

# Achieving Email Security Usability

Phillip Hallam-Baker

Principal Scientist, VeriSign Inc.

## Abstract

Despite the widespread perception that email security is of critical importance cryptographic email security is very seldom used. Numerous solutions to the problem of securing email have been developed and standardized but these have proved difficult to deploy and use.

One of the main reasons for this difficulty is that each piece of the required technology has been developed independently as a generic platform on which security solutions may be built. As a consequence the user is left with an unacceptably complex configuration problem.

This paper proposes a means of providing transparent email security without the need for additional configuration based on existing security standards (XKMS, S/MIME, PGP, PKIX) and the recent DKIM standards proposal. Although the client deployment mode is considered the same approach would be equally applicable to an edge security configuration. Possible extensions of the protocol allow support for document level security approaches and to resist attack by quantum cryptanalysis.

## The Usability Problem

It is a truth universally acknowledged that an Internet user in possession of an email application must be in want of encryption.

Despite the strong and nearly universal belief in cryptographic security within the information security field, users have proven exceptionally reluctant to use the encryption features built into practically every major email program for close to a decade.

It is time for the security community to recognize that the users do not reject cryptographic solutions out of ignorance. They reject them because they are too difficult to use and often fail to meet their real security needs.

The cost of public key infrastructure that impedes deployment is mental rather than financial. Users do want security. But they are not prepared to do their work any differently or learn any new tools to achieve this. Users demand security that is completely seamless and transparent, built into the fabric of the Internet infrastructure.

The need for ubiquitous Internet security has never been more apparent or more acute. Internet crime is now a professional business conducted for profit. The twin engines of Internet crime are spam and networks of compromised computers (botnets). The lack of a ubiquitous email authentication infrastructure allows phishing

gangs to steal credit card numbers and access credentials by impersonating trusted brands.

The demand for usable security is critical even in classified applications that have traditionally relied on sophisticated operating systems designed to be secure at all costs<sup>1</sup>.

## What is usability?

A secure application should require no more training and be no more difficult to use than an insecure one.

In order to realize these goals it is necessary to:

- Employ consistent and familiar communication methods
- Eliminate all non-essential interaction
- Communicate all essential security information

While these goals may not prove to be sufficient it is clear that they are necessary and that current email security implementations do not achieve them.

## How current systems fail

Instead of being presented with a solution that provides security automatically and reliably the user is given a 'self assembly kit'.

Once the user has selected a Certificate Authority and enrolled for a digital certificate S/MIME allows her to sign individual email

messages or set a policy of signing all outbound email. If there is a digital certificate available for the recipient she may choose to send the message encrypted, or not.

For the average user this already represents a bewildering array of decisions but the user is still far from having a fully functional email security solution. She has not yet configured her LDAP directory or her SCVP interface. She has not loaded her smartcard drivers. And after completing all these tasks she will have to renew her certificate a year later when the original expires.

PGP suffers from similar usability problems, notably described by Whitten and Tygar<sup>2</sup> in 1999. Like most S/MIME interfaces the PGP 5.0 interface described in the paper is designed with the goal of allowing the user to use cryptography as if this was the end rather than merely the means.

Later versions of PGP, notably PGP Universal have attempted to overcome the usability deficit. However this has been achieved by having “declared peace in the certificate and message format debates”<sup>3</sup> and essentially implementing every variant of every standard. As such PGP universal is agnostic on the critical question as to which software architecture is most likely to enable a ubiquitous Internet wide email security infrastructure.

Traditional PGP offers the non-technical user an even more puzzling requirement. Before they can use their key they should get it signed by one or preferably several other PGP users that they already know.

Enterprise strength PKI systems allow network administrators to substantially mitigate this pain for the enterprise user. The personal Internet user is left on their own. Their perception of their security needs and thus their tolerance for deployment pain is very substantially lower, yet as the problem of phishing demonstrates personal Internet users have more than sufficient assets to be the target of professional Internet criminals. Personal users may have less confidential information to be stolen but they have money that can be stolen and they are much more likely to be tricked into parting with it.

## The deployment problem

*"Philosophers have only interpreted the world in various ways, the point is to change it" – Karl Marx*

In the mid 1990s a considerable effort went in to ensuring that every major email client supported the S/MIME protocol. But even though this top-down ‘deployment’ was almost completely successful in making secure email available to over a billion users it was entirely unsuccessful in persuading them to use it.

The bottom-up deployment strategy of PGP was only marginally more successful. PGP persuaded a significant minority within the technical community to install and configure a security plug in. But even amongst this community security is the exception, not the rule. Only a tiny number of PGP key holders use it every day. Neither protocol has succeeded in achieving ubiquitous use today, nor is there reason to believe that this will change in the future.

## Metcalf’s law and its corollary

Metcalf’s law states that the value of a network is proportional to the number of people it reaches. Metcalf’s law is often quoted in the context of breathless pitches for ‘viral marketing’ programs premised on the fact that once a network has gained ‘critical mass’ its growth becomes self-sustaining.

The unfortunate corollary to Metcalf’s law is the chicken and egg problem. The same process of positive feedback can cause a network that has not reached critical mass to quickly *loose* members. The Internet now has over a billion users and ‘critical mass’ for an application is likely to be several tens of millions of active users.

The problem of network effects is even more acute when a new network is in competition with an established one. If an S/MIME signature is added to an email there is a small but significant risk that the receiver will not be able to read it. Some email programs cannot process messages in S/MIME format. Other programs can process the message but display it to the user in a distinctly unhelpful fashion. An early version of the Internet access software provided by one major ISP displays a helpful message ‘warning’ the user that a signed email has been received.

## The installed base

As we have seen the success of any new security infrastructure depends in large measure on how it interacts with the existing infrastructure.

In particular the development cycles for client applications are typically three years or more<sup>i</sup> and at any given time at least half of the installed base of applications is three years old or more.

It is clearly desirable for a security proposal to be as compatible with the installed base as is possible. But it is unrealistic to expect that legacy systems will be as secure as those that are updated.

It is important that a secure email protocol be compatible with the legacy infrastructure but it is also important that expectations be realistic. It is essential for legacy users to be able to communicate and interact with secured systems. It is neither essential nor realistic to expect a new security protocol to offer infallible protection for the user who does not have an up to date application or whose machine has been compromised by a Trojan.

#### **Essential criteria**

- Provide acceptable security and usability when used with an aware client
- Provide acceptable usability when used with a non-aware client

#### **Non-Essential criteria**

- Provide protection against bug exploits in legacy applications or platforms.
- Provide protection when the user's machine has been compromised by a Trojan.

### **Early adopter community**

The usual solution to this corollary is to identify a community of early adopters with an urgent need for an email security solution that meets a particular need within that community.

The early adopter generally targeted for this approach is government, in particular the United States Government. In the early days of the Internet the US government and government funded research institutions represented a clear majority of Internet users.

---

<sup>i</sup> For example consider the release cycle of Microsoft Windows for home use, major updates occurring in 1995, 1998 and 2001<sup>[4]</sup>

The problem with this approach is that the needs of early adopter communities tend to be specialized. A solution that meets these needs may not meet the needs of Internet users as a whole. Early adopter communities are also likely to be tolerant of usability problems that are show stoppers for Internet users as a whole.

The problem of specialist needs is particularly acute in the US government. In addition to being considerably larger and more complex than the largest corporation the US government has considerably more information to protect and a greater need to keep it secure. The military alone has over 1.4 million active duty personnel, 1.2 million reservists, a further 654,000 civilian employees and indirectly employs a similar number of contractors<sup>5</sup>. In addition approximately two million retirees and family members receive benefits. In comparison Wal-Mart, the worlds largest corporate employer has 1.6 million employees<sup>6</sup>.

Early adopter communities can also be unrepresentative of even their own needs. The US government certainly has a need for a security infrastructure that allows confidential and classified information to be protected. But it is not clear that these needs are met by an email security protocol. A classified document should be encrypted whether it is stored on disk or traveling over the Internet. This requirement is more appropriately met by document level security systems being developed in the context of Trustworthy Computing and Digital Rights Management.

It appears that S/MIME has failed to meet government needs by offering too little even as it has failed to achieve widespread deployment by requiring too much.

### **Pain Point**

Deployment of new Internet infrastructure is expensive and time consuming. This expense is only likely to be met by a security protocol if it meets a critical pain point that is urgently felt at the time it is being deployed.

Unlike the 'early adopter' strategy which attempted to identify a subset of users for whom the proposal represents a 'killer application' in the 'pain point' strategy we attempt to identify particular functionality that addresses an issue of immediate and urgent concern for the community of Internet users as a whole.

The pain that is being felt most urgently on the Internet today is caused by Internet crime, in particular spam and phishing<sup>7</sup>.

## Bootstrap strategy

Addressing an urgent pain point is a necessary requirement for achieving a critical mass of support. If we are not careful however we may end up with a proposal that meets the requirements for addressing the pain point and only those requirements. Instead of establishing a ubiquitous and pervasive security infrastructure for all email we will have only succeeded in meeting our current needs with no plan for extending the solution scope in the future.

Future-proofing a solution is particularly important in the context of Internet crime. Professional Internet criminals seek the largest return for the least amount of effort. Phishing spam is not their first criminal tactic to exploit the lack of security in email and unless we have a comprehensive email security plan it is unlikely to be the last.

## Accountability not Control

Since its beginnings the field information security has been dominated by government needs and in particular academic perception of military needs. This has led to the development of security systems designed to control access to information:

### Control Approach

- Authentication: Who is making the request?
- Authorization: Is the request permitted for this party?

The control approach is based on the assumption that there is a clearly defined set of parties, a clearly defined set of rules that are to be applied and that both the rules and the parties to which they are to be applied are known in advance.

There is no set of rules that can be written *in advance* that will infallibly identify spam email without mistake yet it is easy to recognize spam when it is received.

Not only do these assumptions fail when applied to a public network, they also fail for a large number of real world situations. Motorists are deterred from speeding through fines, license

suspensions and prison terms rather than being prevented from speeding using a speed limiter. Even if every motorist was required to install a speed limiter this would only prevent one type of traffic violation; it would still be necessary to use the deterrence approach to control reckless driving, driving under the influence of alcohol.

The glue that holds social networks together is *accountability* rather than control. Control based security systems are not applicable to the principle security issues facing the Internet today: the problems of Internet crime, in particular spam and phishing. Nor should it be a surprise that the Internet security problems that have not been solved today are the ones which the control approach is not suited for. The problems for which it is suited have already been solved.

The accountability approach to information security is better suited to applications where the consequences of individual security failures are small but the aggregate consequences of many small security failures are significant.

### Accountability Approach

- Authentication: Who should be held accountable?
- Authorization: What the likelihood of compliance?
- Consequences for default

As in the control approach the first two steps in the accountability triad are authentication and authorization. The principle difference is that in the control approach authorization is the last step in the process. The authorization decision is binary; access is either granted or withheld.

In the control approach there is a bias towards refusing access unless the criteria for granting it are met. The Internet security problems that have proved intractable using the control approach are problems where the consequences of incorrectly granting access on a single occasion are small (a single spam is an annoyance) but the consequences of incorrectly granting access on a large number of occasions are severe (a thousand spam messages a day is a crisis).

In the accountability approach there is a bias towards granting access, provided that we are confident that there will be significant consequences if the other party defaults. This is a much closer match to our typical 'real world' behavior than the principle of 'do nothing until

completely sure' that characterizes the control approach.

The consequences of default may be loss of use, civil actions or even criminal prosecution. What is important in the accountability approach is that the perceived probability of the consequences being imposed and the consequences themselves be sufficient to deter an unacceptable rate of default.

## The Responsibility Problem

Domain Keys Identified Mail (DKIM<sup>8</sup>) is an email authentication technology that allows an email sender, forwarder or mailing list to *claim responsibility* for an email message. A party that claims responsibility for an email message informs the recipient that they can be held accountable and thus may increase the probability that the intended recipient will accept it.

Although DKIM does not and cannot solve the spam problem directly, DKIM allows email senders who volunteer to be held accountable to distinguish themselves from likely spammers. The spammers have a vast array of tactics but each and every one is designed to avoid the spammer being held accountable.

The DKIM message signature format allows a signature to be added to an email message without requiring modification of the message body. This ensures that (unlike S/MIME or PGP) the addition of a signature to an email does not negatively impact any recipient. Another significant departure from previous schemes is that recipients are advised to treat a message carrying a signature that cannot be verified as if it were unsigned.

The DKIM sender signature policy record allows a domain name owner to explicitly deny responsibility for unsigned mail message by stating that all authentic mail is signed. This makes it possible for an email recipient to conclude that an unsigned message is likely to be a forgery, a conclusion that is not possible with any of the previous cryptographic email security proposals.

## Edge Architecture

Unlike the traditional approaches that attempted to identify the individual responsible for sending the email, DKIM is designed to identify a

domain name owner that take responsibility for the email. The Internet has a billion users, attempting to hold each and every user accountable for sending unwanted email is a futile effort. Holding ISPs, Corporations, Schools and Universities accountable for policing their own users is much more promising.

In particular the DKIM architecture is designed to the assumption that messages are signed at the outbound email edge server of a network rather than by individual who sent it. On the receiving side the design is optimized to meet the needs of a signature verification filter at the incoming email edge server. In most cases this filter would be a part of a spam and virus filtering solution.

The edge architecture of DKIM allows for rapid deployment as an organization can deploy DKIM through an infrastructure upgrade limited to the email servers.

## DNS Key Distribution

DKIM is a highly focused proposal designed to solve the responsibility problem using minimal extensions to existing protocols and infrastructures. Instead of proposing deployment of a new Public Key Infrastructure for key distribution DKIM keys are distributed through the DNS using unsigned public key values stored in a standard text record.

Using the DNS to provide the key distribution mechanism allows any email sender to start accepting responsibility for outbound email by signing it without requiring the sender to deploy any new infrastructure beyond adding the email signature module to their outbound mail server and adding a small number of text records to their DNS.

The disadvantage to this approach is that the key distribution mechanism is limited by the architecture of DNS which is designed to provide a fast response to contemporaneous requests. The DNS has no concept of history and there is no way to ask 'what did this DNS record look like two months ago'. While this is not a significant constraint when an email message is being validated in-transit (e.g. at the inbound email edge server) the DNS is not an ideal infrastructure for serving the key distribution needs of an email client which might want to verify a signature on an email opened hours, days or even months after it was originally sent.

## The Authenticity Problem

Traditional email security approaches consider confidentiality and integrity to be complimentary tasks that are equally important. This assumption introduces a subtle bias into the architecture as it is assumed that senders and receivers must both upgrade their email clients to exchange secure mail.

This assumption certainly holds for encrypted mail where a recipient must have the means to decrypt the message in order to read it. But the assumption that a recipient must have the means to check the signature on a signed mail before reading it is a major departure from existing practice. It has led to a situation where S/MIME signatures cannot be used against the problem of phishing because of the minority of email readers that are unable to present a signed message to the user in an acceptable fashion.

The problem of phishing highlights the need to consider authenticity separately from the problem of integrity. It is much more important that a recipient be able to identify the sender of an email than know with certainty that the content has not been modified in any respect since.

Traditional email security approaches have attempted to identify the sender of an email by means of an X.500 distinguished name or an RFC 822 email address. The second approach has proved more successful than the first but still allows email senders to be impersonated through use of 'cousin' or 'look-alike' domains. DKIM allows 'AnyBank' to prevent an attacker successfully impersonating anybank.com. DKIM does not prevent the attacker registering a similar domain name such as any-bank.com or anybank-security.com. The introduction of internationalized domain names<sup>9</sup> provides additional scope for this type of attack.

A phishing impersonation attack is directed at the weakest link in the security chain, the gap between the computer screen and the user's head. To close that gap the authenticity of the message must be demonstrated using cues that are familiar to the user. A user cannot and should not be expected to recognize AnyBank by its Domain name any more than by its telephone number or ABA routing number. Customers recognize businesses in the physical world by their brands. Every large bank has a team of people whose sole job is ensuring that every

piece of information issued by the bank, every letter, every credit card, every ATM is consistently branded with the current logo. To solve the authentication problem the same cues must be applied to Internet communications.

## Secure Internet Letterhead

Secure Internet Letterhead is a proposal for a comprehensive Internet authentication infrastructure that allows every trustworthy Internet communication to be securely marked by a trusted brand.

The SSL padlock interface is designed to tell the user 'if the padlock icon is present *the domain name component in the address bar can be trusted*'. The Secure Internet Letterhead approach is direct: 'if the trusted brand logo appears in the secure area of the browser *it can be trusted*'.

For a user interface component to be trustworthy it must always be trustworthy. DNS Domain Names and X.500 distinguished names were both designed to provide a directory function. Attempting to overload this function and in addition use them as a security indicator is doomed. Secure Internet Letterhead introduces a new indicator whose sole purpose is to provide a security indicator.

If the authentication mechanism is to be successful it must be applied consistently and ubiquitously. In addition to its application to email described in this paper work is underway to apply the same principles and underlying technology to Web transactions (using SSL) and to Internet Messaging, telephony and Video.

Secure Internet Letterhead is a realization of the PKIX LogoType extension proposed by Stefan Santesson et. al., expected to be accredited as an IETF draft standard in the near future.<sup>10</sup> The PKIX LogoType extension allows a certificate issuer to embed links to one or more logos representing the brands of the certificate subject and/or issuer.

Linking a certificate record to a DKIM public key record<sup>11</sup> allows the DKIM signature format to be used as a vehicle for applying secure letterhead. The brand of the message sender is only shown if the message signature verifies and the signature key is authenticated by an X.509v3 certificate carrying the corresponding LogoType extension that is issued by a trusted certificate issuer (Figure 1).



**Figure 1: DKIM Secure Letterhead**

The prototype implementation of Secure Internet Letterhead was developed as a Web Mail interface. This approach was chosen to further the deployment strategy. If one or more of the principal providers of Web Mail services were to deploy Secure Internet Letterhead critical mass would be achieved instantly. Even adoption by a single Web Mail provider would provide a compelling business case for Financial Institutions targeted by phishing to obtain a Secure Letterhead certificate.

### Qui Custodiet Custodes?

The security of Secure Internet Letterhead is critically dependent on the trustworthiness of the certificate issuers. If an attacker can persuade a Certificate Authority to issue them a certificate with a logo that impersonates a trusted brand the introduction of letterhead makes the phishing problem considerably worse.

Various control based mechanisms have been proposed to ensure that Certificate Authorities carry out their duties accurately and effectively. Like all control based security approaches these suffer from the weakness that they can only define minimum standards for compliance. Control based security does nothing to encourage the development of improved authentication criteria above and beyond the minimum.

The most appropriate way to ensure the trustworthiness of Certificate Authorities in an accountability based security scheme is to apply accountability principles to the problem. Displaying the issuer logo to the user, either directly in the email message dialog or through a 'pop-up' or 'mouse-over' window forces the Certificate Authority to put its own brand on the line every time a certificate is issued (Figure 2).

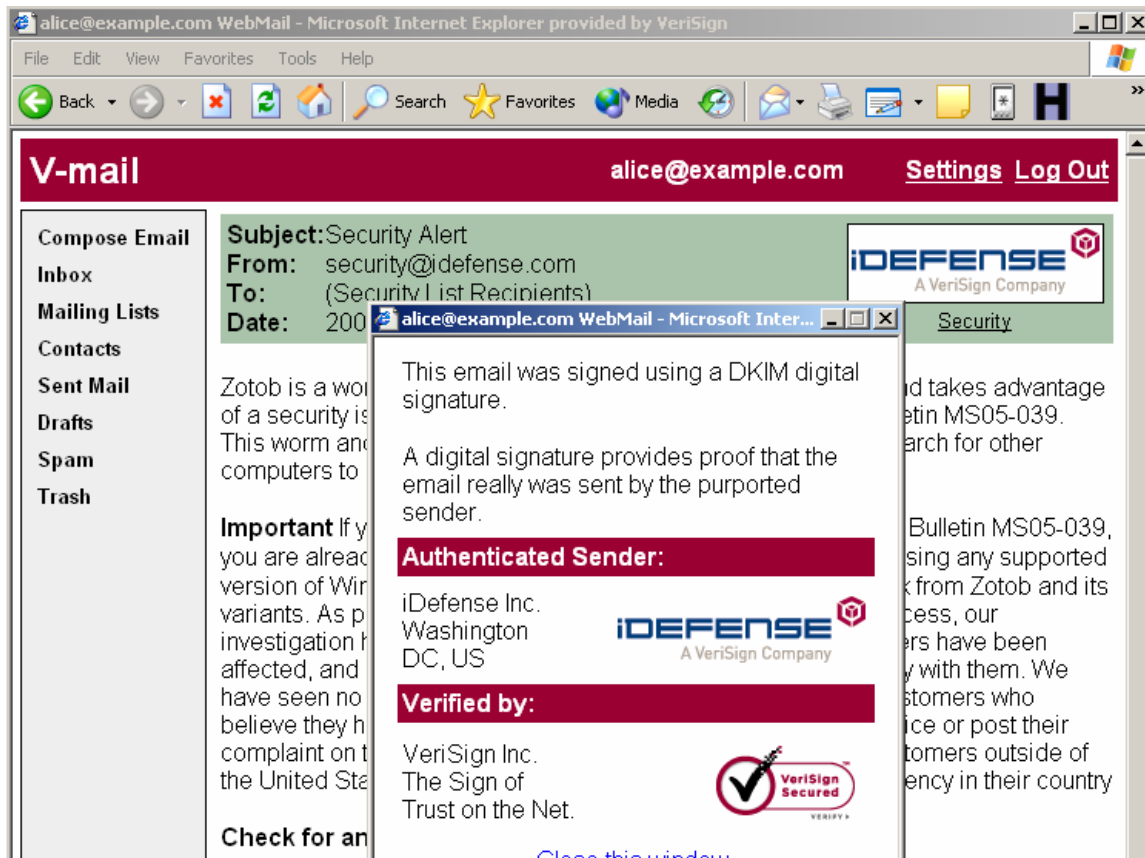


Figure 2 DKIM Secure Letterhead Issuer Logo

While effective authentication processes and rigorous quality control can minimize the risk of issuing a fraudulent certificate no amount of prior investigation can ensure that the Certificate subject will not default at a future date. Even the best known and trusted brand can be acquired by a company that is later discovered to be run by crooks and swindlers. For secure Letterhead to be trustworthy as well as merely trusted it is essential for the Certificate Authority to support rapid revocation of keys that are used fraudulently. For example by supporting a real time certificate status protocol such as OCSP<sup>12</sup>.

## Client Application Validation

The DKIM protocol combined with Secure Letterhead provides a robust solution to the authentication problem for users of hosted Web Mail services. As previously discussed however, DNS begins to show weaknesses as a key distribution infrastructure when signature verification is performed offline in the email client rather than during the transaction flow by the messaging infrastructure. A signature verifier

can expect a DNS record to still be available minutes or hours after the message was sent. Demanding records to be available at an indefinite time in the future represents a significant change to the operational requirements of DNS.

For signature validation in the client application to be viable, persistent credentials are required. DNS is not designed to provide a persistent credential repository but other existing PKI protocols are. In particular XKMS<sup>13</sup> was designed to provide a persistent store for PKI credentials that is entirely agnostic with respect to the architecture of the underlying PKI. Like the DKIM DNS based key distribution model, XKMS realizes a key centric PKI model similar to the original Public Key Directory model proposed by Diffie and Hellman<sup>14</sup>. XKMS may also be used as a gateway to a traditional certificate based PKI following the Kohnfelder model<sup>15</sup>.

The DKIM signature format allows additional key distribution mechanisms to be specified by means of an attribute. In a typical application both key distribution mechanisms would be

supported. This allows in-transaction signature verification filters to acquire keys quickly while ensuring that the needs of offline clients for a persistent and dependable key distribution infrastructure are both met.

## Per User Signatures

Support for signature verification in the email client extends the scope of the DKIM signature to the receiving end of the communication. It is logical to look for ways in which the scope of the security context can be extended to the sending end of the communication, allowing the individual email sender to sign their correspondence with their own individual key.

Even though support for 'per-user' keying is outside the scope of the initial DKIM charter the base specification provides all the mechanism necessary to sign messages with individual user keys and to use them for message validation.

What the base DKIM specification lacks is support for management of the private key lifecycle. This is not a major concern for deployment at the edge. Even a large enterprise is unlikely to need more than a ten or a hundred domain keys. With 'per user' keying even a moderately sized enterprise may quickly find that it is managing hundreds, thousands or even hundreds of thousands of keys. Domain names tend to be relatively stable but students, employees and customers come and go. Unless the secure email client application provides support for key lifecycle management per user-keying quickly becomes unmanageable.

## Key Lifecycle management with XKRSS

Fortunately XKMS also provides for key lifecycle management. The XML Key Registration Service Specification (XKRSS) component of XKMS is designed to support registration, reissue, revocation and recovery of private keys.

An XKRSS client may be written from scratch in a few days if an XML parsing library is available and open source toolkits are available for many languages.

## The Configuration Problem

As the experience of S/MIME deployment demonstrates, support for a security feature is unlikely to be used if the end user is required to

make an effort to configure it. XKMS supports automatic discovery of the local XKRSS registration service using the DNS service discovery (SRV) record<sup>16</sup>.

If the user's email address is `alice@example.com` an XKMS aware client can discover the DNS address of the local XKRSS service by requesting the SRV record `_XKMS_XKRSS_SOAP_HTTP._tcp.example.com`. Once the XKRSS service is located the email client can register keys for any purpose they are required for: signature, encryption or key exchange.

The development of a prototype implementation revealed a minor shortcoming in this aspect of the XKMS design. The only way that the XKMS client can discover the features supported by the XKMS service is to attempt each one in turn. A richer service description language would allow the XKMS service to tell the client which services are available.

## Encryption

DKIM, X.509 certificates and XKMS provide all the support necessary to support a comprehensive yet completely user friendly email authentication mechanism. Adding support for encryption completes the requirements for secure email as they are traditionally understood.

Instead of proposing yet another email message encryption format however we observe that the existing S/MIME<sup>17</sup> and PGP<sup>18</sup> message formats provide almost everything that is needed. While either message format would meet the technical requirements support for both formats is required to meet the political constraints created by the S/MIME vs. PGP standards war. To date this struggle has reached a stalemate, S/MIME dominates deployment but PGP dominates in mindshare. The quickest way to resolve this stalemate is to declare both formats winners and move on.

## Problems

Although the S/MIME and PGP message formats are entirely sufficient both protocols have significant usability defects that must be addressed if our deployment criteria are to be met.

## Key Distribution

The principle defect in the most commonly used implementations of the traditional email encryption formats is that both lack an effective mechanism for key distribution. Given an email address `alice@example.com` there is no simple process for locating the encryption key to use to send email to that address.

XKMS, and two recent PKIX extensions, PKIXREP<sup>19</sup> and the proposed CERTStore<sup>20</sup> extension solve this problem by allowing the email sender to discover the location of the key distribution service for the recipient using the same SRV mechanism used to discover an XKMS registration service.

Once the key distribution mechanism is made automatic an email client can be configured to automatically encrypt outgoing messages whenever an encryption key is available for the recipient. Email encryption becomes entirely seamless and automatic.

## Encryption is Message Body Only

In S/MIME and PGP the SMTP encryption is applied to the message body alone, the subject line is left unencrypted despite the fact that the subject line is very likely to contain confidential content. As a result the legitimate expectations of the user are not met.

Solving this particular problem requires only the recognition that it is more important to meet the security expectations of the user. The solution adopted in the prototype is to introduce a confidentiality option into the email composition window. If the confidentiality option is selected the email client ensures that the entire message is encrypted by moving the subject line into the message body and adding a new subject line 'Confidential' or if applicable 'Client confidential – Attorney work product privilege asserted'.

If the confidentiality option is selected and it is not possible to send the message encrypted the user is warned. The user is given the option of canceling the message sending the message without encryption. The user might also be given the option of having the message printed out and sent by courier or sending the recipient a notice telling her to retrieve the message from a secured Web site.

## Security is End to End Only

Although some effort has been made to introduce an edge-to-edge model to both PGP and S/MIME both specifications are essentially predicated on an end-to-end security model.

This causes particular difficulty where encryption is concerned since many enterprises do not want to accept encrypted email messages unless they are certain that they do not contain a virus or other form of executable code. Nor is end-to-end encryption likely to be acceptable to end users if it renders spam filtering measures inoperative.

Another source of difficulty with end to end encryption is the current trend towards receiving email on a wide variety of portable and mobile devices. It is not unlikely for a user to require access to their email by means of a desktop, laptop and PDA. The end to end principle is also inappropriate in the context of a Web mail service.

The XKMS architecture allows the domain name owner to control key distribution infrastructure for and hence the use of encryption in their domain. If the domain name owner wants to ensure that encrypted email can be read by virus scanning or compliance systems at the incoming edge server this can be achieved by returning the public key of the edge server in response to key location requests.

While this violates a core premise of the traditional email security protocols, that the end user should be empowered to control their own security, domain names are inexpensive. The user who feels the need for 'empowerment' and has the ability and inclination to control their own security can readily do so by obtaining their own domain name.

After decryption at the email edge server the message may be re-encrypted under the end-user's key. The resulting 'encryption with a gap' need not mean a weaker security solution than the traditional end to end approach. For most enterprises the risk of trojan code bypassing their firewall and anti-virus filters is considerably greater than the risk of unintended disclosure of confidential information. If a trojan is loose inside the enterprise the security of the email system is moot in any case.

In cases where the 'encryption gap' is a concern, the process of decryption, scanning for active code and re-encryption could be performed by

trustworthy hardware configured to refuse any administrative interference.

## Complex Trust Infrastructures

The protocol profile described so far allows authentication and encryption capabilities to be added to an email application with a minimum of code and without affecting usability. While these capabilities are likely to be sufficient to meet the security needs of most enterprises they do not necessarily meet the needs of an enterprise which has already achieved a substantial deployment of a sophisticated PKI built on traditional principles.

Fortunately XKMS provides an answer to these cases as well. All that is necessary is for the email application that is attempting to locate an encryption or signature key to delegate the task to a local XKMS Validate service discovered using the same DNS SRV mechanism used to discover Locate and Registration services.

During the development of the prototype a minor bug was discovered in the XKMS specification which only defines a single SRV prefix for identifying an XKISS Locate or Validate service. While these functions might be combined in a single server the Locate service is primarily concerned with servicing external requests and the Validate service is like the Registration service essentially an exclusive service for the local domain.

It is therefore more likely that a Validate service would be combined with a Registration service than a Locate service. A simple solution to this oversight is to define a separate SRV prefix for the Validate service:

`_VALIDATE_XKMS_XKRSS_SOAP_HTTP`

## DNS Security

A possible objection to the use of the DNS as a key distribution or service discovery mechanism as described in this paper is that the security of the key distribution infrastructure is ultimately dependent on the security of the DNS, a protocol that does not currently have a deployed cryptographic security infrastructure. While DNS security has not proved to be a source of chronic security problems as email has it is clearly unsatisfactory for the security of a cryptographic security protocol to rely on an insecure infrastructure.

Fortunately DNSSEC<sup>21</sup> meets this objection for both XKMS and the DKIM DNS key distribution. The principal obstacle to DNSSEC deployment has been the lack of a compelling use case for the domain name owner. The professional Internet criminal attacks the weakest, most profitable link in the chain. Until the systemic security failures of email are addressed the security shortcomings of the DNS are practically irrelevant. Using the DNS as the lynchpin of a ubiquitous cryptographic security system for email creates one of the strongest business cases imaginable.

## Responding to change

As previously mentioned one of the most important tests of a security infrastructure is its ability to respond to changing needs. While it is impossible to foresee every need a system that is designed to meet the foreseeable needs is much more likely to meet unforeseen needs as well.

## Document Lifecycle Security

The next major step forward in Information security is likely to be a transition from transport and message based protection to schemes that protect the integrity and confidentiality of *documents* throughout their entire life cycle. While an email message *may* contain sensitive information an attached spreadsheet titled 'Accounts' is almost certain to.

Various schemes for 'Digital Rights Management' or 'Content Management' have been proposed but in practice most effectively end at the enterprise border. Without the ability to exchange the necessary key information across the open Internet it is not possible for the CFO to send a document to external counsel for review, a sales person to send confidential contract proposal to a customer or meet many similar real world business security needs.

Although the XKMS based key distribution system and SRV discovery mechanism described in this paper is applied to the PGP and S/MIME encryption formats it could in principle be extended to support DRM or CM encryption formats as well. Alternatively if this approach proved to be too constraining the same SRV discovery mechanism could be applied to a SAML<sup>22</sup> service publishing the appropriate authorization assertions.

## Incremental Advances in Cryptology

An ongoing concern for every developer of a cryptographic protocol is that advances in cryptanalysis might result in the underlying cryptographic algorithms being compromised.

Fortunately there is good reason to believe that DKIM and XKMS both offer realistic mechanisms for achieving a transition from one encryption algorithm to another. A paper simulation of a transition from the current RSA based signature algorithm to an ECC algorithm was conducted with satisfactory conclusions<sup>23</sup>.

## Quantum Computing

The worst case scenario for developments in cryptanalysis is the development of a quantum computer capable of performing calculations of significant complexity. Such a machine could in principle break every public key algorithm currently in use and it is prudent to assume that this represents an intrinsic property of public key algorithms.

Fortunately quantum computing is not currently believed to threaten symmetric key algorithms in the same degree and even the best quantum computer cannot factor an RSA public key it does not know. These premises and a minor modification to the XKMS key information protocol allow an XKMS configuration to be established which is secure even if the adversary has a quantum computer yet remains compatible with legacy systems.

In the standard public key model everyone who wants to send an encrypted message to Alice uses the same public key. In the modified model a separate key pair is established for each correspondent. The key Alice discloses to Bob is different from the key she discloses to Carol. The use of separate key pairs for each bilateral relationship allows the keys to be kept confidential so that Alice's public key used to receive encrypted email from Bob is only disclosed to Bob. Mallet cannot then cryptanalyze the key no matter how effective his quantum computer might be.

In effect the XKMS services at both ends of the communication act in the manner of a Kerberos<sup>24</sup> Key Distribution Center. The keying material that Bob receives from Alice's XKMS Locate service has an additional element carrying the

private key encrypted under a symmetric key shared only by Alice and the XKMS Service.

The requirement for public keys to be kept private effectively eliminates the flexibility and convenience that makes public key cryptography such an attractive technology. In effect the parties end up with the convenience of a symmetric system and the performance of an asymmetric one. This is however an acceptable price to pay in the context of a worst case scenario in which the objective is to transition the network from the use of public key based technology to a symmetric system without a loss of service or functionality.

The only addition required to the XKMS protocol is the specification of appropriate algorithm identifiers and (as keys are now specific to a relationship between two users rather than just a key holder) a mechanism to allow the counterparty to the communication to be specified. A possible objection to this approach is that each message would have to contain both a public and a private key. The use of a public key encryption mechanism such as ECC that supports a more compact public key would meet this objection.

## Conclusions

The problems of deploying ubiquitous email security are significant but as this paper demonstrates may be met by using a combination of existing protocols which are with the sole exception of DKIM all existing standards. The challenge of email security is thus similar to the challenge facing the field of networked hypertext applications in the early 1990s. The components all exist. The challenge that must be met is integrating those components in such a way that the user experience is fluent, seamless and learned automatically.

Despite the insistence that the user interface be at least as simple as the user interface for insecure email the system described in this paper offers at least as much security as existing schemes. It is not only possible to achieve usability and security, it is impossible to achieve security in practice unless an uncompromising approach is taken to both.

## Acknowledgements

This paper has greatly benefited from the work and insights of many people. In particular Nico Popp, Siddharth Bajaj, Alex Deacon and Jeff

Burstein at VeriSign and Mark Delaney, Miles Libbey (Yahoo), Jim Fenton (Cisco), John Levine, Harry Khatz (Microsoft), Barry Leiba (IBM) and Stephen Farrell (Trinity College Dublin) in the DKIM working group. The Secure Letterhead concept was developed from concepts originally proposed by Stefan Santesson and refined by Amir Herzberg at Haifa University.

---

1 **Central Intelligence Agency Inspector General Report Of Investigation Improper Handling Of Classified Information By John M. Deutch** February 18, 2000

2 **Alma Whitten and J. D. Tygar**. Why Johnny Can't Encrypt, 8th Usenix Security Symposium, 1999

3 **Jon Callas**, PGP Inc. CTO, <http://www.pgp.com/library/ctocorner/automagic.html>

4 **Microsoft Inc**, <http://www.microsoft.com/windows/lifecycle/default.msp>

5 **US Department of Defense** statistic, see e.g. [http://www.defenselink.mil/pubs/dod101/dod101\\_for\\_2002.html](http://www.defenselink.mil/pubs/dod101/dod101_for_2002.html)

6 **WalMart Inc**. see: <http://walmartstores.com/GlobalWMStoresWeb/navigate.do?catg=1>

7 **Phillip Hallam Baker**, *The dotCrime Manifesto, To be Published*

8 **E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas**, *DomainKeys Identified Mail (DKIM)*, IETF Draft, July 9, 2005

9 **P. Faltstrom, P. Hoffman, A. Costello**, *FRC 3490 Internationalizing Domain Names in Applications (IDNA)*, March 2003 <http://www.ietf.org/rfc/rfc3490.txt>

10 **S. Santesson, R. Housley, T. Freeman**, *RFC 3709 - Internet X.509 Public Key Infrastructure: Logotypes in X.509 Certificates*, IETF, February 2004, <http://www.ietf.org/rfc/rfc3709.txt>

11 **Phillip Hallam-Baker**, *Use of PKIX Certificates in DKIM*, September 2004, <http://www.ietf.org/internet-drafts/draft-dkim-pkix-00.txt>

12 **M. Myers, R. Ankney, A. Malpani, S. Galperin, C. Adams**, *RFC 2560 X.509 Internet Public Key Infrastructure Online Certificate*

---

*Status Protocol – OCSP*, IETF, June 1999.

<http://www.ietf.org/rfc/rfc2560.txt>

13 **Phillip Hallam-Baker, Shivaram H. Mysore**, *XML Key Management Specification (XKMS 2.0)*, W3C Recommendation 28 June 2005, XKMS <http://www.w3.org/TR/xkms2/>

14 **W. Diffie and M.E. Hellman**, *New directions in cryptography*, IEEE Trans. Inform. Theory, IT-22, 6, 1976, pp.644-654.

15 **Kohnfelder**, *Toward a Practical Public Key Cryptosystem*, in Department of Electrical Engineering. 1978, MIT.

16 **A. Gulbrandsen, P. Vixie, L. Esibov**, *RFC 2782 A DNS RR for specifying the location of services (DNS SRV)*. IETF, February 2000. <http://www.ietf.org/rfc/rfc2782.txt>.

17 **B. Ramsdell**, *RFC 3851 Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification*, IETF, July 2004, <http://www.ietf.org/rfc/rfc3851.txt>

18 **J. Callas, L. Donnerhacker, H. Finney, R. Thayer**, *OpenPGP Message Format*, IETF, November 1998, <http://www.ietf.org/rfc/rfc2440.txt>

19 **S. Boeyen and P. Hallam-Baker**, *Internet X.509 Public Key Infrastructure Repository Locator Service*, RFC 4386, <http://www.ietf.org/rfc/rfc4386.txt>

20 **Peter Gutmann**, *Certificate Store Access via HTTP*, RFC 4387 <http://www.ietf.org/rfc/rfc4387.txt>

21 **R. Arends, R. Austein, M. Larson, D. Massey, S. Rose**, *RFC 4033 DNS Security Introduction and Requirements*, IETF, March 2005, <http://www.ietf.org/rfc/rfc4033.txt>

22 **E. Maler et al.**, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-core-1.1 <http://www.oasis-open.org/committees/security/>

23 **Phillip Hallam-Baker**, *DKIM Transitions*, To be published

24 **B. Clifford Neuman and Theodore Ts'o**, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, 32(9) pp33-38. September 1994, <http://gost.isi.edu/publications/kerberos-neuman-tso.html>