Botnets can be markedly limited by authenticating every computer involved in an internet transaction, but traditionally this has been an unattainable goal as universal computer authentication would require the perfect and ongoing cooperation of a massive number of computer owners and systems administrators around the world.

Universal computer authentication can be achieved at the server level by a novel implementation of digital signature technology called Mail Transfer Agent Authentication. A second method, called Personal Computer Authentication, will authenticate all personal computers via a very different implementation of digital signatures.

These two systems are most easily understood by watching the following narrated PowerPoint that was presented at the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS2011):

Universal Computer Authentication (Part 1 of 2) [http://www.youtube.com/watch?v=dlufpuCXr-4]

Universal Computer Authentication (Part 2 of 2) [http://www.youtube.com/watch?v=y5klhGR-Vco]

These two methods will be effective because their implementation is independent of the participation of personal computer owners and systems administrators. The one-time participation of a handful of major software vendors is all that is needed to guarantee near universal computer authentication – these two methods therefore have a profound advantage over many other anti-botnet systems. Rather than replace other anti-botnet techniques these two systems will ideally serve as one layer of a multi-layered anti-botnet strategy.

A paper describing this technique in more detail follows, though the video presentation should be reviewed first.

Automatic Authentication of Email Servers and Personal Computers Independent of the Active Participation of Server Administrators or Personal Computer Users

Michael G. Kaplan
spamfizzle@gmail.com
spamfizzle.com

ABSTRACT

Universal email authentication is impossible with existing authentication schemes, namely DomainKeys Identified Mail (DKIM) and Sender Policy Framework (SPF), primarily because at a minimum this would require the ongoing participation of every domain administrator in the world. Consequently a vast quantity of current email is unauthenticated, thus empowering spammers. This paper describes two unique methods employing digital signatures to automatically authenticate every computer involved in sending an email. The first method will authenticate the mail transfer agent (MTA) used to forward an email, while the second method will authenticate the personal computer that originated the email. Universal authentication occurring redundantly at both the MTA and the personal computer is achievable because these two methods do not require the participation of email users or administrations.

The first method, **MTA Authentication**,will authenticate every MTA listed in an email header regardless of forwarding or the use of a dynamic IP address. This is made possible by having MTA software sign all outgoing email with an autonomously generated private key that is unique to that server. The distribution of the corresponding public key (an issue that plagues all other public key schemes) will require no human intervention as each mail server will automatically provide its public key to any computer in the world that queries it.

The second method, **Personal Computer Authentication**, will authenticate the personal computer used to send an email. The email client will sign all email by using a public key – the entire world can potentially use the same universally known public key. These digital signatures will encrypt not only the message hash but also a secret ID number that is unique to the personal computer. Receiving email systems will submit this encrypted digital signature to a single global database that will use the private key to decrypt the hash and the secret ID number. A reputation report corresponding to the secret ID number (but *not* the secret ID number itself) will be sent back to the receiving mail system. Personal computers will transparently acquire these secret ID numbers in a way that is resilient to botnets. Web browsers will employ a similar mechanism to authenticate personal computers used for webmail and other online transactions; one benefit of this will be that CAPTCHA can be eliminated.

Universal authentication via these methods is easily achievable as it requires only a onetime software update by the relatively miniscule number of developers of the MTA programs, email clients, and web browsers that are in common use.

1. INTRODUCTION

Malicious behavior over the internet is fostered by the inability to routinely authenticate the computers involved in a transaction. In the absence of special measures an email receiver can only be certain of the IP address of the final MTA to handle an email, and even this MTA's identity will be obscured if a dynamic IP address is used. An email receiver is also unable to authenticate the personal computer that sent the email. Website operators are also unable to authenticate the personal computer used to fill out a webpage registration for a service such as a free webmail account.

This paper describes two related and synergistic techniques, MTA Authentication and Personal Computer Authentication, to authenticate all of the computers used in an internet transaction. These systems avoid the pitfalls of other techniques by meeting the following criteria:

- Neither the active participation nor the awareness of administrators or users is required.
- The system is completely backwards compatible with existing email and internet infrastructure.
- The system will benefit the first users who deploy it. Widespread adoption is not needed before there is a benefit.
- A onetime update by a small handful of software vendors will result in near universal adoption of this system.
- Neither bounces nor any other potentially annoying backscatter will be generated. On the contrary backscatter will be reduced.

2. MTA AUTHENTICATION

MTA Authentication uses digital signatures to authenticate every MTA listed in an email envelop and guarantee that the email message remains unaltered once it has left an MTA. It will also allow an individual server to be repeatedly recognized regardless of its use of dynamic IP addresses. It works by having new versions of current MTA software upgraded to do the following:

1) Every server's MTA software will autonomously generate a single private key and its corresponding public key. The private key will forever remain hidden on the server that generated it.

There is no sharing of keys, not even between servers owned by a single organization.

2) The MTA will use its autonomously generated private key to digitally sign every email. A single private key unique to a single MTA will 'blindly' sign all email emanating from the MTA regardless of the sender's '**From**' address. This exclusively automated process authenticates the server instead of directly authenticating the domain.

In stark contrast DKIM is used to selectively sign emails from a specific domain. DKIM requires manual configuration and the manual distribution and manual management of keys across multiple servers to authenticate a domain.

3) The MTA will automatically report both its public key and the starting date when all of its outgoing mail was signed to any computer in the world that queries it. Email receivers will attain an MTA's public key by simply querying the MTA's IP address that is listed in the email header. The distribution of public keys in this manner requires absolutely no human intervention, thus perfectly solving the quandary of public key distribution that has plagued every other digital signature base authentication scheme (see Figure 1).

DKIM, in contrast, is dependent on domain administrators manually maintaining an updated list of public keys in the DNS record.

2.1 MTAs Listed in the Header are Authenticated despite Subsequent Forwarding

To illustrate this method we will look at the scenario of three different mail servers named MTA#1, MTA#2, and MTA#3. We will imagine that an email is sent to MTA#1, then forwarded to MTA#2, and finally forwarded to MTA#3 before reaching the email receiver. With conventional MTA software (meaning software that does *not* support MTA Authentication) the receiver can only be certain that the email came via MTA#3 as every other fact about the email can be fabricated.

Now imagine that these MTAs have upgraded to versions of software that *do* support MTA Authentication. After the upgrade MTA 1, 2, and 3 each autonomously generate a single public/private key pair and each MTA uses its private key to sign all outgoing email. Absolutely no manual configuration is required for this process so the MTA administrators can remain oblivious to this entire process.

Once again an email is sent to MTA#1, forwarded to MTA#2, and finally forwarded to MTA#3 before reaching the destination email system. This time, however, each MTA has signed the email before forwarding it to the next MTA. The anti-spam software of the destination email system will now directly query MTA#1 to attain its public key and verify the digital signature of MTA#1 present in the email envelope. The destination email system now has absolute certainty that the email originated from MTA#1 and that the email has not been altered since leaving MTA#1.

There actually is no need to even bother checking the signature of MTA#2 since the receiver knows that the email cannot have been modified since leaving MTA#1. The signature of MTA#2 will only need to be checked if MTA#1 had not signed the email – in this case the integrity of the email will be judged by the reputation of MTA#2.

2.1.1 Domains with SPF Records Can Finally Be Authenticated Despite Forwarding

Currently the most commonly used domain authentication protocol is Sender Policy Framework (SPF), whereby domain administrators publish a list of all of their domain's MTAs. One of the great flaws of SPF is that forwarding makes it impossible to use SPF records to authenticate the domain.

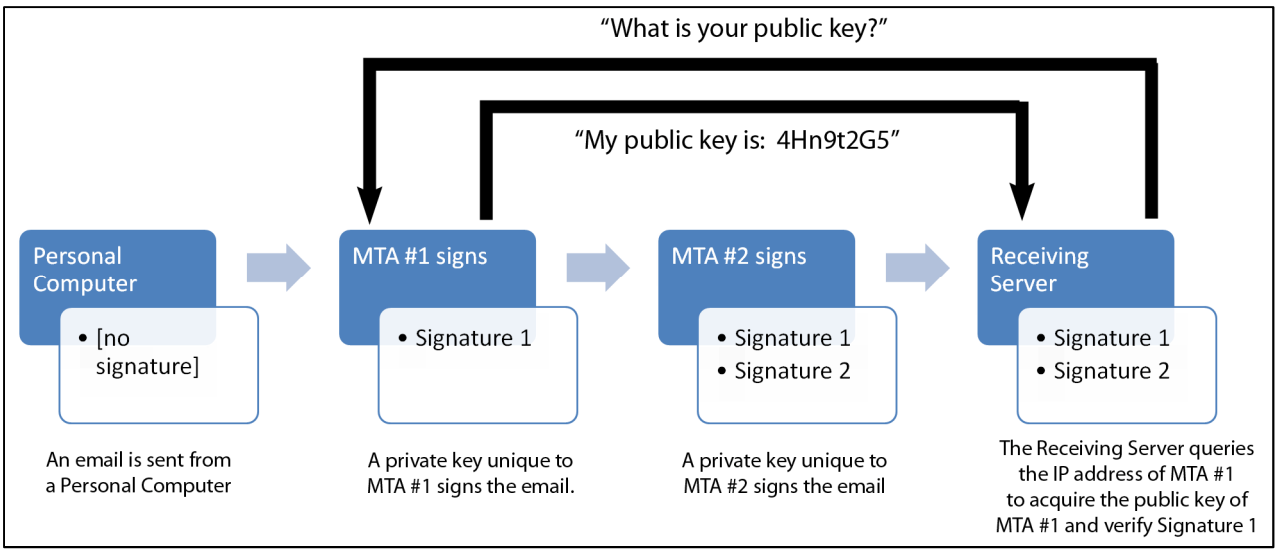

Figure 1. MTA Authentication showing each MTA signing an email. The Receiving Server is shown getting the public key of MTA #1 directly from MTA #1, so unlike other public key schemes no human intervention is involved in public key registration or distribution.

By using MTA Authentication email forwarded from SPF compliant domains will always receive an SPF PASS because the receiver is now absolutely sure that email originated from the first MTA listed in the email header and that the email was not altered while in transit.

2.1.2 MTA Authentication Compliant Servers Can Never Be Spoofed

Now we will imagine that a spam email arrives with MTA#1, MTA#2, and MTA#3 in the header but this time none of the MTAs has a signature. The receiver will *still* query the MTAs listed in the header about their signing policy. If any of these MTAs confirms that it always signs its email then the email receiver will know that the absence of a signature is proof that the email is spam.

Example: A user receives a forwarded email claiming to originate from Citibank. The IP address of the first MTA listed in the header corresponds to a true Citibank MTA but the second MTA does not. The email is not signed. The user now queries the IP address of the Citibank MTA and this MTA responds with the message "All outgoing email has been signed for the last 15 months". The email receiver can now mark this unsigned email as spam.

2.2 Dynamic IP Addresses Cease To Be a Problem

IP address reputation is a fundamental anti-spam tool. Databases such as Senderbase.org and DNS black lists are a valuable tool used by all anti-spam services. These databases are ideal for static IP addresses but they are often frustrated by dynamic IP addresses. A dynamic IP address may have been used by a spammer one day and an honest user the next. It is often difficult for these databases to even know if an IP is dynamic.

MTA Authentication solves the problem of tracking the reputation of a server that uses a dynamic IP addresses since a server's public key never changes. Existing IP address reputation databases will assign reputation scores to public keys the same way they currently assign reputation scores to IP addresses. For the first time these reputation databases will provide the full reputation history of a server despite its use of a dynamic IP address.

As a corollary IP address reputation databases will easily know if an IP address is dynamic because only a dynamic IP address will rotate through different public keys.

Example: An individual uses his laptop as an MTA to send the email for his personal domain. The laptop computer uses a new dynamic IP address every day, so consequently IP reputation databases are more likely to misclassify his email as spam. However, the laptop's MTA software has MTA Authentication functionality and for the past year the reputation of the laptop's public key has been building in the Senderbase.org database. The laptop now sends an email to a receiving mail server and this server queries the laptop for its public key (this can occur even before the receiving mail server has finished accepting the email). The email receiver can successfully review one year's worth of reputation data for this laptop-based MTA despite its use of a dynamic IP.

2.3 Reducing the Frequency of Broken Signatures due to Content Modification

Content modification to email while in transit can break a digital signature. Legitimate advertisements or certain notifications by mailing lists or antivirus companies are sometimes added as a footer to the email while in transit; the digital signature will not match the email if the addition of these footers is not accounted for. To address this problem the length of the original message will always be designated in the header; the receiver can now exclude the footer and only check the signature against the original email. Message length designation currently exists as an optional feature of DKIM, but it will be the default for MTA Authentication as well as for the process describe in section 3 of this paper.

Receiving mail systems, before delivering an email to a user's inbox, should excise footers that are added after the email is signed unless it is determined that the footer itself is not spam.

2.4 A Proposal for a New SPF Designation

Many domains accurately list all of their MTAs in their SPF record. Ideally these domains would instruct receiving email systems to reject all mail that is not from one of their listed servers by placing a '**-all**' tag in their SPF record. In reality most domains realize that if they use a '**-all**' tag and if their outgoing email gets forwarded then their email will be issued an SPF FAIL. Fear that this forwarded email will never reach the receiver forces these mail systems to apply the ambiguous tag of '**~all**' or '**?all**'. The receiving email system now has the burden of figuring out how to handle this ambiguity. Fear of listing '**-all**' results in more spam reaching the receiver and more backscatter in the form of bounces for the domain in question.

MTA Authentication ensures that forwarding will never break SPF. In response to this improvement a new SPF tag should be established that will effectively mean the following:

> "All of our servers use MTA Authentication so use '**-all**' if you check signatures. If you do not check signatures then use '**?all**' (or '**~all**')."

This will eliminate the sender's concern that forwarded mail will be issued an SPF FAIL. Receivers will receive less spam and less backscatter will be created because receivers will no longer send bounces to spoofed addresses.

2.4.1 MTA Authentication Combined with SPF Renders a 'DKIM Equivalent'

Combining an SPF record with MTA Authentication reproduces the security provided by all common deployments of DKIM. Effortlessly transforming SPF into a 'DKIM equivalent' is significant as the number of domains that deploy DKIM is only a fraction of the number that deploy SPF [1] [2].

2.5 Signing All Emails will not Overburden an MTA

Concerns that signing every outgoing email will result in a debilitating amount of CPU overhead for a mail server are unfounded. It was appreciated back in 2004 when DomainKeys was launched that the additional burden on MTAs was minimal, and processors today are much faster [3]. It is nearly impossible to detect the increased burden of signing emails against the background of common mail server tasks such as anti-virus scans [4].

3. PERSONAL COMPUTER AUTHENTICATION

Personal Computer Authentication uses digital signatures so that email receivers can verify the reputation of the personal computer that originally sent the email. The digital signature also guarantees that the email has not been altered while in transit. It promises to control spam sent via botnets while remaining unnoticed by individual users. In addition this technology will also be incorporated into web browsers to control malicious behavior on the Web; one benefit of this will be that the need for CAPTCHA will be eliminated.

Prior proposals [5] [6] to limit the amount of abusive behavior emanating from individual computers have often centered on a fixed payment for each internet transaction, typically in the form of a time-consuming computation. These proposals met with failure in part because virtually all spam is sent from botnets [7]. A per transaction payment large enough to cripple a botnet would be intolerable for legitimate users. Personal Computer Authentication also uses a payment system to limit abuse emanating from individual computers, but critical differences that make it practical include:

- There is no payment-per-transaction; instead there is a single payment-per-personal-computer-identity as represented by a secret ID number.
- Nearly every personal computer will acquire a secret ID number transparently and, from the users' perspective, it will cost nothing. Conversely relative extreme expense will profoundly limit the number of secret ID numbers that a botnet can acquire.

3.1 First Step: Distributing Secret ID Numbers to the World's Personal Computers

Each personal computer will identify itself via an individual secret ID number that is at least 128 bits in length; a trusted global reputation database will securely maintain a comprehensive list of these secret ID numbers. Personal computers will always encrypt this number before sending it to a third party such as an email receiver. This third party can relay this encrypted ID number to the trusted global database – a reputation report for this secret ID number (but not the secret ID itself) will then be sent to this third party.

This section discusses a few of the most universal and convenient ways to securely issue these secret ID numbers. Obviously there are many possible distribution methods.

3.1.1 Transparently Retrofitting Computers with Secret ID Numbers

Most desktops are Windows computers and each already has a product key in excess of 128 bits that is known only to Microsoft. A hash of this product key can be used as the secret ID number for Personal Computer Authentication. Microsoft can provide the global database with a comprehensive list of these hashes to be used as secret ID numbers.

Apple can securely retrofit Macintosh computers and iOS devices with secret ID numbers generated from preexisting unique identifiers within the Apple hardware.

Linux personal computers or personal computers with pirated versions of Microsoft Windows will acquire a new secret ID number at the time that the personal computer's email client and/or web browser is upgraded to support Personal Computer Authentication. The personal computer must then perform a time-consuming computation, and completion of this time-consuming computation will activate the secret ID number. Sharp disparities across computer systems raises concerns that proof-of-work computation time will vary greatly between users with high-end and low-end systems. The use of a memory-bound computation, as opposed to a processor-bound computation, will markedly minimize the disparity across systems [8]. Botnets will not be able to acquire large numbers of newly issued secret ID numbers (see section 3.4.1).

3.1.2 Future ID Numbers can be distributed via Network Cards

A simple addition to a hardware standard will ensure that every future computer is shipped with a secret ID number. Every network card has a ROM chip with a unique MAC address burned into it. Manufactures can burn a random 128 bit number into the network card ROM to serve as a secret ID. For enhanced security the manufacturers should only maintain and release a list of the hashes of these secret ID numbers.

Network card standards are regulated by the Institute of Electrical and Electronics Engineers (IEEE) [9]. The IEEE can amend the network card standard to stipulate that all network cards include these numbers.

3.2 Second Step: Upgraded Email Clients will Sign Every Email Sent from a Personal Computer

Personal Computer Authentication is an update to email clients that will place digital signatures on all outgoing email. These digital signatures are unique for the following reasons:

- All email sent by a personal computer is 'blindly' signed regardless of the identity of the email sender – the goal is to authenticate the personal computer, not the particular user.
- Conventional digital signature schemes such as DKIM create a signature with a private key, but this system instead uses a *public* key to create the digital signature. In principle every personal computer in the world can share just a single public key.
- The digital signature will not only contain the encrypted hash of the email message, but it will also contain the personal computer's encrypted secret ID number.

3.2.1 The Mechanism of Personal Computer Authentication for Email Clients

Personal Computer Authentication Clients works via the following steps (see Figure 2):

1. A user composes an email and hits 'Send'.
2. The email client takes the personal computer's secret ID and the message hash {secret ID + message hash} and encrypts them together using a public key. The resulting signature is inserted into the email envelop.
3. The email is sent and is received.
4. The receiver sends the encrypted signature to a trusted global database. This global database is the only entity that possesses the private key.
5. The global database decrypts the signature, thus revealing the secret ID number and the message hash. The message hash is sent back to the receiver along with the reputation report for the personal computer. The report will detail information such as email volume as well as feedback from previous email recipients. The global database *does not release the secret ID*; it is always kept secret.
6. The receiver authenticates the message content by running a hash on the email and comparing it to the hash sent back from the global database. The receiver uses the reputation report help determine if the email is spam.
7. The receiver will give feedback to the global database if the email is determined to be spam.

Personal Computer Authentication does not directly authenticate the '**From**' address but rather it allows for the reputation of the sending personal computer to be assessed directly. Spam originates from compromised computers, so authenticating the personal computer that sent an email may prove to be more useful than authenticating the '**From**' address.

Variations of this technique (e.g. issuing each personal computer its own cryptographic key instead of a secret ID number) are also possible. The subtle differences in function that these variations would entail do not justify exposition within the confines of this paper.

3.2.2 Near Universal Participation Requires Only a Few Major Players

Personal Computer Authentication requires an updated email client; otherwise no participation or even awareness is required by the sender. The email client market is dominated a handful of vendors so near universal participation will be achieved with the participation of just a few major players.

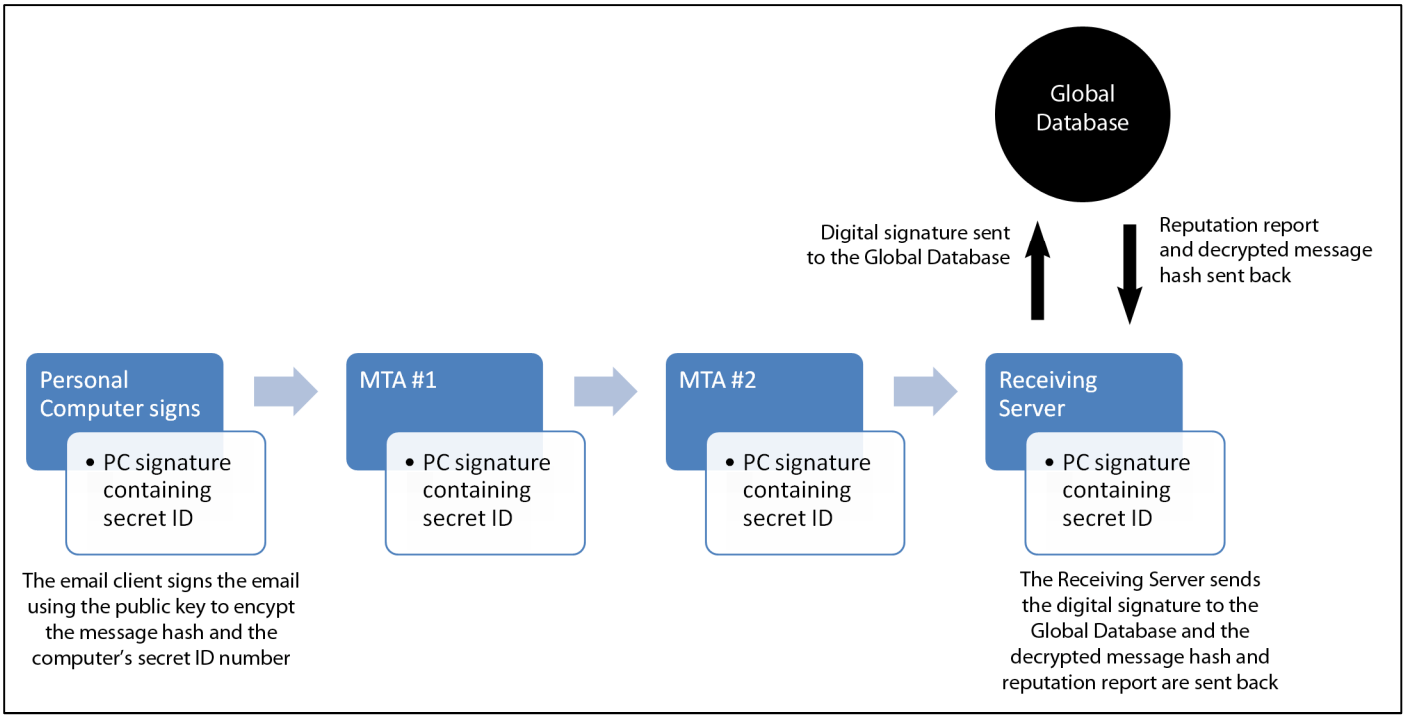

Figure 2. Personal Computer Authentication for Email Clients. The email client creates a signature by combining the message hash and the secret ID and using a public key to encrypt them. The Receiving Server will send this signature to the Global database and in return will receive the Personal Computer's reputation report.

3.3 Controlling Malicious Activity over the Web with Personal Computer Authentication by Web Browsers

Personal Computer Authentication will be implemented by the web browser using the same public key, secret ID number, and global database that is also used by the computer's email client as described in sections 3.1 and 3.2. The ability of a website's administrator to evaluate the reputation of the personal computer accessing a free service will be of tremendous utility in controlling abuse such as spammer webmail registrations and blog spam.

3.3.1 Personal Computer Authentication via Web Browsers Will Allow for the Elimination of CAPTCHA

CAPTCHA are currently essential despite the following flaws:

1. CAPTCHA hinder visually impaired people.
2. CAPTCHA are frequently defeated by spammer run computer vision software [10].
3. Spammers employ workers in developing nations to manually solve CAPTCHA on a massive scale [11].
4. CAPTCHA offer little impediment against individuals who repeatedly vandalize public message boards or wikis.
5. CAPTCHA, even in the best of circumstances, are slightly annoying and slightly time-consuming for pretty much everyone.

The modicum of security provided by CAPTCHA will be replaced by the higher level of security provided for by Personal Computer Authentication.

3.3.2 The Mechanics of Personal Computer Authentication by Web Browsers

We will start with the premise that the small handful of vendors for the commonly used web browsers have instituted this functionality via a onetime update. A user will bypass CAPTCHA while registering for a free online service such as a webmail account via the following process (see Figure 3):

1. A user accesses a webpage to complete an online registration for a free webmail account.
2. The webmail provider, detecting an updated browser, presents the user with a registration page devoid of a CAPTCHA.
3. The user's web browser takes the personal computer's secret ID number and encrypts it with a universally available public key provided by a global database. This is the same ID number, the same public key, and the same global database that is also used by this computer's email client described in sections 3.1 and 3.2. The web browser will encrypt the secret ID number along with a time-stamp (or even just some random number) so that the encrypted code is unique to that session, thus preventing a malicious entity from continuously resubmitting the encrypted secret ID number to the global database.
4. Before the browser sends the encrypted secret ID to the website a dialogue box will pop up on the user's browser with the text: "This website is requesting verification of your computer's reputation to complete this registration. Click 'OK' to allow." The user clicks on 'OK'.
5. The web browser sends the encrypted secret ID number to the webmail provider. The webmail provider forwards the encrypted secret ID to the global database and in return is provided with a reputation report. The report includes all reputation information connected to the secret ID number in question – i.e. both the web browser and the email client activity.
6. The webmail provider grants the user a free webmail account. CAPTCHA has been successfully bypassed.

The webmail provider will continue to give feedback to the global database regarding the activity of that particular email account and the global database will continue to provide updated

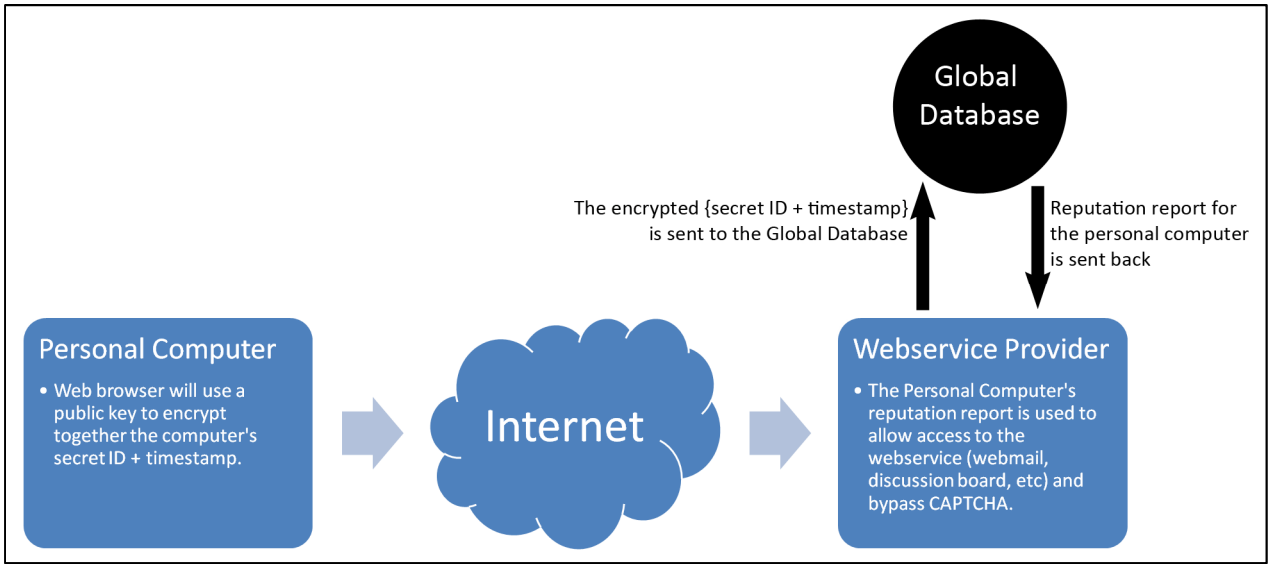

Figure 3. Personal Computer Authentication for Web Browsers. The web browser uses a public key to encrypt together the Personal Computer's secret ID and a timestamp and sends it to the Webservice Provider. The Webservice Provider sends this encrypted information to the Global Database for a reputation report. The Personal Computer user is transparently granted a webservice without needing to solve a CAPTCHA.

reputation reports upon request. By example a zombie may have used a single secret ID number to slowly register a large number of webmail accounts across several providers before attempting to simultaneously send large quantities of spam from these multiple accounts – this attempt will not succeed as the central database will report this activity back to the webmail providers.

Almost every computer user in the world uses a browser developed by one of five vendors, and the most commonly used email clients are also developed by some of these same vendors. Subsequently near universal adoption of Personal Computer Authentication will be rapidly realized with the participation of a handful of participants.

3.4 Countering Infected Computers

Spammers will inevitably infect a large number of computers and steal their secret ID numbers. Any stolen secret ID will be deactivated soon after it is used to send spam and the computer user will be alerted to the infection because their emails sent via clients will be rejected with codes such as:

> 550 5.7.1 Your email is rejected because your computer may be infected with a virus.

The computer user will encounter a similar security warning when attempting to use the web browser to access web services that make use of Personal Computer Authentication.

3.4.1 Replacing a Compromised Secret ID Number

The process of replacing a secret ID number will obligatorily start with the user's operating system downloading the most recent OS and web browser security patches – after this a new secret ID number will be downloaded. The personal computer may be required to perform a onetime time-consuming computation to activate the secret ID.

Recall that conventional pay-per-transaction schemes to limit spam are essentially futile as botnets can easily 'pay' any computational 'fee' as the computation time must remain brief so as to remain tolerable for legitimate users. A pay-per-identity scheme, however, is profoundly more resistant to spammer exploitation. In a pay-per-identity scheme a computer that has been cleansed of malware will need to do a time-consuming computation only once for the lifetime of that computer (unless it is infected again). In contrast a zombie computer will invalidate its secret ID number almost instantly after it is used to send spam, forcing the zombie to devote almost all of its time to computation instead of sending spam. The goal is not to make it absolutely impossible for a botnet to send spam, but rather to decrease by several orders of magnitude the amount of spam it can send.

Example: A zombie computer must perform a two hour computation to generate a new secret ID. Spam is then sent using this newly furnished secret ID. Feedback to the global reputation database causes this secret ID number to be blacklisted after just 10 spam emails. Therefore sending one billion spam will require more than 22,000 years of computing time – it will take a botnet of more than 24 million computers running eight hours daily to send one billion spam each day. Much of this spam will still not reach the user's inbox in part because it will also bear the suspicious stigma of being sent with a freshly minted secret ID number acquired solely via a computation, something that will be characteristic of only a miniscule amount of legitimate email.

3.5 Email from Reputable Personal Computers Residing behind Disreputable MTAs will be Safely Delivered

Some email senders with clean computers use an MTA that is also used by spammer compromised computers to send spam. Traditionally all email sent through such an MTA was tainted by the MTA's poor reputation. Personal Computer Authentication will help prevent a legitimate user's email from being misclassified as spam despite the poor reputation of the user's domain or MTA.

3.6 Privacy is Protected

The information within the reputation reports issued by the global database will be generalized enough to ensure a high degree of anonymity. It will generally not be possible to determine that two separate emails were sent by the same computer by comparing only the reputation reports.

Users that require an unusually high level of anonymity will be able to delve into the options menus of their email clients and web browsers so that the secret ID number that was included with their computer is not used. These users will then 'purchase' a new secret ID number by performing a time-consuming computation.

4. A PATH TO UNIVERSAL AUTHENTICATION

Widespread authentication via DKIM and SPF is impossible as it would require every domain administrator in the world to manually set up and then manually maintain accurate records in perpetuity. MTA Authentication and Personal Computer Authentication can achieve near universal authentication with a onetime software update by just a handful of major software vendors. MTA administrators and personal computer users can remain oblivious to this transparent and non-disruptive update, so there is little reason for software vendors to avoid instituting this update.

MTA Authentication prevents spammers from hiding behind dynamic IP addresses and it eliminates problems induced by email forwarding. Personal Computer Authentication uniquely provides a practical method to authenticate a personal computer's email and web browser transactions and allows for the seamless elimination of CAPTCHA. These two systems, by operating at two different levels, are synergistic. A comparison of key aspects of these two systems is listed in Table 1. They also complement, rather than replace, existing authentication methods. Instituting these authentication methods will require a miniscule fraction of the effort that existing schemes have required.

Table 1. Comparison of MTA Authentication and Personal Computer Authentication

	MTA Authentication	Personal Computer Authentication
How is it implemented?	A onetime update to existing MTA software by a relatively miniscule number of software vendors.	A onetime update to existing email clients and web browsers by a relatively miniscule number of software vendors.
Does each computer autonomously generate a public/private key pair?	Yes, each MTA generates its own unique public/private key pair.	No
Is a public or private key used to sign email?	A private key unique to the MTA.	A public key that is not unique to the personal computer.
Is a secret ID number unique to the computer encrypted within the signature?	No. (More sophisticated implementations can, but this is not discussed in this paper.)	Yes.
'Blind' signing of emails without regard to the sender's identity?	Yes.	Yes.
What is authenticated?	Each MTA	The personal computer
How do receivers check the signature?	The receiver retrieves the public key by querying the MTA's IP address. The public key is now used to check the email's signature.	The receiver submits the signature to a global database that has the private key and the decrypted message hash is sent back to the receiver.
What database does the receiver use to check the sending computer's reputation?	Currently existing MTA reputation databases (Senderbase.org, DNSBL, etc.).	The newly created global database will issue reputation reports for each personal computer.

5. REFERENCES

[1] McAfee Research Blog (July 3rd, 2008): SPF / DKIM Use on the Decline Among Fortune 500s? http://www.trustedsource.org/blog/130/SPF-DKIM-Use-on-the-Decline-Among-Fortune-500s

[2] Sendmail.org (2009): Fortune 1000 DKIM Survey http://www.sendmail.org/dkim/surveyFortune1000

[3] L. Seltzer (November 15, 2004): The Beginning of the Crypto Era, *Eweek.com* http://www.eweek.com/c/a/Security/The-Beginning-of-the-Crypto-Era/

[4] R. Guimerà: DKIM http://www.dkim.org/Misc/DKIM-perform-en.pdf

[5] A. Back (August 1, 2002): Hashcash - A Denial of Service Counter-Measure http://www.hashcash.org/papers/hashcash.pdf

[6] Microsoft Research: Penny Black http://research.microsoft.com/en-us/projects/PennyBlack/

[7] G. Cluley (January 2, 2009): Spammers defy Bill Gates's death of spam prophecy, *Sophos Blog* http://www.sophos.com/blogs/gc/g/2009/01/22/spammers-defy-bill-gatess-death-spam-prophecy/

[8] M. Abadi, M. Burrows, M. Manasse, T. Wobber (May 2005): Moderately Hard, Memory-bound Functions http://research.microsoft.com/pubs/54395/memory-longer-acm.pdf

[9] IEEE Registration Authority http://standards.ieee.org/regauth/index.html

[10] Websense Security Labs (February 15, 2009): Microsoft's CAPTCHA revolutions busted by spammers - again and again http://securitylabs.websense.com/content/Blogs/3306.aspx

[11] D. Danchev (August 29, 2008): Inside India's CAPTCHA solving economy http://blogs.zdnet.com/security/?p=1835