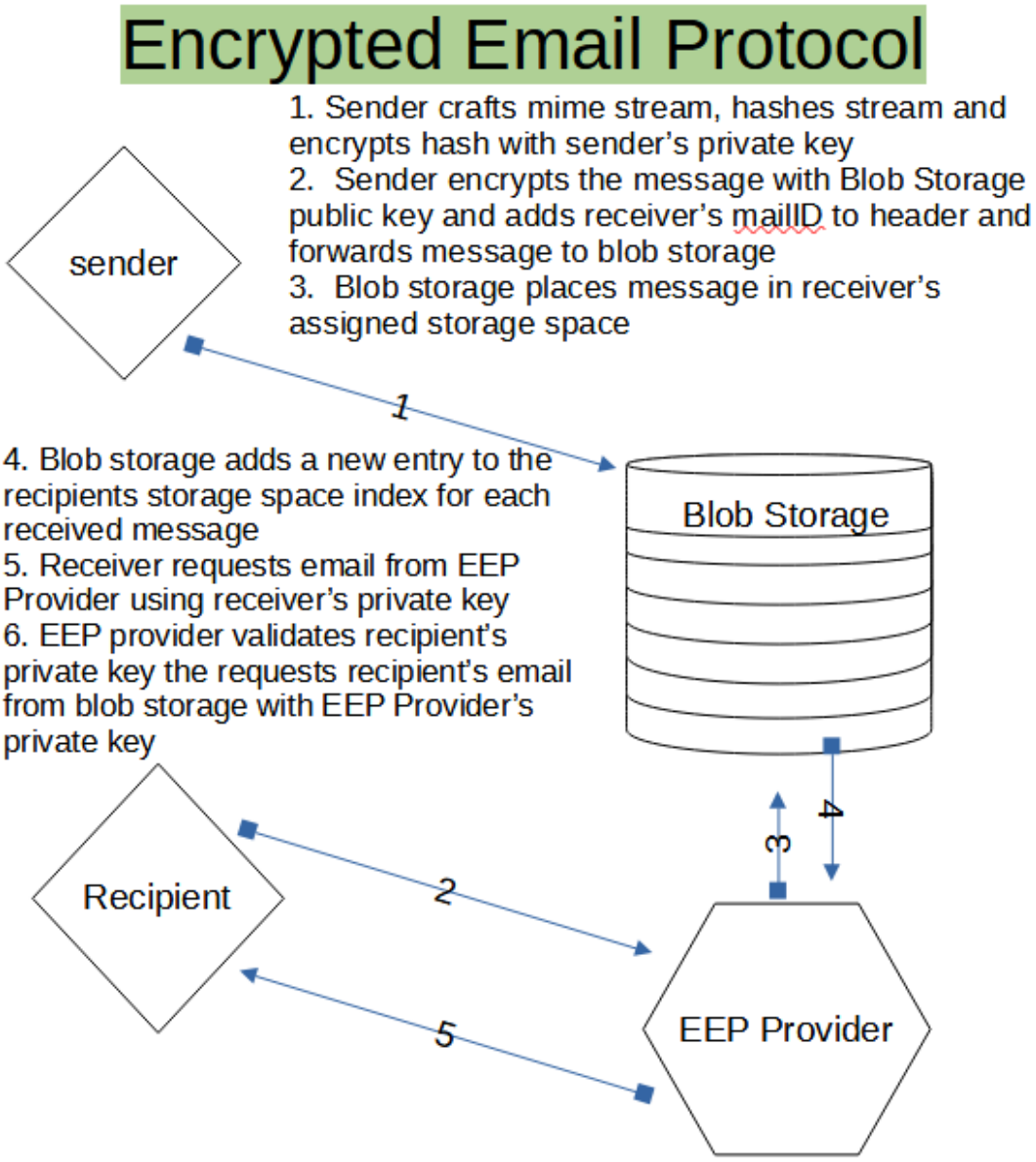

Email Encryption Protocol

Written by: Andrew Kosakowski

Table of Contents

Document Introduction

As of the time of publishing this document Email Encryption Protocol (EEP) is a notional protocol that is being released for discussion and consideration for development. This protocol and related documentation is free for all uses, development, implementation, discussion, and/or dissemination as long as this document and this permission-set is attached to all derivative, non-derivative, and/or future works, discussions, development, solutions, implementations, dissemination, and/or all other uses. This document is a technical introduction into EEP. The flow is designed to provide a provide an introductory overview of EEP to permit a base layer of understanding that can be leveraged throughout the remainder of the document. Assumption are then provided to provide insights into what was considered to be within the responsibility of EEP and what is the responsibility of other solutions and implementations as well as other relevant assumption. The EEP provider is covered in detail as the first part of the technical portion of this EEP introduction as it is the anchor of the EEP protocol much like a DNS server is the anchor in the DNS protocol. Technical sender-side, storage-side, and receiver-side EEP integration follow to provide insights about how clients and various storage types can leverage EEP based on intended best practices; however, the integration will not be exhaustive as this protocol will require tailoring to specific operating systems, solution providers, and software versions. Once the technical details of EEP are understood this document continues to ways to automate portions of EEP management and scaling capabilities. Once some automation and scaling of EEP certificate management have been provided a wrap-up summary will be provided. Overview of EEP.

Overview of EEP

Email Encryption Protocol (EEP) is a protocol that has been designed to increase non-repudiation, authenticity, and confidentiality of email in a manner that can scale easily and is simplistic. This has been accomplished by initially building off Better Encryption Protocol (BEP) then adapting the architecture to provide the specific needs of email users. When an identity (user or service) wishes to send an email the identity hashes the message and encrypts the hash with the identities private key. The email is then completely encrypted with the EEP Provider's public key and a header is attached that contains the mailID of all recipients. The message is then forwarded to the blob storage that provides email storage. The blob storage reads the email header to identify the recipients then places the email file in the storage locations assigned to each recipient and adds an entry into the storage location index that includes the email file location, some metadata, and whether the email had been retrieved by the recipient or is still pending retrieval. To retrieve any new messages the recipient sends a request to the EEP Provider whom forwards the request to blob storage. Blob storage checks the storage location index for the recipient and forwards all messages that are waiting to the EEP Provider. The EEP Provider decrypts the messages using the EEP Provider's private key then encrypts them with the recipients public key. When the recipient receives the messages they are decrypted using the recipient's private key, then the message signatures are validated using the sender's public key, and finally added to the recipients mailbox.

Assumptions

The assumption made when creating this document and EEP are as follows:

- The integrity of data in storage is not the responsibility of EEP and will be ensured using other mechanisms
- The blob storage provider will only permit read actions from the EEP Provider

- The blob storage provider will prevent the EEP Provider from any actions and permissions other than read permissions
- The blob storage provider will perform authentication and authorization to ensure only permitted entities are permitted to write data
- EEP is not used for DDOS, Brute Force, Certificate, or any other network or end-user identity attacks
- Knowing the mailID or the salt will not affect the confidentiality of data in storage or the authenticity of messages

EEP Provider

The EEP Provider is a security device that acts as the sole identity which holds the decryption keys for all data in blob storage and is also the only identity that has authorization to read data from blob storage. Thus, the EEP Provider acts as a read-only proxy and certificate manager for all email identities in a way that ensures confidentiality, authenticity, and non-repudiation. Many of the responsibilities of the EEP Provider are taken directly from BEP (Better Encryption Protocol) the BEP Provider responsibilities. The EEP responsibilities include EEP certificate management, mailbox layer encryption, acting as a read-access proxy, and recipient identity authentication and authorization.

EEP Provider certificate management

The EEP Provider is responsible for managing the email storage private key as well as the creation, dissemination, verification, and revocation of client identity certificates. For the remainder of this document all sender and receiver identities will be referred to as clients or client identities. Upon EEP Provider initialization a recovery storage certificate can be imported or a self-signed root certificate can be created. If the EEP Provider creates a self-signed root certificate upon initialization, that root certificate is used to create all other certificates and should be placed in the trusted root store of all clients. Once the root certificate is trusted the initial storage key pair is created with the public key being made available to all clients. When the initial storage certificate is created the private key is available for a one-time export to a physically attached (not a network attached) storage device. This physical copy of the private key can be used for back-up and recovery purposes or for horizontal scaling. Once the storage certificates are created and the public key is made available to clients client certificates are created. Every client identity that will be used to send or receive email will receive a unique certificate based key pair. Client certificates should contain two EEP unique fields. The first unique client field is a binary field that indicates if a client identity should be permitted to retrieve and send email or just send email. The second EEP specific field is a reference field that states whom the identity is permitted to impersonate (spoof) in the cases of services that need to send authenticated emails on behalf of other identities. An example of this type of impersonation is having a learning management system and application tracking system both send authenticated emails on behalf of the general HR email.

Rotating EEP storage keys

The rotation of the EEP Provider private key (the storage key) takes a very specific process and should be performed in the following order.

1. Use the root certificate to create a EEP Provider key pair (Do not activate the new certificate)
2. Save the new EEP Provider key pair and certificates to the mounted physical storage.
3. Create a new EEP Provider and import the new certificates.
4. Create a read/write “rotate” identity in the identity provider and a sync script or service.
5. Create certificates for all identities found in the identity provider.

6. Create a new blob storage provider.
7. Have the “rotate” identity read all stored data in the first blob storage and write it to the new blob storage. In this step the old key is used to decrypt the data and the new key is used to encrypt the data again. This will also preserve the double encryption for mailbox encryption.
8. Once 95% of data has been migrated to the new storage have users whom have mailbox level encryption download their mailboxes and sync them with the new storage solution to replace their former encryption certificates with their new ones.
9. Remove clients’ old certificates and block access to the old EEP Provider.
10. If certain users need access to the old storage account they can be individually be granted access to the old EEP Provider.
11. Once migration is complete delete the “rotate” identity and sync service.

EEP Provider Mailbox layer encryption

Mailbox layer encryption is possible by encrypting messages with a recipients public key prior to encrypting the message with the storage public key. In this way not even the EEP Provider will have access to the plain-text version of the message. When the recipient receives the message it double-decrypts the message with its private key. The mechanics of this will be further discussed in the Sender EEP Integration section. To rotate the mailbox layer encryption the client should use the “old” certificate for read actions for all email files that are dated prior to the issuance of the new certificate and the “new” certificate for write operations

EEP Provider read-access proxy

The EEP Provider is a read-access proxy for every client that wishes to retrieve email from the blob storage. The EEP Provider receives requests for email retrieval from clients that are encrypted using the clients’ private key. The EEP Provider decrypts the request using the clients’ public key (if available) then forwards the request to the blob storage provider using HTTPS. When the storage provider receives the request it locates the applicable files and returns them to the EEP Provider. When the EEP Provider receives the requested content over HTTPS the content is still encrypted with the EEP Provider public key (storage public key) and the EEP Provider used the private key to decrypt the content. The content is then encrypted again with the public key of the client and forwarded to the client. The following steps are taken when a client identity requests retrieval of email:

1. The EEP Provider receives an encrypted request from a client identity
2. The EEP Provider decrypts the request using the client identity’s public key
3. The EEP Provider forwards the request to blob storage
4. The EEP Provider receives encrypted messages from blob storage
5. The EEP Provider used the EEP Provider private key (storage private key) to decrypt the messages
6. The EEP Provider encrypts the messages with the client identity’s public key and forwards the messages to the client
7. The EEP Provider verifies successful receipt of the messages and attempts transmission up to 3 additional times
8. Upon successful transmission of the messages the EEP Provider send a verification message to blob storage

EEP Provider client authentication and authorization

The EEP Provider provides client authentication and authorization by verifying the legitimacy of client certificates that request read access (email retrieval) as well as certificate verification for all clients that wish to send email using an EEP client certificate. To determine which clients should be issued certificates the EEP Provider reads the user list of EEP Provider specific groups in the enterprise's identity provider database. The EEP Provider specific groups should be dynamic where possible and at a minimum must include one group to list all identities that require read (email retrieval) and write (email creation and sending) permission, one group for every shared email account, and one group for every email account that permits impersonation. The EEP Provider must have read-only access to these groups and should be configured to read the members of these groups every 24 hours by default with the ability for administrators to manually initiate a sync or to change the EEP Provider read frequency to fall between a frequency between 2 and 48 hours. If a user is removed from the group their certificate is removed from the EEP Provider instantly. If a user connects to the EEP Provider and there is no associated certificate an error message is presented to the client stating they need to be issued an email certificate. If a client's certificate is erroneously deleted adding the client back into the associated identity provider group will trigger the issuance of a client certificate and full access is restored.

EEP Client Integration

In EEP there are two types of client actions and the integration into EEP is based on the actions the clients are authorized to perform. Every EEP client is identified by the EEP Provider based on EEP specific groups in the enterprise identity provider. There are sender clients which include both traditional clients with full email accounts and mailboxes as well as impersonation clients which are only permitted to send email on behalf of other identities and which do not have mailboxes. There are also EEP recipients which includes standard identities and inbox-only accounts that have no authorization to send messages. Both client types require an authentic certificate must be in a trusted location such as a read only directory that is limited to only being read by the identity to which the certificate is issued (or the client's certificate store). A scheduled task can be activated when "opening" or "running" a client EEP email solution or a solution that has EEP integration to validate client certificates. When the email solution is opened the client activates a process that checks to see if the client certificate is present. If the certificate is present it requests certificate verification with the EEP Provider to authenticate the signature. This verification is encrypted using the certificate that is being validated. If the certificate is not present or not valid the client receives an error message stating "a valid EEP certificate can not be located, an administrator will need to provide a one-time pass-code." This pass-code will be used to connect to the EEP Provider over HTTPS to download the client certificate. Both sender and recipient email client integrations are discussed in the associated subsections that follow.

Sender EEP Integration

Any attempts to connect with the EEP Provider and download messages will result in an error message if the sender is not authorized to receive emails. After the certificate is validated the sending identity the client software performs the following steps:

1. The client software receives a "send email" signal.
2. The client software asks the EEP Provider if mailbox encryption is configured by asking the EEP Provider.
3. The client software encrypts the message with the recipients public key.
   This is an optional step only used for mailbox level encryption

4. The client software encrypts the entire message with the EEP Provider (storage) public key and adds a header that includes the email creation date and time, mailID, and message ID and adds a non-encrypted version to a temporary cache.
5. The client software verifies the validity of the EEP Provider public certificate with the EEP Provider.
    1) If the certificate is verified as valid the message is sent
    2) if the certificate is not verified the correct certificate is retrieved
    3) If the correct certificate cannot be retrieved (3 failed attempts) an error message is presented to the client identity stating “EEP Provider certificate cannot be retrieved”.
    4) If a new certificate is retrieved the client software proceeds to step 4 with the cached message
6. The client software sends the message and verifies delivery
7. The client software erases the cached message

Recipient EEP Integration

The recipient identity configures mailbox level encryption by sending an encrypted messages to the EEP Provider email that contains the associated configuration setting. The message is encrypted with the client identity’s private key to verify legitimacy of request. When a client wishes to retrieve messages the receiving identity the following steps are taken:

1. The client software encrypts a retrieval request with the client identity’s private key.
2. The client software forwards the request to the EEP Provider
3. The EEP Provider decrypts the message with the client identity’s public key
4. The client software receives the messages from the EEP Provider
5. The client software decrypts the messages using the client’s private key
6. The client software decrypts the message a second time (for mailbox level encryption) with the client’s private key and added to the client identity’s email folder

Blob Storage EEP Integration

The role of storage is limited to creating mailIDs, creating storage locations, creating storage location indices, and updating storage location indices. The mailID is an arbitrary value that associates a storage location with a client identity and can be the location of the storage space assigned to a mail identity. The MailID is used in headers to determine which location received messages should be placed while maintaining the confidentiality of sender, recipient, and content. Storage locations are created for every client that has read (receive email) authorization. Inside each storage location is an index that list each messageID received, the date and time received, and whether the message has previously been retrieved by the recipient. In the case of shared mailboxes the index has an added value that indicates the unique client count that have retrieved messages. Once the number of unique client count matches the number of group members the value that indicates a message has been retrieved is set to true. When messages are retrieved the indices are updated after the EEP Provider verifies successful deliver of the messages.

Automation and Scaling

Reducing management effort is key in any implementation and is becoming increasingly important as technology progresses beyond human manageability therefore this section attempts to list the ways EEP can utilize automation. The first place automation can be introduced into this protocol is in the addition and removal of members from the identity provider EEP specific groups using dynamic membership. The next place to look is EEP Provider scaling. While scaling does require access to a physically

mounted storage device that holds the private key, it is possible to load the private key on a physically attached storage device that is then connected to a physical security hardware that permits read access to a limited number of (configurable static) IPs in a defined subnet. In this scenario extreme precaution should be taken to protect the private EEP Provider key. The final place automation can be introduced is in the dissemination of client private EEP certificates. A script can determine when a new client identity is added to a group in the identity store, create a one-time pass-code, then place a text file with the one-time pass-code in a text file on a client identity's share drive or desktop.

Caveats

There are some caveats with EEP that should be understood and taken into consideration. The first caveat is the protection of client identity's private key. The protection of the client identity's private key is contingent on the client and not the responsibility of EEP. If the client identity's private key can be accessed by an unauthorized identity messages can be sent and received by unauthorized identities. If a user receives a hostile termination it will take the configured amount of time before that user loses access to email unless an administrator manually enters the EEP Provider and initiates a sync. If the EEP Provider private key is not exported or the exported key is lost and the EEP Provider gets corrupted all email will be unrecoverable. The EEP Provider admin will have access to the EEP Provider private key and will have the ability to decrypt all email in storage unless the emails have been encrypted with mailbox level encryption. When a user is removed from the identity provider's EEP specific groups all mailbox level encrypted messages will be unrecoverable unless the enterprise takes a back-up of client identities' private keys.

Summary

Email Encryption Protocol (EEP) is a protocol developed by the author to open a discussion about advancing email to a place that increases non-repudiation, confidentiality, and authenticity in a way that can be scaled easily and quickly. With EEP the ability to send messages and the ability to receive messages are integrated into one solution while ensuring a high level of authorization that even permits the authentication and approval of multiple identities being able to impersonate other identities in a secure manner. EEP is a replacement for legacy protocols that were built without security in mind but should be considered a starting place for discussion and may not represent a complete and independent solution.