makes a company’s website unusable and uninviting. The ability to notify customers of goods and services that are available at one-click targeted URLs is the “bread-and-butter” of Internet commerce and marketing. If all Internet users are trained to stop clicking on links in emails, Internet commerce will slow down considerably.
- ***“We’ll never ask for a password or personal information in an email”***
 - The promise of driving more business processes to the Internet requires that visitors to a website are adequately authenticated. Providing a customer with a link to a protected section of a website is the crux of many financial services Internet applications today (e.g. online banking, statement presentment, etc). Business want to be asking for passwords of the clients that come to their website so they can provide more personalized information and services as well as drive more sales.
- ***“Make sure the email address in the FROM field contains example.com”***
 - This advice is actually extremely important (as will be demonstrated later), but irrelevant today due to the security holes in the SMTP standard as it is implemented on the Internet. No one can trust that an email stating it came from *example.com* in fact came from *example.com*.

Industry’s current anti-phishing education is a desperate attempt to “stop the bleeding” by giving users overly broad advice. This advice may irrevocably damage many of the productivity gains that email and browsers have provided for commercial business over the past decade. There must be a way to let people use email and browsers as they always have, but also make informed decisions about the trustworthiness of an email’s origin. A technical analysis of the problem provides more insight into where a primary solution needs to be targeted.

Why Phishing Attacks Work – SMTP and Its Limitations

SMTP email is typically originated and read using MUAs (Mail User Agents), also known as email clients. Once authored, email is sent to recipients through servers on the Internet using a series of store and forward agents known as MTAs (Mail Transfer Agents). There may be only two MTAs involved in an email transaction (sender’s and recipient’s), or there may be multiple intermediate MTAs. In order to accurately distinguish the author of a message (to which replies should be sent) and sender of a message (to which bounces should be sent), SMTP implementations on the Internet support two type of “from” addresses.



The first is the FROM address. This address is typically the actual originator of the email and is also typically the same as the REPLY-TO address that recipients use when replying. The FROM address is displayed in most every MUA in deployment today as it purportedly tells the recipient who the message is from. The SMTP protocol allows any value to be inserted as the FROM address, which is the primary weakness phishers take advantage of.

The second type of sender address in SMTP is the envelope sender, also referred to as the MAIL FROM address. This is typically the address to which undeliverable mail responses (bounces) should be sent. Again, SMTP allows any value to be inserted as the MAIL FROM address. Phishers further hide their true identity by spoofing this address as well.

Described below is a detailed methodology for a sample phishing attack. It demonstrates how the combination of SMTP security holes and social engineering can provide an effective attack. Assume the following actors and components of the attack:

- The phisher (P)
 - The recipient (R)
 - The company whose domain is being spoofed by the phisher (C)
 - The phisher's email server (Ps)
 - The recipient's email server (Rs)
 - The recipient's email client (Rc)
1. P generates a fraudulent email with content that looks just like a legitimate email from C to all its customers. The colors, graphics, text treatment, and composition are identical to what C uses to normally contact its customers. There is no way for C to technically prevent P from creating this type of content. The message in this email is particularly insidious as it describes how *"A recent set of phishing attacks have corrupted our customer account database at C. Please help us reinstate your account at a secure website provided by the C Security Service by clicking on the link below"*. P then inserts the email address *customerservice@C.com* in the FROM field of the email.
 2. P sends this email to as many email addresses as he can get a hold of using server Ps. He may have done previous spam attacks to understand which email addresses are likely C's customers to reduce the number of phishes he must send to be effective. The domain of Ps happens to be *Csecurityservice.com*. The address P uses in the MAIL FROM address of the email is *customerservice@Csecurityservice.com*. This is so that any bounces from invalid recipient email addresses are sent to P's servers and not C's. This will prevent C from being Joe Jobbed and noticing a large number of bounces coming from email it knows it did not send.
 3. R happens to be a customer of C and has heard about these phishing attacks on the Internet. R's email server (Rs) receives a connection from Ps, accepts it, and stores the message in R's email



- inbox. R uses his email client (Rc) to download the message. When he sees P's email, he inspects it closely. The address that Rc displays, *customerservice@C.com*, has *C.com* in it, and the "customerservice" string to the left of the "@" matches another email he got from C a few months back regarding a product return he made. R trusts this address as legitimately coming from C.
4. While R thinks he's savvy to phishing attacks, the content of P's email takes him off-guard. "*Why sure*", R thinks to himself, "*These phishing attacks probably are wreaking havoc on corporate account databases. I read about it in the papers. I better make sure my account is not ruined, because I use C's site often.*" R clicks on the link and is taken to a Web site.
 5. R is taken to <https://www.Csecurityservice.com/accountreinstatement>. R remembers that the email said the site to reinstate his account was provided by a "C Security Service", so the syntax of this URL makes perfect sense to him. The SSL connection his browser tells him has been made further convinces him that the site is legitimate. The form on the site asks for just the right kind of personal information that C would ask of its customers in order to reinstate an account. R fills out the form and hits the submit button. The phisher has now succeeded in stealing personal information from R.
 6. R realizes some time later that he's been phished. C may discover P's attack from R or other victims' reports. By the time C is able to get P's ISP to shut down the <https://www.Csecurityservice.com/accountreinstatement> URL, P has already gathered hundreds if not thousands of sets of passwords and other personal account information. The fact that P can never again use <https://www.Csecurityservice.com/accountreinstatement> as a Web site or even customerservice@Csecurityservice.com as the MAIL FROM address in another phishing attack (due to updated blacklists in anti-spam servers) is beside the point. He's made a killing with stolen identities he already has. P can easily move on to the next company's set of customers to launch a phishing attack. He might even try phishing C's customers again with an entirely different set of legitimate-looking domain names and email content.

Where were the key decision points for R to protect himself? The first decision point was when he got the email. He believed the language of the email message and he believed it was from a trusted business partner because it said *C.com* in the FROM field. The second decision point was when he got to the Web site after clicking the link in the email. The content of the message set an expectation that *Csecurityservice.com* might be legitimate. Also, because C is not consistent with the domain and sub-domain names it uses in its URLs across its portal, R had no reason to believe this site was not legitimate. Finally, because there was a lock icon in his browser that confirmed a valid SSL connection, he implicitly trusted that his information was being protected by *Csecurityservice.com*, which in turn was vouched for by *C.com*.



After R trusted the origin of the email based on the value in the FROM field, all other spoofing techniques used by the phisher were more believable. For this reason, a solution to authenticating the FROM address of an email is the most important place to try and solve the phishing problem. Solving the problem at this point of the attack implies that the email senders who actually own the purported domain name in the FROM address can prove their identities to any recipient who receives email from this domain. With this feature in place for SMTP, phishers will be less likely to convince recipients that their email with spoofed FROM addresses is legitimate. By enabling recipients to immediately determine the origin of email, decisions can be made early to drop phishing emails without even having to read them. This has profound implications for bandwidth savings by an inbound email firewall solution that could utilize this same feature.

Requirements for a Solution

Rather than waiting for a more secure version of SMTP to be designed and implemented across the Internet, an incremental approach to a solution must be taken. Regardless of the approach, some amount of additional work must be done on the part of the sender and recipient above and beyond what happens with today's typical SMTP email transaction. Five basic requirements can be outlined for a solution:

1. It must adequately authenticate the visible FROM address in the header.

Unlike the MAIL FROM address exposed in the initial SMTP connection, the FROM address is typically the only address the inbound email recipient ever sees. It is this address that all decisions about an email sender's authenticity are made.

2. It must limit end-user training.

Experience has shown that end-users are not likely to deploy new email software or proprietary plug-ins *en masse* to avoid a perceived threat that they assume can be avoided by intelligent analysis of email content. Agreeing on what such client software should do and who controls ownership of it will inhibit adoption for several years anyway. So either existing MUAs must be able to support a solution, or the MTAs that send and receive email on the Internet must be enhanced. The one area of end-user training that may be mandatory is how authentic FROM addresses are distinguished from unauthentic ones. A visual cue in the MUA or the message itself will be necessary to educate the recipient of an email's authenticity.

3. It must use standards-based technology.

There are arguably two types of Internet standards: *de facto* and *de jure*. The Internet Engineering Task Force (IETF) membership approves Requests for Comments (RFCs) after years of research and debate to make *de jure* standards. The marketplace occasionally demonstrates adoption to make a *de facto* standard in lieu of the IETF involvement. Having both is ideal, but *de facto* standards are needed if the rising tides of phishing attacks are going to be stemmed quickly.



4. A unilateral deployment of a solution must add value.

This requirement is included in order to avoid the classic “chicken and egg” question that plagues so many messaging technology implementations. If both the sender and recipient must have special technology to provide a valuable solution, then neither will deploy. Because the phishing problem is fundamentally a problem for the legitimate senders of email whose brands are being eroded, a solution must be unilaterally deployable and at least partially effective.

5. It must be cost-effective for senders, recipients and email service providers.

A solution that allows only the elite to participate will not have long-lasting effects in the fight against phishing and email spoofing in general. While corporate email systems may have the resources to spend their way out of the problem, the public email ISPs must consider the cost of solutions that need to service millions of users.

Other technical requirements related to email sender reputation are specifically not mentioned in this discussion. While obviously important for automated systems that must determine whether the content of the email is legitimate for spam-blocking purposes, the phishing solution is more narrowly defined and not necessarily in need of a reputation semantic. Because the content of a well-designed phishing email usually appears to be legitimate, recipients need only evaluate whether the sender is also legitimate in order to protect themselves from the attack.

Survey of Leading Technical Solutions

There are four leading solutions for email sender authentication that have been introduced. Each meets at least one of the requirements listed above.

- SPF (Sender Policy Framework)
- Microsoft® Caller-ID
- Yahoo!® Domain Keys
- S/MIME (Secure Multipurpose Internet Mail Extensions) Digital Signatures

SPF

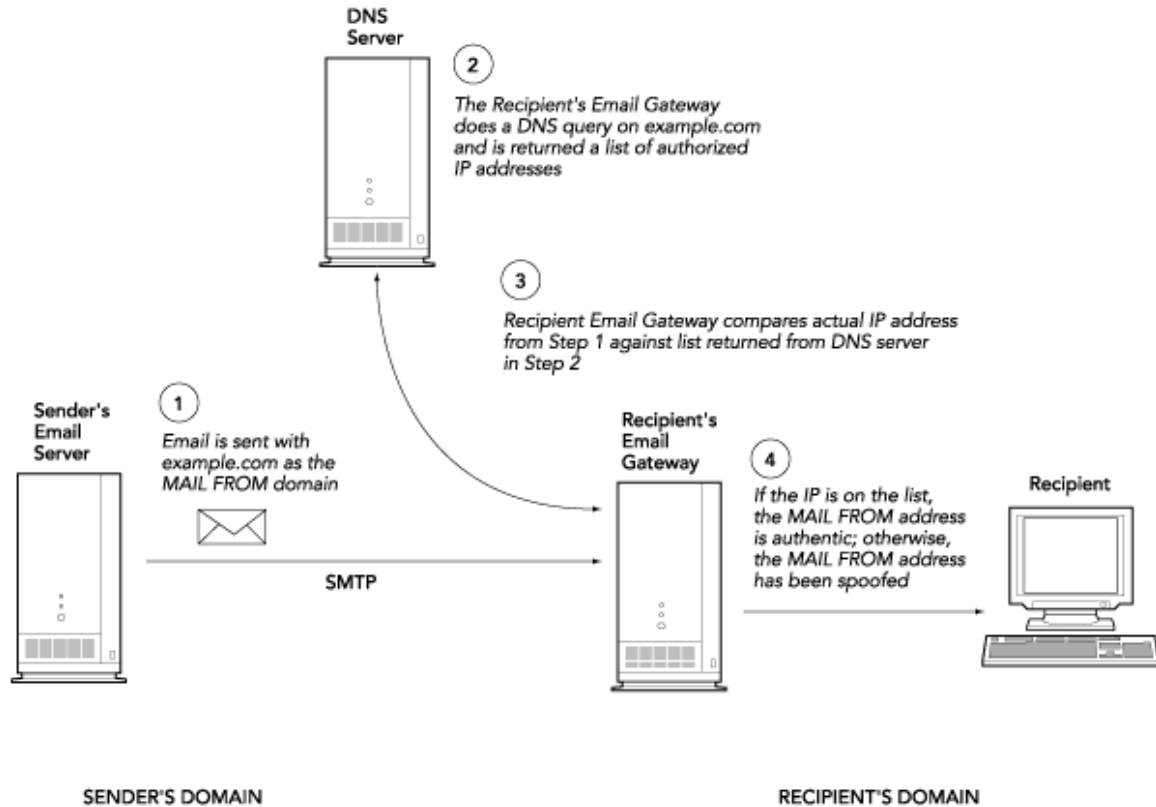
Note: *The current SPF specification is owned by Meng Weng at <http://spf.pobox.com/draft-mengwong-spf-00.txt>.*

SPF attempts to authenticate the MAIL FROM of an email by using a combination of the IP address in the TCP/IP connection from the upstream MTA and a list of approved IP addresses in the DNS record of the domain claimed in the MAIL FROM address.



A diagram and basic outline of how SPF would work to help deter phishing attacks is below:

SPF



1. A sending domain enters information in their DNS record listing the IP address of the outbound email servers that are authorized to send email on its behalf. All legitimate email from that sending domain uses a MAIL FROM address that matches that domain.
2. The recipient's SPF-enabled email gateway (the terminal MTA receiving email on behalf of the recipient) receives email from an email server participating in SPF. It performs a DNS query on the domain listed in the MAIL FROM address in the email. This query will return one or more IP addresses.
3. The recipient's email gateway then compares the returned IP address(es) in step 2 with the actual IP address in the TCP/IP connection made with the sending MTA in step 1. This MTA may in fact be the sender's email gateway, but it doesn't have to be. A third party mailing service may be allowed to send email on behalf of the domain in the MAIL FROM address.
4. If the actual IP address is in the list of approved IP addresses returned by the DNS query for that domain, then the MAIL FROM address is authentic. If no match is found, then the MAIL FROM is spoofed. Local policy determines how the email is handled at this point. Typically, a spoofed email will be dropped.



The security behind SPF lies in the fact that both the IP address in a TCP/IP connection and a DNS record are extremely difficult to spoof.

Advantages of SPF

- Owners of mail domains can independently register their own authorized IP addresses in DNS free of charge.
- Recipient MTAs can perform queries of DNS records for SPF entries free of charge
- There are currently over 8,000 Internet mail domains that have SPF entries in their DNS records.
- Phishers that attempt to use spoofed MAIL FROM addresses will likely have their email dropped immediately at the MTA performing the SPF check.

Disadvantages of SPF

1. It must be supported on both sides of the email connection to work.
2. There are hundreds of thousands of Internet mail domains that don't have SPF entries in the DNS records.
3. It is designed to be used by MTAs only, so MUAs that don't have SPF-enabled MTAs processing their email are unable to use the authentication information.

Does SPF Satisfy the Requirements For a Solution?

1. **It must adequately authenticate the visible FROM address in the header.**
No. It only authenticates the MAIL FROM domain. A small number of phishing attacks use the same domain in the MAIL FROM as in the FROM address.
2. **It must limit end-user training.**
Partially. It is designed as a protocol that a receiving MTA would use, thereby protecting MUAs from having to support it. However, this may be a disadvantage until a sufficiently high number of Internet MTAs support SPF. Also, without any visual cue in the MUA, end-users will not be able to know whether the SPF check was successful, or if the domain in the MAIL FROM address simply doesn't publish SPF information. Getting agreement among the MUA vendors and web mail providers as to what the standard visual cue should be will be a long and difficult process.
3. **It must use standards-based technology.**
Partially. The DNS TXT extension field is a standard place to insert ancillary information about a domain (like SPF entries). The process of performing a SPF query and acting on the results is not standards-based. Only in February of 2004 has an RFC for SPF been submitted to the IETF for consideration.
4. **A unilateral deployment of a solution must add value.**
No. Both hosts in the last hop of the email connection must support SPF for it to work. Both sender and recipient MTAs must support Caller-ID for it to work.



5. It must be cost-effective for senders, recipients and email service providers.

Yes. Registering SPF/Caller-ID claims in a domain's DNS record is free. Performing the query and possibly caching the results has fairly low bandwidth and CPU cycle requirements.

How could phishers work around email infrastructure that uses SPF?

- A phisher could use an email server that does not publish SPF entries in its DNS record. A receiving MTA that attempts to do an SPF validation will receive no reply. This MTA cannot arbitrarily drop the email, so it will be forced to accept it and hope that a downstream content filter will catch the phish. Until it is deemed unacceptable to NOT publish SPF/Caller-ID entries in a mail domain's DNS record, phishers will be able to ignore this protocol.
- A phisher could simply use valid MAIL FROM addresses with SPF entries in their DNS records. Phishers could choose to use hijacked MTAs to do this and they could cycle through valid email domains, both of which happen today with regularity. Only after a valid SPF MAIL FROM domain has been identified as facilitating a phishing attack can it be blacklisted, at which point the attack will have been completed.

Microsoft Caller-ID

Note: The current Caller-ID specification is published at http://www.microsoft.com/mscorp/twc/privacy/spam_callerid.msp

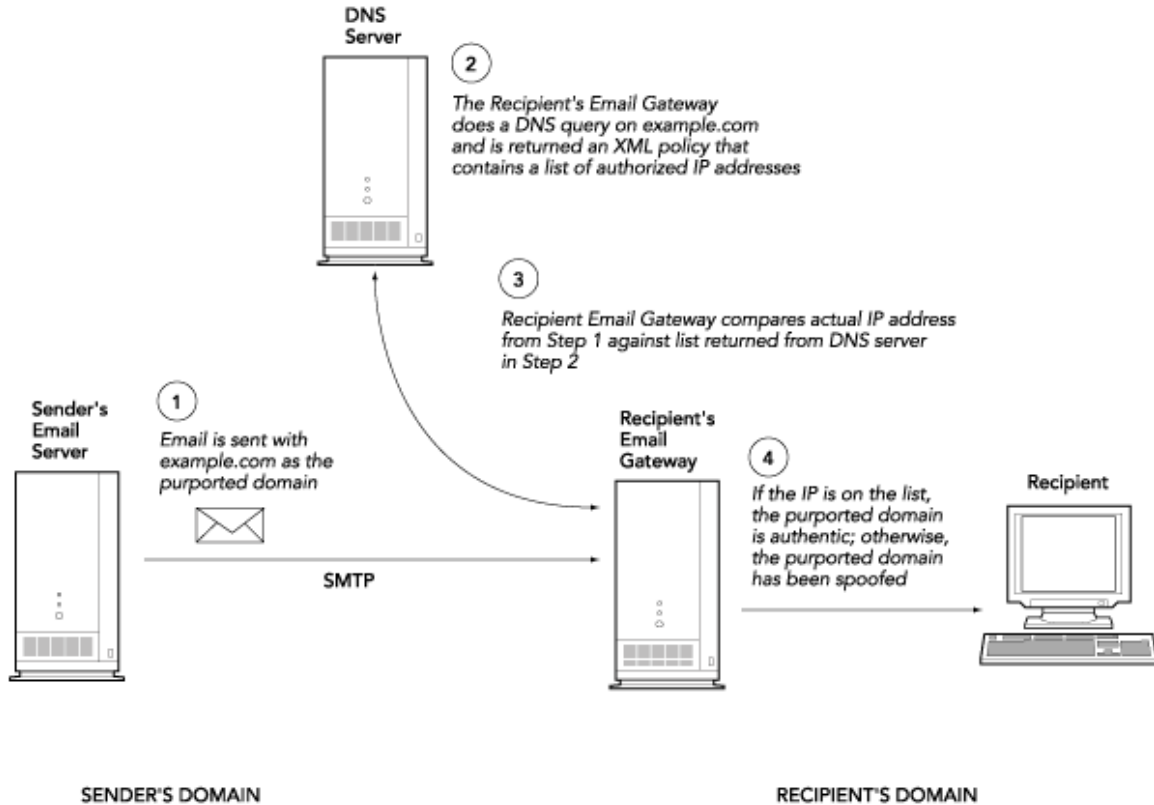
Caller-ID attempts to authenticate the email by using a combination of the IP address in the TCP/IP header and a list of approved IP addresses in the DNS record of the domain claimed in the "purported responsible address" of the email header. This address can be one of the following RFC2822 fully-qualified addresses:

- RESENT-SENDER
- RESENT-FROM
- SENDER
- FROM



A diagram and basic outline of how Caller-ID would work to help deter phishing attacks is below:

Caller-ID



1. A sending domain enters information in their DNS record listing the IP address of the outbound email gateways that are authorized to send email on its behalf. This entry is in the form of an XML document that could include additional policy information about the sender's email infrastructure. All legitimate email from that sending domain inserts a purported responsible address using only that domain.
2. The recipient's Caller-ID-enabled email gateway (the terminal MTA receiving mail on behalf of the recipient) receives email from an email server participating in Caller-ID. It performs a DNS query on the domain associated with the purported responsible address in the email. This query will return an XML policy document that will contain one or more IP addresses.
3. The recipient's email gateway then compares the returned IP address(es) in step 2 with the actual IP address in the TCP/IP connection made with the sending MTA in step 1. This MTA may in fact be the sender's email gateway, but it doesn't have to be. A third party mailing service may be allowed to send email on behalf of the domain in the purported responsible address.
4. If the actual IP address is in the list of approved IP addresses returned by the DNS query for that domain, then the purported responsible address is authentic. If no match is found, then the



purported responsible address is spoofed. Local policy determines how the email is handled at this point. Typically, a spoofed email will be dropped.

The security behind Caller-ID lies in the fact that both the IP address in a TCP/IP connection and a DNS record are extremely difficult to spoof.

Advantages of Caller ID

- Owners of mail domains can independently register their own authorized IP addresses in DNS free of charge.
- By authenticating the purported responsible address, the protocol provides a more granular approach to sender authentication than other proposals.
- The protocol's use of XML as the definition language makes it more easily extensible
- Recipient MTAs can perform queries of DNS records for Caller-ID entries free of charge
- Phishers that attempt to use spoofed purported responsible addresses will likely have their email dropped immediately at the MTA performing the Caller-ID check.

Disadvantages of Caller-ID

- Caller-ID must be supported on both sides of the email connection to work.
- It is currently designed to be used primarily by MTAs, so MUAs that don't have SPF-enabled MTAs processing their email are unable to use the authentication information. Microsoft may choose to enable their MUAs (Outlook and Outlook Express) with Caller-ID in the future.

Does Caller-ID Satisfy the Requirements For a Solution?

1. It must adequately authenticate the visible FROM address in the header.

Partially. It authenticates the purported responsible domain of the email. This may or may not be the FROM domain, depending on how the email header is constructed.

2. It must limit end-user training.

Partially. It is designed as a protocol that a receiving MTA would use, thereby protecting MUAs from having to immediately support it. However, this may be a disadvantage until a sufficiently high number of Internet MTAs support Caller-ID. Also, without any visual cue in the MUA, end-users will not be able to know whether the Caller-ID check was successful, or if the domain in the purported responsible address simply doesn't publish Caller-ID information. Getting agreement among the MUA vendors and web mail providers as to what the standard visual cue should be will be a long and difficult process.

3. It must use standards-based technology.

Partially. The DNS TXT extension field is a standard place to insert ancillary information about a domain (like Caller-ID entries). The process of performing a Caller-ID query and acting on the



results is not standards-based. Microsoft appears to have no current plans to submit Caller-ID to the IETF for approval.

4. A unilateral deployment of a solution must add value.

No. Both sender and recipient domains must support Caller-ID for it to work and the upstream MTA's IP address in the email flow must be listed in the sender's DNS entries.

5. It must be cost-effective for senders, recipients and email service providers.

Yes. Registering Caller-ID claims in a domain's DNS record is free. Performing the query and possibly caching the results has fairly low bandwidth and CPU cycle requirements.

How could phishers work around email infrastructure that uses Caller-ID?

- A phisher could use an email server that does not publish Caller-ID entries in its DNS record. A receiving MTA that attempts to do a Caller-ID validation will receive no reply. This MTA cannot arbitrarily drop the email, so it will be forced to accept it and hope that a downstream content filter will catch the phish. Until it is deemed unacceptable to NOT publish Caller-ID entries in a mail domain's DNS record, phishers will be able to ignore this protocol.
- A phisher could mix valid and invalid Caller-ID addresses in the header. Since the receiving MTA is looking for the first purported responsible address, the phisher could make the RESENT-SENDER, RESENT-FROM, or SENDER addresses map to real domains with valid Caller-ID entries, but the FROM address could be spoofed. Since most MUAs only present the FROM address to users, phishers could bypass a Caller-ID check at the receiving MTA (or even MUA if that capability were implemented). Microsoft acknowledges that loophole and suggests that MUA vendors display all the email header addresses to the user. This way, any inconsistencies in the purported responsible addresses could be identified by the recipient and used as a reason not to trust the email. However, waiting for MUAs to change their implementations doesn't meet the requirements listed above and would likely take an unacceptably long time.

DomainKeys

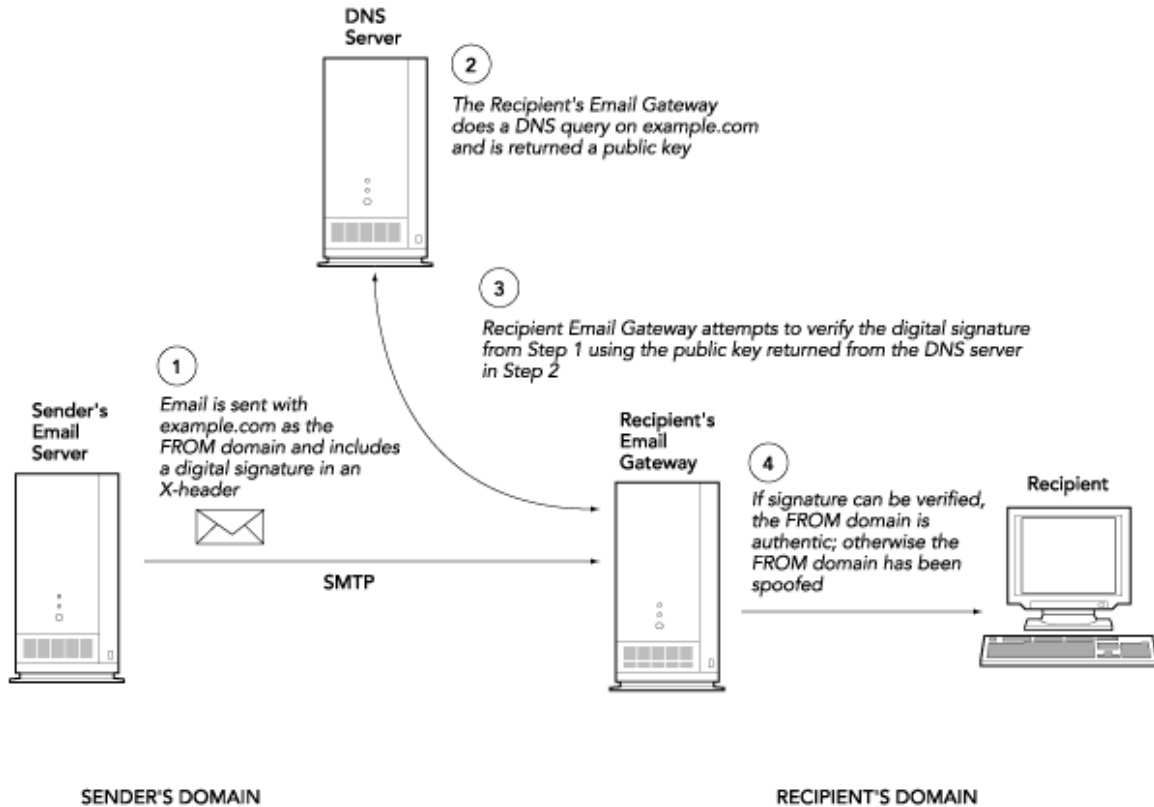
Note: Yahoo! first proposed an implementation overview for DomainKeys in Nov 2003. The exact specification is not yet publicly available at the time of this writing.

DomainKeys uses the power of asymmetric cryptography to provide evidence that an email in fact came from the domain in the FROM field of the email header. Note the fully qualified email address is not authenticated in this solution, only the domain. For the purposes of preventing email spoofing, domain authentication is likely a powerful first step. It may be a reasonable assumption that the owner of an email domain can police its own users and control access to its sending mail servers to avoid hijacking by phishers.



A diagram and basic outline of how DomainKeys would work to help deter phishing attacks is below:

DomainKeys



1. A sending domain generates a unique public/private key pair. The email administrator for that domain then takes the public key and enters it into a TXT record in the DNS entry for that domain. All future emails with that domain in the FROM address will then be digitally signed using the private key generated in step 1. This digital signature will be comprised of a digest of the entire email contents (including header values), which is encrypted using the private key. Text entries called *selectors* that describe the specific domain or sub-domain key used for signing the message will be inserted into the email header along with additional optional entries that describe the cryptographic algorithms used for signing
2. The recipient's email gateway (the terminal MTA receiving mail on behalf of the recipient) extracts the digest and other descriptive value from the appropriate X-headers in the email. The receiving MTA looks up the public key from a DNS query of the FROM domain when the email is received
3. The recipient MTA attempts to verify the digital signature in the header. The retrieved public key is used to decrypt the encrypted digest pulled from the email's X-header. The receiving entity then compares the resulting clear text with its own digest of the email contents it creates.
4. If the two values from step 3 match, then the digital signature is valid and the email was in fact signed by the domain that provided the public key from the DNS query. If the values don't match,



then the FROM address domain has been spoofed. The receiving entity then takes local action to process the message based on its authenticity. Typically, a spoofed email will be dropped.

The security behind DomainKeys lies in its use of asymmetric cryptography. When the retrieved public key successfully decrypts the message digest, it proves that the message was signed with the private key (ostensibly) available only to that domain at their outbound email servers. To ensure the proper public key is used to decrypt a particular signed message, the security of the DNS records is assumed. The likelihood that a phisher would be able to redirect a valid DNS public key query to a spoofed DNS entry is sufficiently small to make this solution secure.

Advantages of DomainKeys

- Assuming sufficiently random key cryptographic pairs can be generated for every email domain on the Internet, authentication of each domain can be verified.
- Any sending domain that wishes to generate a public/private key pair can do so with minimal cost. Yahoo! intends to make public the utilities needed to generate key pairs at no cost.
- The use of TXT extensions in DNS records provides a universally available “directory” service for recipients to retrieve the needed public keys.
- The extra processing power required on the recipient’s email infrastructure to verify a sender is somewhat mitigated by the fact that the relevant information is in the email header, and not in the MIME contents of the email itself.

Disadvantages of DomainKeys

- DomainKeys must be supported on both sides of the email connection to work.
- Because SMTP email is often transformed as it passes through the MTAs on the Internet, there is a chance that a valid digital signature will not be verifiable at the recipient end. This suggests that DomainKeys signing operations must be done at the boundary of sending and recipient domains.

Does DomainKeys Satisfy the Requirements For a Solution?

1. It must adequately authenticate the visible FROM address in the header.

Yes. Cryptographic principles and the assumed security of DNS provide domain-level authentication of the FROM address.

2. It must limit end-user training.

Partially. This assumes that all Internet domain MTAs will eventually provide the signing and verification services for end-users. As with SPF and Caller-ID, arriving at an agreed-upon standard for how to present authenticated FROM address information to the end-users in their MUAs will be a challenge for the email infrastructure community.



3. It must use standards-based technology.

Partially. While there is no existing digital signature protocol that maps to what the DomainKeys suggests, it does use many standards-based algorithms (SHA-1 and RSA, for example).

4. A unilateral deployment of a solution must add value.

No. For DomainKeys to work, both the sending and receiving parties must support the protocol.

5. It must be cost-effective for senders, recipients and email service providers.

Uncertain. There is clearly an upfront cost of enhancing Internet MTAs to support the protocol. Once embedded in these MTAs, ongoing administration of the protocol may be cost-effective, as it only requires a minimal set of public/private key pairs to be managed for an organization's entire outbound email. Inbound processing of DomainKeys signed email is slightly more CPU intensive than SPF or Caller-ID, but it may still be manageable by the large email processing domains. It is reasonable to assume that Yahoo! at least is willing to add the necessary processing power to its infrastructure.

How could phishers work around email infrastructure that uses Domain Keys?

- Because both sides of the email connection must support DomainKeys for it to work, recipient email servers must allow non-signed email through if the FROM domain doesn't participate in the DomainKeys protocol. The danger is that a phisher may choose to spoof emails using a domain he knows doesn't have a corresponding DomainKeys record in its DNS. Until all domains of possible phishing-victims support Domain Keys, phishers will continue to leverage the fact that an inbound MTA won't categorically drop a message coming from a domain that doesn't support the protocol.
- A phisher could still create a "cousin domain" of the organization he's trying to spoof. If a phisher can register Csecurityservice.com and appear like a part of C.com, he can easily create a domain key for Csecurityservice.com. This domain key will continue to be valid until Csecurityservice.com is blacklisted by all Internet MTAs. The lack of oversight for creating DomainKeys becomes a weakness in its ability to stop this type of "cousin domain" attack.

S/MIME Digital Signatures

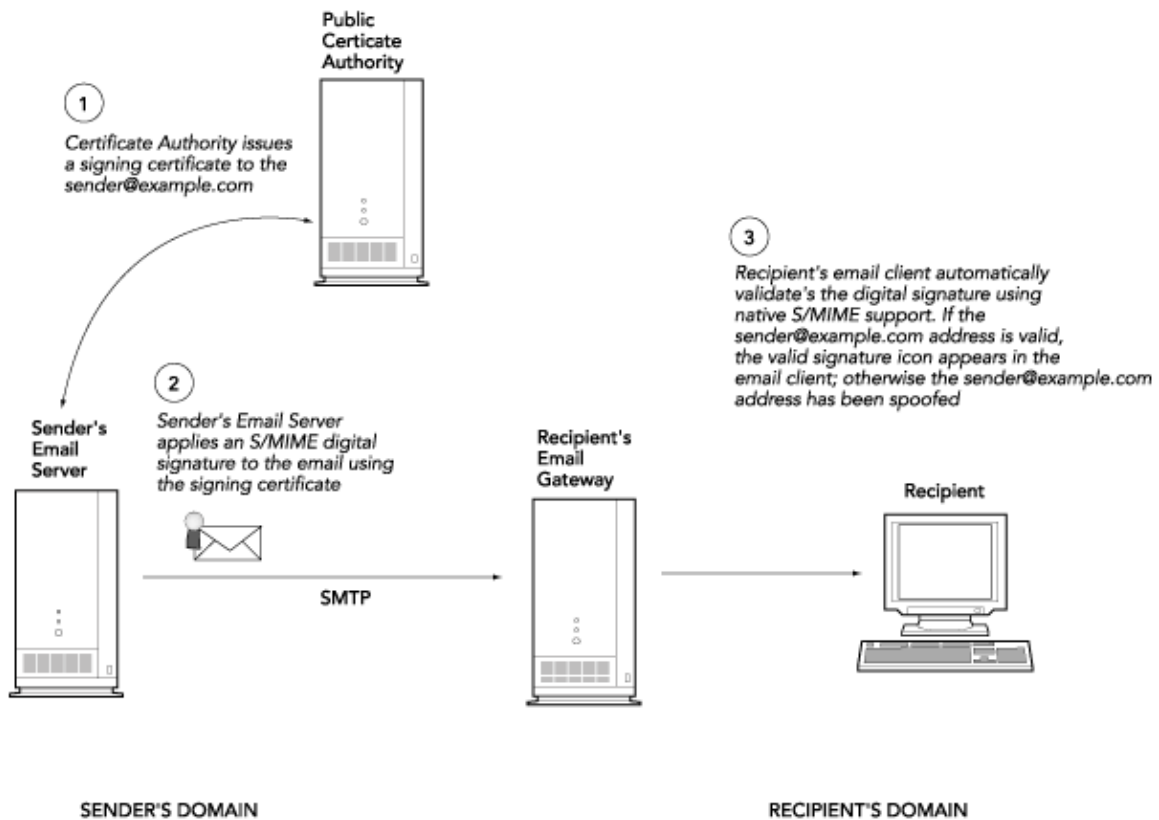
Like DomainKeys, S/MIME uses the power of asymmetric cryptography to provide sender authentication that cannot be easily spoofed. Unlike DomainKeys, S/MIME has the power to authenticate not just the domain but the actual sender of the email as listed in the FROM address. Also, unlike DomainKeys and any other proposed method for fighting phishing attacks, S/MIME has been deployed in the marketplace for several years as a general-purpose secure messaging protocol. While primarily used for encrypting email, S/MIME provides a powerful digital signature semantic that could help to authenticate email to deter phishing attacks.



S/MIME has been the acknowledged standard for providing both sender authentication through digital signatures and confidentiality through encryption since 1995. The protocol is currently undergoing final edits to version 3.1 in the IETF and there are multiple RFCs that describe various aspects of the standard. Over 350 million MUAs are deployed on the Internet today that support S/MIME. Each of these MUAs provides a visual cue to the user to describe the validity of an S/MIME digital signature.

A diagram and basic outline of how S/MIME would work to help deter phishing attacks is below:

S/MIME Digital Signatures



1. A public Certificate Authority (e.g. VeriSign, Thawte, GlobalSign, etc.) issues a signing certificate to an end-entity with a particular email address. The CA will have performed some level of verification that the end-entity in fact owns that email address.
2. Future emails originating from that address will then be digitally signed using the private key associated with the signing certificate issued in step 1. This digital signature provides a cryptographically secure way to prove that the message originated from the fully-qualified FROM address contained in the signing certificate. While it is assumed that the vast majority of signing operations will happen at the outbound email gateway to simplify the deployment process and speed adoption, signing at the sender's MUA is not precluded.



3. The recipient's S/MIME-enabled MUA attempts to validate the digital signature. If valid, the MUA presents a recognizable icon to the user that confirms the authenticity of the FROM address. If the signature is invalid, a different icon is presented warning the user the signature should not be trusted. MUAs that don't support S/MIME will still present the message contents normally as the digital signature will be an attachment to the email that is not processed.

The security of S/MIME lies primarily in the fact that the signing certificate is being issued by a trusted third party that must validate the sender's identity and ownership of the email address. Also, the private key associated with the signing certificate is guaranteed to be unique to the email address, so phishers would not be able to spoof this private key.

Advantages of S/MIME

- Over 350 million MUAs already support the standard and provide automatic, graphical cues about the authenticity of an email.
- Because S/MIME certificate issuance is controlled by an accredited third party (the Certificate Authority), legally binding contracts are in place to govern the process. Because CAs have a vested interest in maintaining good business practices, there will be sufficient verification of end-entities requesting digital certificates. This verification process can be extended to prevent the issuance of "cousin domain" signature certificates that the DomainKeys approach cannot address.
- Authentication of the fully qualified FROM address of the email provides non-repudiation services, which can provide additional value for business email.

Disadvantages of S/MIME

- It requires certificate registration and lifecycle management services from a public Certificate Authority. An accredited CA with a root certificate that is already available to all the S/MIME-enabled MUAs on the Internet must be involved in the process. This introduces additional costs to the sender not associated with the other proposals. However, it is possible that given the number of qualified CAs, competitive pressures will bring these costs down over time.
- The MUAs and MTAs of the email ISPS (e.g. AOL, MSN Hotmail, Yahoo!) do not support S/MIME. These ISP handle a significant amount of all Internet email traffic and are the predominant handlers of consumer email, often the target of phishing attacks.

Does S/MIME Satisfy the Requirements For a Solution?

1. **It must adequately authenticate the visible FROM address in the header.**

Yes. The third-party CAs provide the address validation before issuing the signing certificate. The strength of the cryptography used in the certificate issuance process and the application of the S/MIME digital signature prevents spoofing



2. It must limit end-user training.

Yes. The over 350 million MUAs that support S/MIME suggest that over half of all email recipients on the Internet could verify an S/MIME digital signature. End-users will have to be educated as to how their MUA graphically displays valid and invalid digital signatures. If the webmail ISPs would support S/MIME signature verification at their inbound gateways, then an overwhelming majority of Internet email infrastructure could support sender authentication using the same protocol. The webmail ISPs could educate their own users as to how valid and invalid digital signatures would be identified in their browser-based MUAs.

3. It must use standards-based technology.

Yes. S/MIME is a current standard in the IETF with multiple RFCs and has been in deployment for almost 10 years. The overwhelming number of email infrastructure vendors that support it in their MUAs make S/MIME a *de facto* standard for email security.

4. A unilateral deployment of a solution must add value.

Yes. Because so many MUAs already support S/MIME, an organization can choose to send S/MIME signed email and know that a majority of their recipients can validate the signature and trust the content. For those recipients whose email infrastructure doesn't support S/MIME, no harm is done as utilizing the clear-signing method within S/MIME ensures that these recipients can still at least read the contents in their MUA.

5. It must be cost-effective for senders, recipients and email service providers.

Yes. An organization need not sign all its outbound mail to protect itself from phishers spoofing their brand. Phishers gain little value in spoofing an email address of an identifiable employee of an organization. They use addresses like support@example.com, or security@example.com. Even very large organizations have a finite number (<50) of such system-based email addresses that are used to generate the type of customer-facing email that is so often spoofed by phishers. Acquiring this small number of signing certificates is not costly to the sending enterprise. A sufficient number of non-webmail recipient MUAs already support S/MIME, so no additional cost is incurred for recipients. The email ISPs who support webmail MUAs will have to build support for S/MIME verification in their inbound gateways. There are processing requirements for verifying S/MIME signed email beyond what other protocols require, but the additional load is typically only 10% above processing normal mail today.

How could phishers work around email infrastructure that uses S/MIME?

- A phisher could somehow convince an accredited CA to issue him a certificate for a spoofed email address. The CA who performs this illegal act will surely have its business ruined by reputation if not legal action if it performs this service knowingly. If it were to happen by accident, the improperly issued certificate could immediately be revoked to invalidate any future email signed with it.



- A phisher could still create a “cousin domain” of the organization he’s trying to spoof. If a phisher can register Csecurityservice.com and appear like a part of C.com, he may be able to register for a certificate for Csecurityservice.com from an accredited CA. This signing certificate will continue to be valid until it is revoked. However, end-users need not rely on revocation status alone to protect them from cousin domain certificates. The legitimate organization (C.com) can simply educate its users to ONLY expect digitally signed email coming from C.com or other approved sub-domains. So, even if a signed email from Csecurityservice.com appears valid to the recipient’s MUA, users can inspect the domain and know that it did not come from C.com.

Summary

It is clear that none of the proposed solutions perfectly meet all the requirements. As with most solutions to difficult problem, tradeoffs will need to be made. The SPF solution is immediately discounted as a compelling solution to stop phishing attacks because it neither authenticates the FROM address nor does it prohibit the easy workarounds phishers can utilize. Caller-ID provides some additional protection against phishing, but it is still vulnerable to phishing attacks until the overwhelming majority of legitimate domains use it. DomainKeys provides an interesting solution because it does directly authenticate the domain in the FROM address and it is a cryptographic solution that can be implemented with relatively low cost. However, like Caller-ID, the fact that not a single piece of email infrastructure supports the protocol today implies that there will be a non-trivial waiting period while Internet MTAs implement DomainKeys. Given the alarming growth of phishing attacks and the cost to users and business alike, neither Caller-ID nor DomainKeys will likely be implemented fast enough. In addition, the bugs in the protocols will have to be worked out over time at the expense of users and to the profit of phishers.

S/MIME remains as the only viable candidate to provide an immediate deterrent to phishing attacks. It authenticates the FROM address, it can be implemented today with immediate support across a majority of Internet email infrastructure, it has longevity as a standard in the IETF and the marketplace, and it provides an extensible framework to layer additional services on email that help solve other compelling business problems (e.g. non-repudiation, encryption). The two primary hurdles to S/MIME adoption can be overcome with concerted effort by the email ISPs to support it and the Certificate Authorities of the world to bring down the cost of certificate issuance.



The perception of certificate management (PKI) as a prohibitive complexity for enterprises can be mitigated by email infrastructure vendors who automate management for email-only certificates in their products going forward. Perhaps the most compelling argument for utilizing S/MIME to deter phishing attacks is one of urgency. Rather than wait for one or more new protocols to be ratified by a standards body or become de facto standards over the period of several months/years, why not use something that works today and can work side by side with other emerging protocols in the future? The cost of waiting is simply too high.

FOR MORE INFORMATION, PLEASE CALL 800.696.1978

Tumbleweed Communications Corp
700 Saginaw Drive
Redwood City, CA 94063
Phone 650.216.2000
Fax 650.216.2001
www.tumbleweed.com
info@tumbleweed.com

Copyright © 2004 Tumbleweed Communications Corp. All rights reserved. Tumbleweed is a registered trademark of Tumbleweed Communications Corp. All other brand names are trademarks of their respective holders.

SMIMEWP0304