CIPHERMAIL EMAIL ENCRYPTION

# Ciphermail Frequently Asked Questions


June 19, 2014, Rev: 8963

# Contents

# FAQ

# S/MIME

## What exactly is a certificate?

S/MIME uses Public Key Infrastructure (PKI) to securely exchange information over insecure networks using public key cryptography. Public key cryptography makes use of two keys: the *public key* and the *private key*. The private key should be kept secret whereas the public key can be made available to everyone. The public key is used for encrypting and the associated private key is used for decrypting. If a sender wants to send an encrypted message, the sender needs to get hold of the correct public key of the recipient.

An X.509 certificate contains the public key of the certificate owner. The sender needs to select the correct certificate for encryption. The certificate therefore contains some extra information which can be used to identify the correct certificate. A certificate used for S/MIME should at least contain an email address for which the certificate was issued. Most certificates however contain more information like name, company etc.

A certificate in essence is just a binary file with a specific format. With the correct tools, a certificate with any content can be easily made. A certificate can therefore not be automatically trusted because the content, i.e., the identifying information about the certificate owner, cannot be trusted.

There are two ways a sender can decide to trust the certificate: explicit trust or implicit trust using PKI. By explicitly trusting a certificate, the sender decides that this specific certificate is valid and trusted. For example the sender has checked the certificate thumbprint over the phone. The main disadvantage of explicitly trusting certificates is that every certificate must be manually checked and placed on the *Certificate Trust List* (CTL).

Using PKI, an automatic procedure can be used to implicitly check the trust of a certificate. S/MIME uses a hierarchical trust model where trust is inferred bottom-up. The root (the bottom) is blindly trusted (which makes it by definition a root) and all leaf nodes and branches (the end-user and intermediate certificates) are trusted because they are 'child's' of the trusted root (to be precise the intermediate certificates are issued by the root certificate). The main advantage of using the PKI trust model (i.e., implicit trust) is that only the root certificate has to be explicitly trusted. All certificates issued by the root are automatically trusted. A certificate is issued by a CA, which can be a root or an intermediate CA. Because the CA digitally signs a certificate, end-users can check whether the certificate was issued by a trusted root.

Besides identifying information about the owner of a certificate, most certificates contain more information like expiration date, certificate usage etc. For example if the extended key usage is set it should contain at least *anyKeyUsage* or *emailProtection* to make the certificate valid for S/MIME.

## What is a root certificate?

S/MIME uses a hierarchical trust model. A certificate can be validated by checking whether the certificate was digitally signed by the issuer of the cer-

tificate. The issuer certificate itself can be issued by another certificate (which makes it an intermediate certificate). This procedure can be repeated until a root certificate is reached. Because a root certificate is explicitly trusted and a valid chain can be built from a certificate to a root certificate, the certificate is implicitly trusted. Because a root is 'blindly' trusted only root certificates that are really trusted should be added to the root store.

## What is an intermediate certificate?

An intermediate certificate is a CA certificate used for issuing other certificates. Sometimes a CA uses multiple intermediate certificates and each intermediate certificate is used for different purposes. For example one intermediate certificate is used for issuing certificates only valid for S/MIME and another one for only SSL. Only certificates with a CA basic constraints extension are allowed to issue other certificates.

## What is an end-user certificate?

An end-user certificate is a certificate which should be used for encryption or signing. An end-user certificate for S/MIME should at least contain the email address of the owner. An end-user certificate should not be used for issuing other certificates (i.e., end user certificates should not have a CA basic constraint extension).

## How can a certificate have 'child' certificates? Do 'child' certificates share the same public key as the parent? What about the private key?

A root and intermediate certificate are used for issuing other certificates. The certificates issued by a root and intermediate certificate are 'child' certificates of the root and intermediate.

Every certificate has it's own public key. Public and private keys are not shared. A private key is not part of a certificate. When a certificate request is created, a private key and public key is generated. The public key is added to the certificate. Although the certificate and private key are associated, they are separate entities. For transport, a certificate and private key are often stored in a password protected *.pfx* file.

## Why do certificates expire? How does Ciphermail react when it is called upon to do encryption with a recently expired certificate?

There are different reasons why certificates have an expiration date. For example when a root certificate is created the protection level of the certificate (for example the length of the public key) is set to a level that should be strong enough to last the lifetime of the certificate. Because of increasing computer power, the level of protection should also be increased over time. By making sure a certificate expires before the level of protection falls below an acceptable

level, the certificate protection level is always strong enough. Another reason why an end-user certificate has an expiration date is that when a certificate should no longer be used, because for example a employee has left, the certificate automatically expires after some time.

Commercial certificate issuers have a business reason why certificates expire (most of them are only valid for one year). They ensure themselves of a constant income when they only issue certificates that expire within one year.

Ciphermail does not automatically use an expired certificate. If an expired certificate should be used by Ciphermail because the administrator has determined that the certificate can still be used, the certificate should be "whitelisted" by adding the certificate to the CTL. Because issuing new certificates is cumbersome, certificates should not expire too soon. We therefore advise to make end-user certificates valid for 5 years.

## What is the difference between a signature and an encryption certificate? Do they have to be different? Should we always sign outbound encrypted emails?

A certificate can contain a *key usage* extension which restricts the certificate usage. For S/MIME encryption, if a key usage is specified it should at least contain *keyEncipherment*. For S/MIME signing, if a key usage is specified it should at least contain *digitalSignature* or *nonRepudiation*. Using a different certificate for encryption and signing, signing of messages can be done on the senders email client and decryption can be done on the encryption gateway. Sometimes policies require signing of messages with a smartcard on the desktop because the signature is considered a legally binding signature. If signing of messages on the senders email client is not required there is no need to use a separate signing and encryption certificate.

Signing a message provides you with authentication and message integrity. The receiver can check who wrote the message (authentication) and whether the message was modified after signing (integrity).

## Should we have a self-signed root certificate or one signed by a certificate authority (CA)? If the latter, then which CA should we use and what should we ask for exactly? What due diligence will the CA perform? How much will it cost?

The problem with using your own root certificate is that the root certificate is not automatically trusted by external users. Every external user should import the root certificate into the root certificate store. Getting an intermediate CA certificate issued by a universally trusted root has the advantage that all issued certificates are automatically trusted. The main disadvantage however is that this is expensive. Most CAs do not provide the private key, they only allow you to issue certificates via an externally accessible portal. Some widely known commercial certificate issuers are: Verisign, Comodo, GlobalSign and StartSSL. The costs of having your own CA issued by a trusted root depends on a lot of details like volume etc.

## What does it mean when a certificate is revoked? Who is able to revoke a certificate? Why is it done? Are we supposed to download a revoked list from our certificate authority?

Sometimes a certificate should no longer be used. For example when the *private key* has been compromised or an employee has left the company. Certificates can be revoked by putting the certificates on a Certificate Revocation List (CRL). A CRL is issued and signed by the certificate authority (CA) that issued the certificate. A CRL is periodically updated. Most certificates contain a "CRL distribution point" with the URL from which the CRL can be downloaded. Ciphermail periodically downloads all the CRLs from all the CRL distribution points of all certificates in the certificate store. Newly downloaded CRLs replace the previously downloaded CRLs. CRLs can also be manually added to the CRL store. However, this is only needed when a CA issues a CRL but the certificate does not contain a CRL distribution point.

## What is a certificate Trust List (CTL)?

A Certificate Trust List (CTL) is a list of "white-listed" and "black-listed" certificates. A CTL is created and updated by the gateway administrator and can contain certificates from different issuers. In most setups, trust management using PKI will be sufficient. Sometimes however, the administrator needs more control over this automatic process. Some examples of when a CTL can be helpful are:

1. A certificate should no longer be used because it was compromised. The certificate issuer however does not publish a CRL. By *black-listing* the certificate, the certificate will no longer be valid.

2. A certificate is not valid because the root is missing. The administrator however knows that the certificate is valid (for example the thumbprint has been checked over the phone). By *white-listing* the certificate, the certificate will be valid for encryption.

3. A certificate is not valid because the certificate has expired. However, the administrator is 100% certain that the certificate is still 'valid'. By *white-listing* the certificate, the certificate will be valid for encryption.

Using the CTL, trust can be managed on a more ad hoc bases. This is more-or-less similar to how trust is managed with PGP. Instead of 'inheriting' trust from other trusted certificates, with a non-PKI approach using the CTL, every certificate should be explicitly trusted.

## Should I use SHA-1 or SHA-256 for CRLs and certificates?

SHA-1 and SHA-256 are hash algorithms used for digital signing. SHA-256 is more secure than SHA-1 and should therefore be preferred over SHA-1. However, older versions of Windows do not support SHA-256. SHA-256 is supported on Windows XP SP3 (service pack 3) and newer versions of Windows. If older versions of Windows should be supported SHA-1 should be used for certificates and CRLs.

## by browsing the database certificates i found that some external recipients have split certificates for "digitalSignature" and "keyEncipherment", some have additionally "dataEncipherment" and some have no Usage Key at all. After reading about this Flags the question arise how Ciphermail handles this Flag if at all

When deciding whether a certificate is valid for S/MIME encryption, Ciphermail checks the following certificate properties: *key Usage* and *Extended Key Usage*.

If the key usage extension is specified, it should contain “keyEncipherment”. If the key usage extension is not specified, the certificate is considered to be valid for all usages. If the extended key usage extension is available, it should either contain “anyKeyUsage” OR “emailProtection”. If the extended key usage extension is not available, the certificate is considered to be valid for all extended usages.

When deciding whether a certificate is valid for S/MIME signing, Ciphermail checks the following certificate properties: *key Usage* and *Extended Key Usage*.

If the key usage extension is available, it should contain “digitalSignature” OR “nonRepudiation”. If the key usage extension is not specified, the certificate is considered to be valid for all usages. If the extended key usage extension is available, it should either contain “anyKeyUsage” or “emailProtection”. If the extended key usage extension is not available, the certificate is considered to be valid for all extended usages.

The “dataEncipherment” key usage is not used very often. It’s used when the private key should be used to encrypt other data than a session key. S/MIME is a two step process. When a message is encrypted, a session key is generated (for example a AES key). The message is encrypted with the session key (for example AES encryption). The session key is then encrypted with the public key. The “keyEncipherment” extension tells that the key can be used to encrypt a session key.

## I found some certificate in our Ciphermail store with key usage “nonRepudiation”. I downloaded the matching root CA but this certificate is still marked as invalid so the question is if this is because of the exclusive use of “nonRepudiation” and what this certificate should be used for anyway?

Non-repudiation is a “strong” form of signing which is normally used for legal electronic signatures. This normally implies that the private key is stored on an approved smart card and that the certificate is issued by some highly trusted issuer. Sometimes, three certificates (and private keys) are issued to one person. An encryption certificate, a signing certificate and a non-repudiation certificate. With three certificates, the signing certificate is typically used only for authentication purposes and the non-repudiation for signing documents.

Ciphermail does not make a distinction between a signing certificate and a non-repudiation certificate. A certificate with signing and/or non-repudiation

key usage is acceptable for signing. The reason why the certificate is invalid in your case is that the certificate only contains the non-repudiation key usage. The certificate is therefore not valid for encryption. It should however be valid for signing. This however requires that you possess the private key.

# OpenPGP

## Is the web of trust supported?

Currently only self signed signatures are supported. For example a User ID is only accepted if it is self signed and if the signature is not revoked or expired. Sub keys must also have a valid self signed sub key binding signature (if the sub key is also a signing sub key, there must also be a valid primary key binding signature). Upcoming version of Ciphermail will probably support web of trust (i.e., support for signatures issued by other entities).

## Is a key trusted by default?

A key is not trusted by default. A key which is not trusted is not used for signing or encryption. A key can be trusted by opening the key details page and then selecting *key trust*. By default all sub keys will be trusted as well (unless the *Include sub keys* is not selected).

## Which keys are used for encryption?

Keys are suitable for encryption if the keys are valid (i.e., trusted, not revoked and not expired etc.) and are valid for encryption and if the email address of the recipient matches the assocoated email addresses or domains of the key. A message is encrypted with all suitable encryption keys, in other words, if there is more then one encryption key which is suitable for encryption, the email will be encrypted with all the suitable keys.

## Why is Incoming PGP/INLINE not enabled by default?

With PGP/INLINE, every individual part of the message (attachments and message bodies) is individually protected (signed and/or encrypted). To determine whether or not a message is PGP/INLINE protected, the complete message must be scanned. A PGP/INLINE message does not contain a specific header which can be used to determine whether the message is PGP/INLINE protected. Scanning every incoming email completely from top to bottom can be resource intensive, especially for very large attachments. It is therefore advised to leave *Incoming PGP/INLINE enabled* unchecked (i.e., disabled) unless PGP/INLINE support for incoming email is a requirement.

## What is *Auto update email addresses* and why should you want to disable this?

If *Auto update email addresses* is selected, all the email addresses found in a valid User ID of a PGP key will be automatically associated with the key. Only User IDs with a valid self signed signature will be used. If *Auto update email addresses* is not selected, email addresses should be manually associated with the key. This is a global only option. The reason you might want to disable this option, i.e., not automatically associates the key with the email addresses from the User IDs, is that a User ID is not validated. In principle the owner of a

key add any email address even if he or she does not own the email address. By disabling *Auto update email addresses*, the admin should manually validate whether the email address from the User ID is valid and should then manually associate the email address.

## What happens if I click *refresh public keys*?

When *refresh public keys* is clicked and keys are selected, those key are fetched from the registered key server(s) and imported. The fetched keys are merged with the existing key. So for example if there are new User IDs, those User IDs will be added to the existing key. Only new signature, User IDs etc are added to the existing key, nothing is removed. For example if the existing local key is revoked but the key on the key server is not revoked, the local key will still be revoked.

# PDF encryption

## Ciphermail supports three different encryption methods: S/MIME, OpenPGP and PDF. S/MIME and OpenPGP sounds very logical, but PDF is a strange method. Why did you add this?

Although S/MIME encryption is one of the most secure ways to encrypt email, the problem with S/MIME (or OpenPGP for that matter) is that it requires the recipient to use an S/MIME capable email client[1] and the recipient must have a certificate and a private key. Although installing a certificate and a private key is not hard, even less so when using the gateways built-in CA functionality, it may still be too cumbersome for some recipients. Especially when only a few secure email messages need to be exchanged over a longer period.

As an alternative to S/MIME encryption, PDF encryption can be used. The PDF standard allows a PDF to be encrypted with a password[2]. Files can be added to the PDF and are encrypted as well. Because most recipients already have a PDF reader installed, they do not need to install or configure any software.

When the gateway PDF encrypts a message, it converts the complete email message, including all attachments, to a PDF. The PDF is then password encrypted and attached to a new message (which is based on a template). This message does not contain any information other than a general note that the message contains an encrypted PDF.

## With PDF encryption, are attachments encrypted as well?

All attachments are added to the PDF document before the PDF is encrypted. The attachments are therefore encrypted as well.

## Is PDF encryption safe? Some companies claim they can crack PDF passwords?

PDF encryption is safe just as long as the password is long enough and the PDF is encrypted with AES128. PDF password crackers try to guess the password using different techniques. For example, some of the password crackers use a large list of common words and names. Ciphermail can generate a random password for every PDF. The default password length is 16 bytes (128 bits) of random data which is practically uncrackable.

## With PDF encryption, how can the recipient reply encrypted?

The encrypted PDF document contains a reply link (only if the gateway enables the reply functionality). When the recipient clicks on the reply link, the recipients web browser opens an online page (running on a Ciphermail gateway) on which the recipient can write the reply message.

[1] Most email clients however support S/MIME out of the box

[2] The PDF is encrypted with AES128 with a key based on the password.

## I want to use the PDF reply function, but no generated PDF contains the "reply" link. What do I have to configure, to enable this feature?

The PDF reply link is only added to the PDF when all of the following prerequisites are true: *a*) the server secret is set; *b*) “Reply allowed” is enabled; *c*) The “Reply URL” is specified and, *d*) The “Reply sender” is set.

## There are different password modes for PDF encryption. Which mode is most secure?

There are different modes to password encrypt the PDF:

1. The PDF can be encrypted using a pre-defined static password.
2. The PDF can be encrypted using randomly generated password. The password will be sent by SMS Text to the recipient.
3. The PDF can be encrypted using randomly generated password. The password will be sent back by email to the sender of the message.
4. The PDF can be encrypted using a One Time Password (OTP) algorithm.

For all password modes, it’s important that the password with which the PDF is encrypted is long enough to withstand a “brute force” attack[3]. Which password mode is most secure is not easy to answer since this depends on a lot of factors. For example, the static password mode can be very secure if the password is long enough. However, with the static password mode, every PDF sent to one recipient will always be encrypted with the same password. If an attacker somehow knows what the password is, all PDFs can be read. The pros and cons of each mode will now be briefly discussed.

### Static password

#### Pros

- Easy to setup. The recipient only requires one password.

#### Cons

- The password has to be securely exchanged with the recipient.
- All PDFs are encrypted with the same password.
- Long passwords required to protect against brute force attacks.

[3] With a brute force attack, the attacker tries to guess the password by trying all possible passwords.

#### Random password via SMS

**Pros**

- The password is sent via a different channel (SMS) than email. An attacker needs access to the email and SMS to read the email.
- Every PDF can be encrypted with a different randomly generated password.

**Cons**

- The recipients needs a telephone number to which the SMS can be delivered.
- The sender has to provide the SMS telephone number.
- Reading the password from the SMS can be cumbersome, especially with long passwords.
- If the password for the PDF is lost, the recipient can no longer open the PDF.
- SMS messages are not free.

#### Random password, sent back to sender

**Pros**

- The password can be sent via a different communication channel.
- Every PDF can be encrypted with a different randomly generated password.

**Cons**

- The sender somehow needs to exchange the password with the recipient using a secure channel.
- Typing the password into the PDF password dialog can be cumbersome, especially with long passwords.
- If the password for the PDF is lost, the recipient can no longer open the PDF.

#### One Time Password (OTP) using the online portal

**Pros**

- Every PDF is encrypted with a different randomly generated password.
- The generated password can be copied and paste into the password dialog. The password can therefore be very long.

- Because the recipient has to log into the portal to generate the password, online security against brute force attacks can be used.
- Since the password is generated using a server stored client secret, the PDF password can always be generated.
- The recipient has to log into the portal to generate the password. The account can be disabled if required.
- Using the “invite” mechanism, setting up secure email communication is very easy.

**Cons**

- The recipient has to login to generate the password.
- If the recipient forgets the portal password, the recipient cannot login and the password has to be reset.
- The account for the recipient has to be pre-configured or the recipient has to be “invited”.
- The gateway portal functionality must be accessible to external users.

## With the One Time Password (OTP) mod, a recipient can be invited. Is this not insecure? What happens if the invite is intercepted?

The hardest part in setting up a secure channel is the exchange of the first secret (and this holds for every product). To do this in a secure way, a different channel should be used. With the one time password mode, you can choose to pre-configure the portal password. The portal password should then be securely transported to the recipient using a different channel (i.e., not using email).

From a practical perspective, it is however not always easy to exchange the password via a different channel or, pre-configuring passwords for every recipient might be too cumbersome. For cases where pre-configuring the portal password is not feasible, the auto invite mode was introduced.

When the auto invite option is enabled and there is not yet a portal password for the recipient, an invite link will be added to the email. After clicking the invite link, the recipient can choose a portal password for the portal account.

If however, an attacker somehow manages to intercept the email containing the invite link, and clicks on the link before the real recipient does, the attacker can choose a password before the real recipient can. Using the new password, the attacker can generate the one time password and read the first encrypted PDF.

While this in theory (and practice) is possible, the real recipient however will be able to detect that this has happened because the real recipient can no longer log into the account since the real recipient does not know which password the attacker has selected. If the attacker clicks on the link **after** the

real recipient has selected a password, the attacker cannot log into the account since the real recipient already selected a password.

The window of opportunity for the attacker is therefore limited: If the real recipient has selected a password, the link in the invite email is no longer usable, if the attacker successfully intercepted the email and selected a password, the real recipient will immediately notice this the first time the recipient tries to login.

If you need to be absolutely sure that the password is set by the real recipient before sending sensitive information, you can send an email containing a keyword. To validate whether the password was selected by the recipient and not by the attacker, you should call the recipient and ask for the keyword.

# Data Leak Prevention (DLP)

## I would like to quarantine an outgoing email when the To and CC header contains a large number of recipients. How can I do this?

Import the email address pattern (see http://www.ciphermail.com/patterns.html), make sure that the action is set to *Quarantine*. The number of email addresses at which the email will be quarantined is determined by the threshold.

## Can the DLP module detect all information leakage?

No. The DLP module protects against accidental data leakage but it does not protect you against a knowledgeable attacker. Any DLP vendor that claims it can detect all information leakage is not telling the complete story. Information can be rewritten/recoded in such a way that it becomes almost impossible to detect. For example the word "secret" can be written in ASCII art like:

```
 ####  ######  ####  #####  ###### #####
#      #      #    # #    # #        #
 ####  #####  #      #    # #####    #
     # #      #      #####  #        #
#    # #      #    # #   #  #        #
 ####  ######  ####  #    # ######   #
```

If all outgoing information should be screened, the best thing to do is to quarantine all outgoing email and let the DLP managers approve/disapprove all outgoing email. DLP rules can be specified for a user, domain or for the global settings. This allows you for example to setup a rule to only quarantine outgoing email sent by certain users.

## I have added a sentence to the list of patterns but somehow the sentence is not matched. Why is the sentence not matched?

It could be that a word from you sentence belongs to the list of words that are skipped (the "skip list"). All words from the skip list are removed from the text before scanning. For example, assume that the skip list contains the word, "this" and that the pattern is "this is a text". Because the pattern contains a word from the list of words to skip, the pattern will never match.

## Why is there is skip list?

The main reason certain words are skipped, is that scanning time is improved since words that are very common but that do not contain any sensitive information are removed before scanning. The default list of words to skip contains the 100 most common words in English.

## I would like to match a word if the word contains uppercase characters but not when it contains lowercase characters. Is this possible?

No this is not possible. When text is extracted from the email, the text is normalized using the following procedure:

1. All carriage returns and line feeds are replaced with spaces.
2. Consecutive spaces are trimmed to one space.
3. All characters are converted to lowercase.
4. The resulting text is Unicode normalized (NFC).

The reason why all text is normalized is that patterns are simpler and therefore faster. Because all text is converted to lowercase, you cannot match uppercase characters.

## If my pattern contains uppercase letters, it never matches any text. Why is that?

For a more detailed answer, see question ***I would like to match a word only if the word contains uppercase letters but not when the word contains lowercase letters. Is this possible?***. The short answer is that all text is converted to lowercase. You should therefore make sure that your pattern only contains lowercase characters.

## Are there any pre-defined patterns?

You can download some patterns from http://www.ciphermail.com/patterns.html. If you have patterns that can be valuable for others and would like to share them, please contact us.

## Are attachments also scanned?

Text based attachments, for example .xml and .html, are scanned. Binary attachments are currently not scanned. Support for binary attachments, like .doc, .zip, .pdf etc. will be added to future versions of Ciphermail.

## I cannot delete certain patterns. Why is that?

If a pattern is used, for example selected by the global settings, the pattern is "in use" and cannot be deleted. Before a pattern can be deleted, make sure that the pattern is no longer selected by a user, a domain or by the global settings. To get an overview of which setting is using the pattern, click the pattern and select *view usage*.

## Are headers scanned as well?

Yes headers are also scanned. However, the following headers are skipped: *a*) Received; *b*) From; *c*) Reply-To; *d*) References, and, *e*) Message-ID. The reason these headers are skipped is to make it less likely to get false positives when scanning for multiple email addresses. If you want all headers to be scanned or add headers to be skipped for scanning, add or remove headers to dlp.xml.

# Gateway

## What are the default login credentials?

**For the Web Admin:**

| | |
|---|---|
| user: | admin |
| password: | admin |

**For the Virtual Appliance console and SSH:**

| | |
|---|---|
| user: | sa |
| password: | sa |

## I forgot the GUI admin password. How can I reset the password?

The admin GUI password can only be reset if you have local access to the machine (i.e., either via SSH or directly on the console).

You can reset the admin password to the default with the following SQL command:

```
UPDATE admin SET password='admin', passwordencoding='0', salt='' where
username='admin';
```

You need to issue this command with psql command using the following steps:

1. Login to the command shell of server where Ciphermail runs
2. login to the database server with psql

**login to gateway** You can either login with SSH or login directly on the console.

**login to the database server** On the command line, issue the following to login to the database server:

```
$ psql -h 127.0.0.1 -U djigzo djigzo
```

**Note: The default database password is 'djigzo' (without the quotes).**

**issue the following SQL statement to reset the password:**

```
UPDATE admin SET password='admin', passwordencoding='0', salt='' where
username='admin';
```

**Note: the SQL was successfully executed if the output says: "UPDATE 1"**

You should now be able to login with the default GUI credentials (see previous item)

**Note: the default password is not hashed. Any new password set with the GUI, will be hashed and 'salted'.**

## Ciphermail comes with no certificates installed. Why is that?

Which certificate authorities should be trusted by the gateway is something the administrator should decide. Ciphermail therefore does not come pre-configured with any root certificates. From Ciphermail's website two *.p7b* files can be downloaded: one with a collection of root authorities and one with a collection of intermediate authorities. These are just a collection of some of the mostly used CAs.

## What is the root password of the Virtual appliance?

The Virtual Appliance uses Ubuntu Linux LTS (Long Term Support). By default, Ubuntu does not allow the root to login. *sudo* should be used for all commands requiring root access.

## Incoming encrypted email is not decrypted?

Make sure that domains for which you receive email have been marked as internal domains. If not, Ciphermail tries to encrypt the message because it thinks the messages is sent to an external domain. Also make sure that a correct certificate with private key is available.

## The certificates from incoming digitally signed email are not stored in the certificates store?

Make sure that domains for which you receive email have been marked as internal domains. If not, Ciphermail does not extract the certificates from the digitally signed email.

## What is the difference between the Ciphermail engine and the Ciphermail web admin?

The Ciphermail engine is the back-end part that encrypts and decrypts the email, sends SMS Text messages, stores certificates and keys etc. Ciphermail web admin is the web application used to control the engine (the front-end part). The web admin uses SOAP calls to control the engine. It is therefore possible to install the engine and the web admin on separate servers.

## If I try to login immediately after starting Ciphermail I get an exception. Why is that?

The back- and front-end are started separately. Starting and initializing the two processes takes some time. If the front-end startup process was finished and the admin tries to login before the back-end startup process was finished, an error will be shown. This only happens during startup when the admin tries to login before the back-end process has finished initializing. The admin should try to re-login after a few seconds.

## Is it possible to store the administrator credentials and roles on an external LDAP?

See the Web LDAP authentication guide for instructions on how to authenticate against an LDAP server.

## Where should a Ciphermail gateway be placed?

The most typical setup is where a Ciphermail gateway is placed between the Internet and the internal server. If a gateway level virus scanner is used, Ciphermail should be placed between the Internet and the virus scanner (see figure 1). This ensures that the virus scanner never sees any encrypted email.

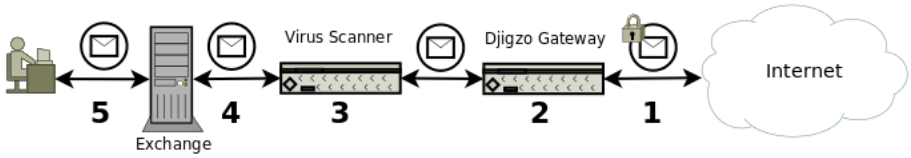


Figure 1: Ciphermail with Virus Scanner

Ciphermail can also be integrated with any existing content scanner (see Figure 2). With a content scanner, encryption can be forced based on message content (for example when the message contains a Social Security Number)

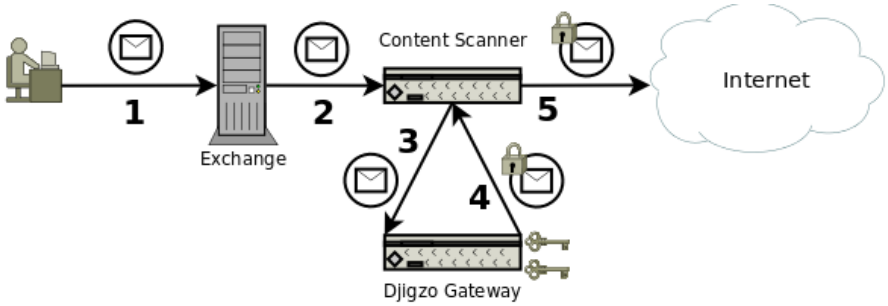


Figure 2: Ciphermail with Content Scanner

# Ciphermail for BlackBerry

## Which BlackBerry smartphones are supported?

All BlackBerry smartphones are supported including touch screen devices. The only requirement is BlackBerry OS $\geq$ 4.5 (contact us if support for lower OS versions is required).

## Why is a Ciphermail gateway required?

The BlackBerry infrastructure blocks access to certain attachments. When a message is S/MIME encrypted, the complete message including attachments is encrypted and the encrypted binary "blob" is then attached to a new message. The message with the S/MIME attachment then gets an S/MIME specific "Content-Type" to indicate that the message is an S/MIME message.

The problem is that the BlackBerry infrastructure blocks the S/MIME attachment. By rewriting the message headers (for example the "Content-Type" header) the S/MIME message will no longer be blocked by the BlackBerry infrastructure.

Strictly speaking a Ciphermail gateway is not required. Any email server which is capable of rewriting certain message headers can be used instead. For more background information on the required header rewriting see the *Ciphermail for BlackBerry Reference Guide*.

## Can Ciphermail for BlackBerry be used with BIS?

Yes Ciphermail for BlackBerry is BIS compatible.

## Can all email sent to my BlackBerry be encrypted?

Yes that's possible. If *Encrypt Mode* for the BIS email address is set to *Force* and a certificate is available for the BIS email address then all email sent to the BIS email address will be encrypted.

## Can Ciphermail for BlackBerry handle email which is S/MIME encrypted by Outlook?

Yes. Ciphermail for BlackBerry is S/MIME compatible and can therefore handle email encrypted by any S/MIME compliant email client (like Outlook, Lotus Notes etc.) as long as the private key for decryption is available on the BlackBerry smartphone. Email encrypted with Outlook (or any other S/MIME client) however, still requires that the headers are rewritten to make sure that the S/MIME attachment is not blocked. See *Why is a Ciphermail gateway required?* for more information.

## Should a copy of the private key be available on the Ciphermail gateway?

No not necessarily. If the gateway receives an email from an external sender and the email is already S/MIME encrypted, the gateway only needs to rewrite the headers. There is no need to decrypt the message. A private key therefore is not required on the gateway.

If all email forwarded to the BlackBerry smartphone should be encrypted, even when the original message was not encrypted, the public certificate of the BlackBerry smartphone should be available on the gateway. The gateway will use the public certificate to encrypt all email forwarded to the BlackBerry smartphone.

## Is Ciphermail for BlackBerry compatible with the BlackBerry smartcard reader?

Yes. Ciphermail for BlackBerry uses the BlackBerry's built-in cryptographic functionality. Private keys stored on smartcards can therefore be used as long as the smartcard is compatible with the BlackBerry smartcard reader.

## Are email attachments supported by Ciphermail for BlackBerry?

Yes. The complete message (body and attachments) is encrypted. Attachments for which a content handler is registered, can be directly opened (for example .doc and .xlt files will be opened with *Documents to Go*). Attachments can be saved on the device or on the SDCard.

## is HTML email supported by Ciphermail for BlackBerry?

Yes. HTML email will be shown with all mark-up (like color, images etc.). If the email contains an alternative text part, the user can switch between the HTML and text version.

## Does Ciphermail for BlackBerry validate signatures?

Yes. If the email is digitally signed, the signature will be validated using all the available intermediate and root certificates. If the signature is not correct (for example the issuer of the signing certificate is not trusted) a warning will be shown.

## Why is email sent with Ciphermail for BlackBerry relayed via a Ciphermail gateway?

Email sent with Ciphermail for BlackBerry is protected with S/MIME and sent to the configured relay email address (S/MIME tunnel). The Ciphermail gateway listens for email sent to the relay address and checks whether the message was signed with an approved certificate and whether the sender is allowed to relay

email. The email is then forwarded to the final recipient. The main advantage of relaying email via the Ciphermail gateway is that encryption management (certificates, PDF passwords, domain settings etc.) can all be done on the gateway.

## Can I prevent the user from sending non-protected email?

The *Clear mail policy* can be set to *Deny*. If the *Clear mail policy* is set to *Deny* and the user tries to compose a normal non-protected email, a warning will be shown and the compose window will be forcefully closed. If *Clear mail policy* is set to *Warn* only a warning will be shown. The user is still allowed to compose a non-protected email.

## I always need to enter the key store password when I open a message. Why is that?

The *Private key security level* of the decryption key determines when and how often the key store password should be entered. If the *Private key security level* is set to *Medium* or *High*, a password must be entered when the private key is accessed. With the *Low* security level, a password is not required when the private key is accessed. You can change the *Private key security level* of a private key by opening the BlackBerry certificate options (options→Security Options→Certificates), select the certificate for which the security level should be changed and then from the content menu select “Change Security Level”.

# Ciphermail for Android

## Why do I need an Android email client?

Ciphermail for Android does not provide functionality to retrieve email. An existing Android email application with attachment support, for example *Gmail* or *K9*, should be used to retrieve the encrypted attached *smime.p7m* message.

## Is Ciphermail for Android compatible with the Android Gmail App?

Yes. Ciphermail for Android is compatible with the Gmail App.

## I'm using K9, why are the subject, recipients and from headers missing?

The *smime.p7m* attachment, which contains the encrypted message body, does not contain the header information. Ciphermail therefore has to retrieve the header information from the original message.

## How are the private keys protected?

The private keys are protected with a randomly generated key. The key is encrypted with the key store password and the encrypted key is stored in the application preferences.

## Does Ciphermail for Android support smart cards or secure sd cards?

Not yet. Upcoming version will have support for smart cards, hardware security tokens and secure sd cards.

## Is there an optimized version for tablets?

Ciphermail for Android works on tablets but is not yet optimized for tablets. Upcoming versions will be optimized for tablets.

## I forgot the key store password. Can I reset the password?

The only way to set a new key store password is by clearing the key store (i.e., removing all private keys). The key store can be cleared by opening settings, Keys Store and then select "Clear key store" from the menu.

# General

## Suppose we issue a client a private key and the client loses it. What should we do?

The two main questions which should be answered are: *is the key required to decrypt old email?* and *is the key compromised?*

In most cases the client should still be able to decrypt previously received email. The client therefore needs a copy of the private key. If the client did not backup the previously received key (i.e., created a copy of the *.pfx* file) a new copy of the key can be securely sent to the recipient. Previously received encrypted email can be opened again after the key has been imported into the email client or operating system.

Whether or not the certificate should be revoked (i.e., placed on the CRL) depends on what is actually meant with '...the client loses it'. If for example the key was lost because the system had to be reinstalled because of a system crash then the key is technically not compromised. If however the key was stored on a laptop and the laptop was stolen, the key should be considered compromised. The certificate should be placed on the CRL and a new certificate and key should be generated for the client.

## Suppose we issue a client a private key and our relationship with that client ends. What should we do?

Technically speaking, nothing should be done. If the relationship was completely ended and no email should ever be sent again encrypted to the client, the client certificate can be revoked and the certificate can be deleted. By adding the certificate to the CRL, the client can no longer use the certificate. Whether or not a certificate should be revoked when the relationship ends is dictated by the companies policy.

The main disadvantage of revoking every certificate after a short period is that the CRL grows with every newly revoked certificate. A certificate should remain on the CRL at least until the certificate has expired. If client relationships are relatively short, it is better to issue certificates with a short validity interval (for example expire within 1 year instead of 5) instead of placing the certificate on the CRL after use.

## Suppose we issue a client a private key and that key is stolen from the client's computer. What's the worse that can happen? What should we do on our end if we suspect a client's key has been stolen?

S/MIME provides the following cryptographic security features: authentication, message integrity and non-repudiation. If a key is compromised (for example the laptop on which the key was stored was stolen) all of the three features no longer hold for messages encrypted and signed with that certificate. Email sent to the client which is intercepted, can be read by the person in possession of the private key (authentication). Anyone in possession of the private key can

sign a document pretending to be the rightful owner (message integrity and non-repudiation).

If a key is compromised, or suspected to be compromised, the certificate should be revoked by placing the certificate on the CRL. This ensures that the gateway will no longer use the certificate for encryption and that all email clients will report the certificate to be revoked.

Instead of issuing "soft keys" (i.e., a *.pfx* which should be installed on the clients computer) a more secure way of providing keys and certificates to clients would be to use a PKI enabled USB token. A key stored on a PKI enabled USB token is physically protected and cannot be copied. The main disadvantage of a USB token is that it is not as cost effective as sending a *.pfx* file.

## Suppose our Ciphermail server becomes compromised. What's the worse case scenario? How should we react?

If the Ciphermail server is completely compromised, all incoming and outgoing email sent through the gateway after the break-in could have been intercepted by the attacker. All keys should also be considered to be compromised and all issued certificates should therefore be revoked and everyone involved should be warned that the root certificate should no longer be used. Because all email could have been compromised, all involved parties should be notified of the security breach (whether or not this is required depends on the companies policies and state and federal regulations).

Ciphermail stores all private keys in a PostgreSQL database. This means that if the server is compromised all keys can be copied. To provide more security, a Hardware Security Module (HSM) can be used. All cryptographic functions involving private keys are handled by the HSM. Because all the private keys are stored on the HSM and all the private key handling is done inside the HSM all private keys are protected against copying. The main disadvantage of an HSM is that an HSM is expensive.

## If we get a root certificate signed by a CA, can we generate unlimited end-user certificates from that without any intermediates? Is this a sound practice?

A root certificate is normally not signed by another CA. A root certificate is a blindly trusted certificate which can issue other certificates. If your intermediate certificate is signed by a root certificate, it is best to issue another intermediate CA and securely store the original. The reason for this is that if you ever need to revoke the (second) intermediate certificate you can always issue another intermediate certificate. Sometimes however, certificate issuers do not allow intermediate certificates from issuing other intermediate certificates (the path length constraint of a certificate can specify a maximum length of a certificate chain).

### When selecting a password to protect private keys sent to clients, should a unique one be used each and every time?

It is more secure to always use a new password. If password protected private keys are sent to multiple clients all within the same company and the same password is used for all keys, in principle one recipient can open the password protected private key file of another recipient (access to the other recipients email box is still required though). If multiple keys are sent to just one recipient the same password can be used if required.

### What admin roles should customer service personnel be granted?

The required roles largely depends on the tasks done by the customer service personnel.

### When issuing end-user certificates, a CA email is supposed to be specified. Who should see the mail associated with this email address?

The password protected private key file is sent by email to the external client. The actual sender of this email is set to the CA sender. The reason this is required is that it allows the administrator to specify the security settings of the CA sender. Suppose that the preferences are set that all email sent by the gateway should be encrypted. If a key and certificate is generated and sent to an external client, the email containing the password protected private key file will also be encrypted. The recipient however cannot open the encrypted email because the private key required for opening the email is inside the encrypted email. The preferences of the CA sender should therefore be set to never encrypt. The sender of the email containing the password protected private key will be equal to the CA sender (i.e., the from header of the email will be set to the CA sender).

### Where can I get certificates?

Ciphermail has a built-in CA which can be used to issue certificates for internal and external users. A certificate and private key can be sent encrypted to an external user. The external user can use the certificate with any S/MIME capable email client. Some external commercial and non-commercial certificate providers are:

| | |
|---|---|
| StartSSL | www.startssl.com (free) |
| CACert | www.cacert.com (free) |
| Comodo | www.comodo.com (free for personal use) |
| Verisign | www.verisign.com |