Messaging, Malware and Mobile Anti-Abuse Working Group
M³AAWG Describes Costs Associated with Using Crypto

March 2017

The reference URL for this document: www.m3aawg.org/Crypto-Costs

I. Introduction

Deploying opportunistic encryption as described in TLS for Mail: M³AAWG Initial Recommendations is an excellent way to start protecting email traffic between providers. Using Forward Secrecy to Secure Data is a further step providers can take. Forward secrecy ensures that encrypted traffic can never be decrypted, even if the relevant private keys are somehow eventually obtained. However, most everything, including cryptographic secrecy and privacy, comes at a cost. This document describes the budget and other costs associated with using cryptography to help the reader make an informed decision about what to do, or not do, when faced with the need to deploy encryption.

II. When Needed, Content based Spam and Malware Filtering Should Be Done On-System, Not Passively On-Network Links

While encryption protects against unwanted eavesdropping or tampering, it also precludes passive network monitoring[1] for beneficial purposes, such as blocking spam or filtering malware. Traffic inspection is still possible; however, it just needs to be done on the endpoints before the traffic gets encrypted or after the traffic gets decrypted.

In thinking about opportunities to do traffic inspection, it is important to distinguish between two cases:

1. Hop-by Hop Encryption
   In the hop-by-hop encryption case (for example, opportunistic SSL/TLS for SMTP[2]), traffic is encrypted and then decrypted for each hop (e.g., each link in the delivery chain). As a result, there are opportunities for filtering, and unfortunately, for eavesdropping or tampering at each intervening node. The integrity and trustworthiness of all intervening nodes thus becomes critical.

2. End-to-End Encryption[3]
   In the case of end-to-end encryption [e.g., PGP (Pretty Good Privacy)[4]/GPG (GNU Privacy Guard)[5] or S/MIME[6]], a message is encrypted at its origin and decrypted at its destination. It is carried in encrypted form for the totality of its time in transit and even while stored on disk before being read by the user. Therefore, the contents of an end-to-end encrypted message can only be inspected or filtered before encryption at its origin or after decryption at its destination.

A message can potentially be protected by hop-by-hop encryption, end-to-end encryption, both, or neither. Doing both is preferred as this choice allows the strength of each encryption option to complement the other.

III. Other Potential Loss of Functionality

Using encryption can result in the loss of other desired functionality:

1. If a user's mail spool contains end-to-end encrypted messages, those encrypted messages cannot be routinely and efficiently searched for messages relevant to a particular sender or issue without decrypting each message first.

2. Similarly, most mailing lists do not support routinely encrypted mail distribution; e.g., when inbound mail submissions are encrypted with the list's own key, they are automatically decrypted by the list manager software upon receipt and then re-encrypted with each subscriber's own public key outbound prior to distribution.

3. Debugging encrypted connections is more difficult than debugging plain text connections; e.g., you cannot just use Wireshark to monitor a network conversation.

IV. Potential Irrecoverable Loss of Encrypted Contents

Crypto does a great job of protecting sensitive content from unauthorized access. However, that protection brings a new risk of its own: if a non-escrowed key is lost or forgotten, any content encrypted with that key will be totally irrecoverable, even by authorized users. For example, imagine a research scientist who has encrypted her research results with a non-escrowed key. If that scientist is killed in an accident and no one else has a copy of her key, that scientist's encrypted discoveries may be lost forever.

On the other hand, escrowed keys, unless implemented with great care, have the potential to act as a "back door," allowing unauthorized access to confidential material. One approach is to allow escrowed key recovery for emergency purposes, such as in the accidental death of the scientist mentioned above, but to require mandatory key revocation if or when key recovery procedures are exercised.

Fortunately, this is not a consideration with opportunistic encryption. On the other hand, key recovery options are a personal or corporate decision for end-to-end email in a personal or business context.

V. Incrementally Increased Effort and Inconvenience

Deploying encryption requires some incremental effort and inconvenience, albeit hardly at a "Herculean" level.

In the case of hop-by-hop encryption, mail system administrators need to configure their MTAs (Mail Transfer Agents)[7] to have opportunistic TLS enabled.[8] A globally-trusted SSL/TLS certificate also needs to be purchased (potentially for less than $5 per system), installed and periodically updated. Cryptographic libraries – such as OpenSSL[9] – need to be installed and kept up to date with current upstream releases. Interoperability issues may occasionally crop up and need to be resolved. All of these issues are minor, but real, and should therefore be explicitly recognized.

In the case of end-to-end encryption, the burden shifts from the provider to the end-user and that burden becomes more significant, at least initially. To be able to implement end-to-end encryption, the user needs to have:

- Required skills and knowledge
  These can be acquired from studying documentation, receiving basic training in the use of the tools, or having access to a mentor who is willing to coach the user through the process of getting

cryptographic software set up and operational. M³AAWG has previously offered training on using S/MIME[10] and PGP/GPG.

- Cryptographic software
  In some cases, such as S/MIME, the required cryptographic software might come fully integrated with popular email client programs such as Microsoft Outlook[11] or Mozilla Thunderbird.[12]

  In other cases, such as PGP/GPG, the required software will need to be downloaded and installed before any crypto activity can take place. Most PGP/GPG users might also want to install a "point-and-click" GUI integration "shim" such as Enigmail,[13] that makes it easier to use PGP/GPG with email clients such as Thunderbird.

- Keys and identities
  Each user of end-to-end encryption will also need to create a public/private key pair, tie that key pair to their identity, and have some mechanism for sharing their public key with potential correspondents.

- S/MIME handles the creation of a public/private key pair during provisioning of a personal certificate.

  - Personal certificates are:
    - tied to an email identity
    - issued by a certificate authority, and
    - "chain" hierarchically back to a globally trusted root, such as a web server or mail server SSL/TLS.

  - Key exchange happens automatically in the S/MIME case when digitally signed messages are exchanged or S/MIME keys can also be obtained from an enterprise directory.

- PGP/GPG requires the user to explicitly create a public/private key pair. That key pair is normally signed by other PGP/GPG users, thereby creating a "web of trust." Those public keys then get uploaded to a public key server or are directly shared between correspondents.

Many people can and do use end-to-end encryption, even many who do are not professional technicians. However, sadly, end-to-end encryption setup is often perceived as being enough of a hassle that hardly any – less than 1/10th of 1 percent of all internet email – is routinely encrypted end-to-end.

Deployment of opportunistic SSL/TLS – by way of contrast – is doing far better, roughly 88 percent of all email sent from Google as of January 2017 is now being opportunistically encrypted by default.[14] That value is quite consistent with the level of opportunistic encryption publicly reported by Facebook.[15]

VI. Potential Loss of Anonymity

Paradoxically, while cryptography increases privacy by increasing resistance to eavesdropping, it may also decrease anonymity. Consider PGP/GPG. If used in traditional "web of trust" mode, a "real" identity, e.g., as established by multiple parties inspecting government issued identification documents, gets bound to a public/private key pair. (We note for the record that it is it is entirely possible to use PGP/GPG without tying an identity to a key pair.) At that point, any content digitally signed with that private key is non-

repudiably tied to the associated identity and any content symmetrically signed with the private key is also non-repudiably tied to that identity.

Also consider S/MIME. It checks the OCSP (Online Certificate Status Protocol)[16] to ensure that a personal certificate has not been revoked. That OCSP checking process potentially ties the recipient IP address, e.g., the IP address checking the OCSP status, to the personal certificate identity being checked. That is an identity leak if exploited by a CA (Certificate Authority) or someone monitoring them. On the other hand, if you do not validate the non-revoked status of each personal certificate, you run the risk of accepting a certificate that has been revoked.

VII. Cryptographic "Failure Modes" Often Tend to Be Brittle and Failures Are Often Undifferentiated

Consider DNSSEC,[17] which relies on cryptography to protect DNS resolution, i.e., the translation of domain names to IP addresses and vice versa, against cache poisoning attacks.[18] That is a worthy objective, and one which at least some M^{3}AAWG member companies have fully embraced.[19]

However, if the DNSSEC keys for a site are misconfigured or allowed to expire, the DNS data for that zone will not validate. That validation failure will be handled – "signaled to users" if you will – by making it appear as if those misconfigured or expired zones simply do not exist. That is a fairly "nuclear" signaling mechanism, particularly since those zones will continue to work fine at sites that do not validate the DNSSEC status of domains.

That is also the networking equivalent of a warning indicator light in a car: you may know that something is wrong, but you do not know what. It is undifferentiated. Maybe a key expired, maybe a record was created incorrectly, maybe something else went wrong. You just do not know. That is often the nature of cryptographic protocols.

VIII. What About Computational Overhead?

An often-mentioned potential downside of cryptographically protecting traffic is that strong cryptography can impose "computational overhead" or "slow things down." In actuality, however, multiple parties have reported that the computational overhead associated with doing strong crypto on modern Unix-based hardware is not even noticeable. M^{3}AAWG welcomes any empirical benchmarking studies that member companies or other entities might be willing to share, and in the meantime, M^{3}AAWG encourages the reader to do their own testing and their own due diligence on this point.

IX. Conclusion

While there are "costs" to doing anything and everything, the Messaging, Malware and Mobile Anti-Abuse Working Group believes the "costs" associated with deploying encryption should not be a "show stopper," that is, should not be a barrier to employing encryption. This guidance is not meant to be treated as comprehensive; it is part of an ongoing series of documents from M^{3}AAWG available under the Best Practices section at www.m3aawg.org that are meant to help improve the protection of user messaging.

X. References

[1] Passive Monitoring, http://en.wikipedia.org/wiki/Passive_monitoring

[2] Opportunistic Encryption, http://en.wikipedia.org/wiki/Opportunistic_encryption

[3] End-to-End Encryption, http://en.wikipedia.org/wiki/End-to-end_encryption

[4] Pretty Good Privacy, http://en.wikipedia.org/wiki/Pretty_Good_Privacy

[5] GNU Privacy Guard, http://en.wikipedia.org/wiki/GNU_Privacy_Guard

[6] S/MIME, http://en.wikipedia.org/wiki/S/MIME

[7] Message Transfer Agent, http://en.wikipedia.org/wiki/Message_transfer_agent

[8] See for example, http://www.postfix.org/TLS_README.html

[9] OpenSSL Software Foundation, https://www.openssl.org/

[10] "Client Certs and S/MIME Signing and Encryption: An Introduction," https://www.stsauver.com/joe/maawg24/maawg24.pdf

[11] Microsoft Office – Outlook, http://products.office.com/en-us/outlook/email-and-calendar-software-microsoft-outlook

[12] Mozilla Corporation – Thunderbird, https://www.mozilla.org/en-US/thunderbird/

[13] The Enigmail Project, https://www.enigmail.net/home/index.php

[14] Google Transparency Report, https://www.google.com/transparencyreport/saferemail/

[15] The Current State of SMTP STARTTLS Deployment, https://m.facebook.com/notes/protect-the-graph/the-current-state-of-smtp-starttls-deployment/1453015901605223/

[16] Online Certificate Status Protocol, http://en.wikipedia.org/wiki/Online_Certificate_Status_Protocol

[17] Domain Name System Security Extensions, http://en.wikipedia.org/wiki/Domain_Name_System_Security_Extensions

[18] DNS spoofing, http://en.wikipedia.org/wiki/DNS_spoofing

[19] Comcast Completes DNSSEC Deployment, http://corporate.comcast.com/comcast-voices/comcast-completes-dnssec-deployment

As with all best practices that we publish, please check the M³AAWG website (www.m3aawg.org) for updates to this document.

M3AAWG109