DJIGZO EMAIL ENCRYPTION

# DJIGZO Gateway Administration Guide


May 29, 2012, Rev: 6854

**Acknowledgments:** Thanks goes out to Andreas Hödle for feedback.

# Contents

# 1 Introduction

DJIGZO email encryption server is an email gateway (MTA) that encrypts and decrypts your incoming and outgoing email. Because DJIGZO serves as a general SMTP email server, it is compatible with any existing email infrastructure and can easily be placed before or after existing email servers. DJIGZO is typically installed as a "store and forward" server. Email is therefore only temporarily stored until it is forwarded to it's final destination.

DJIGZO currently supports two encryption standards: S/MIME and PDF encryption. S/MIME provides authentication, message integrity and non-repudiation (using X.509 certificates) and protection against message interception (using encryption). S/MIME uses public key encryption (PKI) for encryption and signing. PDF encryption can be used as a light-weight alternative to S/MIME encryption. The PDF standard allows PDF documents to be password encrypted. PDF documents can also contain attachments embedded within the encrypted PDF.

# 2 Setup

This guide assumes that DJIGZO has already been installed. For installation instructions of DJIGZO see the *installation guide*. The administrator can login to the administration page by opening the following URL in a browser: `https://192.168.1.1` (change the IP address to to match the address of the gateway).

**Note:** If DJIGZO was manually installed (i.e., not using the Virtual Appliance) the URL should probably be `https://192.168.1.1:8443/djigzo`.

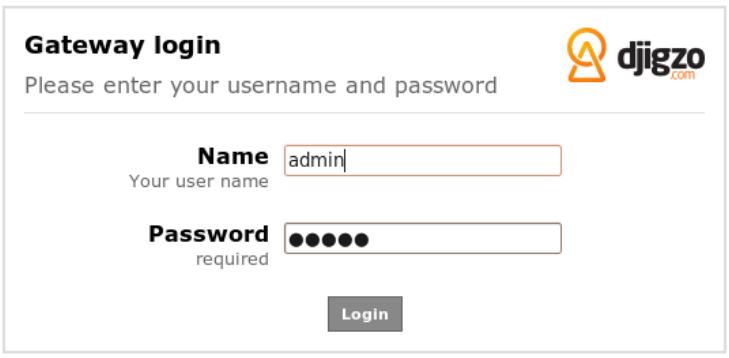


Figure 1: Login dialog

The login page should appear (See figure 1). After logging in with the correct credentials, the *users* page will be opened.

**Login credentials:** Use the following default credentials:

username: admin
password: admin

**Note:** it can take some time to login after a restart because the web application must be initialized upon first login.

# 3 MTA setup

DJIGZO uses Postfix for the *Mail Transfer Agent* (MTA) part of the gateway. The encryption and decryption is done by the *Mail Processing Agent* (MPA). DJIGZO contains an *MTA config* page that can be used to configure most of the relevant Postfix parameters.

The *MTA config* page can be opened from the Admin menu. The *MTA config* page (see figure 2) contains most of the relevant Postfix parameters for a "store and forward" email server. Postfix parameters which cannot be set with the *MTA config* page should be set with the *MTA raw config page* (or alternatively by directly editing the Postfix configuration files). We will briefly explain the relevant Postfix settings. For a more thorough explanation of all the Postfix settings see the Postfix documentation (http://www.postfix.org/documentation.html).

## 3.1 Main settings

**Relay domains** Relay domains are domains for which the gateway needs to receive email. These are the domains for which the internal users receive email. A "store and forward" server normally has one or more relay domains (unless DJIGZO is only used for sending email).

If *Match Subdomains* is selected, subdomains of the relay domains automatically match. If *Match Subdomains* is not selected, subdomains of the relay domains only match if the subdomains are explicitly added to the relay domains.

**My networks** Most email senders (users and other internal email servers) are not allowed to send email to domains not specified as a "relay domain". To allow outgoing email to be sent to external domains, the sender IP address should be "white listed". The *My networks* list contains all the networks that are allowed to send email to external domains. The networks must be specified in CIDR notation. **Example**: 192.168.1.1/32, 10.1.2.0/24.

**Warning:** be careful only to allow internal email servers from sending email to external recipients otherwise the gateway will be used for sending spam.

**My Hostname** This should be the fully qualified domain name of the email server and is used as the default value for many other configuration parameters. Example hostname: djigzo.example.com.

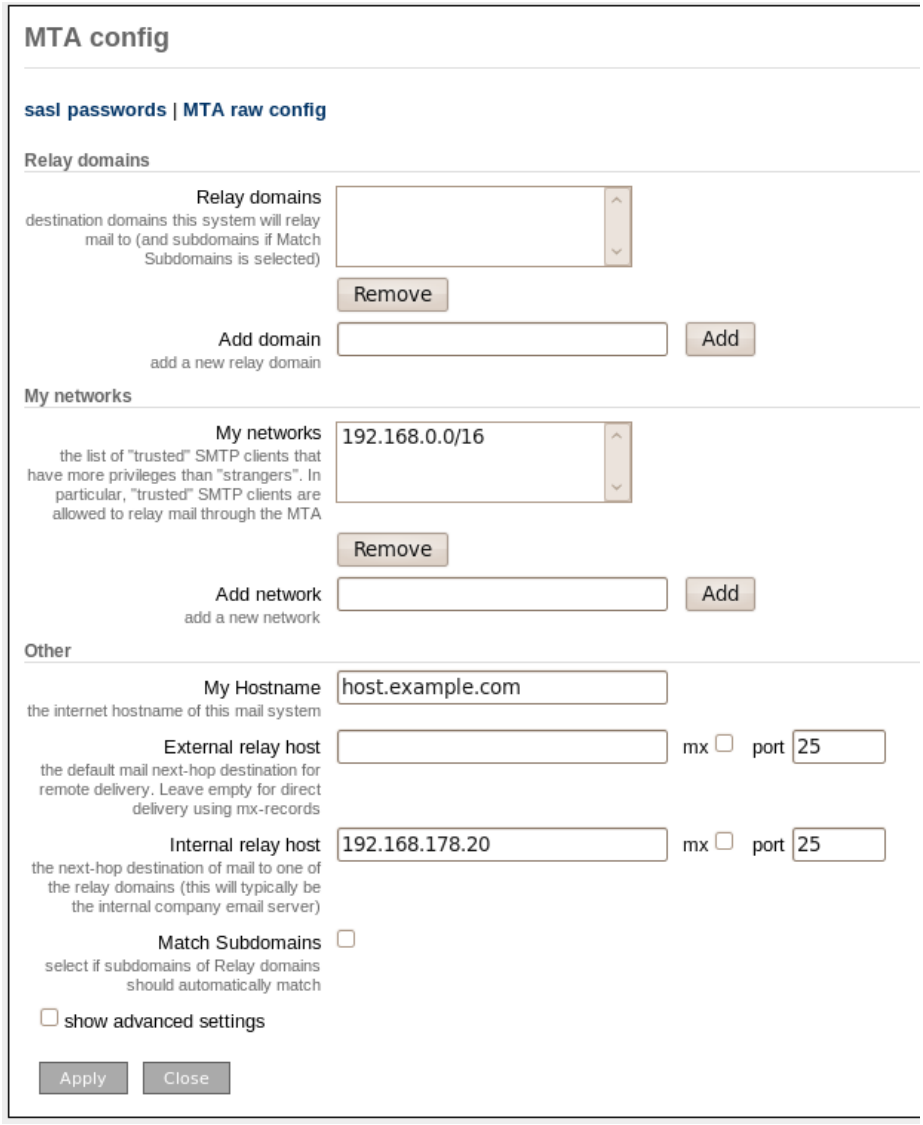


Figure 2: MTA config

*My Hostname* is used as the default domain for email messages sent with a missing domain name. *My Hostname* is also used for the default SMTP helo/ehlo name (see *SMTP helo name* setting below).

If the gateway directly delivers email to external recipients (i.e., not using an external relay host) it is important that the helo/ehlo name of the gateway is equal to the reverse lookup of the external IP address. If not, outgoing email can be flagged as spam. See Appendix A for more information.

**External relay host** The external relay host is used when email should be sent to an external domain (i.e., a domain which is not a relay domain). This can be the ISPs email server or some internal email server responsible for sending email to external domains.

If *External relay host* is not specified, email will be delivered using DNS MX-records. *External relay host* can be an IP address or a domain name. If the option *mx* is checked, the MX-records of the *External relay host* will be used instead of the A-record (this setting is only used when the *External relay host* is specified). The *port* setting is the port the *External relay host* server listens on (which in most cases should be the default SMTP port 25).

**Internal relay host** The internal relay host is used when email should be sent to an internal domain (i.e., sent to a relay domain). Typically this will be the companies internal email server hosting the users email boxes.

If *Internal relay host* is not specified, email will be delivered using DNS MX-records. *Internal relay host* can be an IP address or a domain name. If the option *mx* is checked, the MX-records of the *Internal relay host* will be used instead of the A-record (this setting is only used when the *Internal relay host* is specified). The *port* is the port the *Internal relay host* server listens on (which in most cases should be the default SMTP port 25).

**Match Subdomains** If *Match Subdomains* is selected, all subdomains of the *Relay domains* will also be relayed.

## 3.2 Advanced settings

The advanced settings can be set when the *advanced settings* checkbox is selected (see figure 3).

**Before filter message size limit** This is the maximum size of a message (in bytes) that the MTA accepts. A message that exceeds the maximum size is rejected by the MTA.

**Note:** Because of Base64 encoding, binary attachments (for example word documents) will be 4/3 times larger if sent by email. The maximum size limit, limits the total number of bytes including encoding. For example, if the limit is set to 10 MB, the total size of all the attachments cannot exceed 7.5 MB.

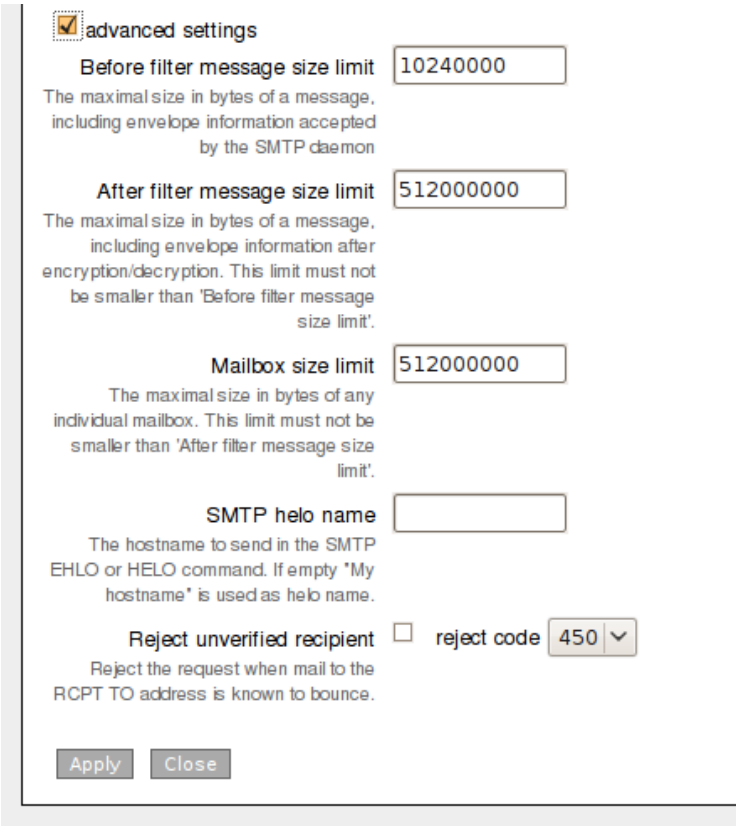


Figure 3: MTA advanced config

**After filter message size limit** The mail processing agent of the gateway is responsible for encryption and decryption of messages. The size of a message after encryption or decryption (or after signing) can be larger than the size of the message before encryption or decryption. The *after filter message size limit* should therefore be larger than the *before filter message size limit* otherwise the MTA will refuse to send the message after the MPA has handled the message. It is advised that the *after filter message size limit* should be at least 2 times larger than the *before filter message size limit*.

**Mailbox size limit** If mail is locally stored (only when *Local domains* are specified) the *Mailbox size limit* will be the maximum size (in bytes) of an individual mailbox. The *Mailbox size limit* should not be smaller than the *after filter message size limit*. This setting is only required when Postfix receives email for a local domain. By default the gateway does not enable the option to directly specify local domains.

**SMTP helo name** The *SMTP helo name* is the hostname used for the SMTP *EHLO* or *HELO* command. If *SMTP helo name* is not explicitly specified, *My Hostname* is used as the SMTP helo name.

**Note:** If the gateway directly delivers email to external recipients (i.e., not using an external relay host) it is important that the helo/ehlo name of the gateway is equal to the reverse lookup of the external IP address. If not, outgoing email can be flagged as spam. See Appendix A for more information.

**Reject unverified recipient** Normally an email server should know which internal email addresses are valid addresses (i.e., email addresses for which an inbox exists). When an email server is setup to relay email for certain domains the email server should know which recipients will be accepted by the server it relays to (in other words it should be a smart relay host). If all email is accepted for relay without knowing whether the next email server will accept the email, there is a risk of generating “backscatter” bounces. Backscatter bounces, occur when an intermediate email server accepts a message without checking whether the next email server accepts the message. Because the intermediate email server accepted the message, it has to be bounced back to the original sender when the next server does accept the forwarded email. If the email was a spam message using a forged sender, the sender will be flooded with bounced messages.

There are multiple ways for an email server to known which recipient addresses are acceptable and which are not. One solution is to let the gateway server “learn” which recipient addresses are acceptable by querying the server it relays to. When an email is received for a yet unknown recipient, the server “asks” the server it relays to whether the recipient is a valid recipient or not. The message is only accepted when the next email server reports that the recipient is a valid recipient. The result of this verification process is cached.

The verification procedure is enabled by checking *Reject unverified recipient*. The *reject code* is the SMTP result code used when the email is not accepted. This should initially be set to *450* (which tells the connecting SMTP

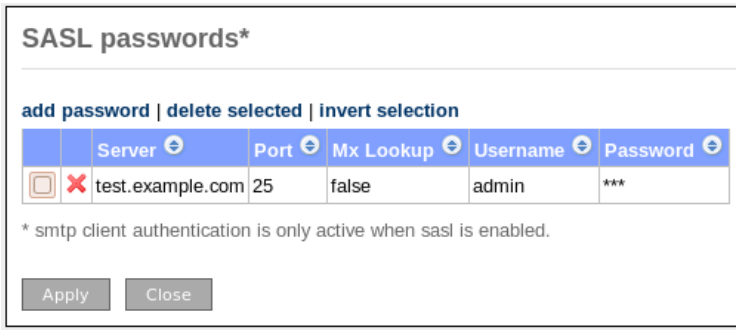


Figure 4: SASL passwords

server that the message is not accepted because of a temporary error). It should be changed to *550* (permanent error) when the verification procedure works correctly. See the Postfix documentation for more information on address verification[1].

There are other ways for the email server to know which recipients are valid, for example using LDAP queries or by specifying `relay_recipient_maps`. These other options are however not directly supported by the *MTA config* page and should therefore be configured using the *MTA raw config* page or by directly editing the Postfix configuration files.

**Applying changes** By clicking the *Apply* button, the changes will be checked and Postfix will be configured with the new settings. Clicking the *Close* button will redirect the browser to the *Admins* page.

## 3.3 sasl passwords

In cases where the *external relay host* or *internal relay host* requires SMTP authentication, a SASL account should be added. For example, if Gmail is used as the *external relay host*, the Gmail smtp server requires that the sender authenticates itself with the correct Gmail credentials. SMTP credentials for a specific host can be added by clicking *sasl passwords*. This opens the SASL passwords page (see figure 4).

SMTP password authentication is only active when SASL is enabled. For more information on how to enable SASL see Appendix B.

## 3.4 MTA raw config

Postfix has a large number of settings. The *MTA config* page only supports a small number of the relevant Postfix settings. For settings not supported by the *MTA config* page, the *MTA raw config* page can be used to directly edit the Postfix main config file (main.cf). The configuration file contains some specific DJIGZO settings (settings that start with *djigzo_*). These settings are modified by the *MTA config* page when applying the changes. These settings should

[1] See http://www.postfix.org/ADDRESS_VERIFICATION_README.html

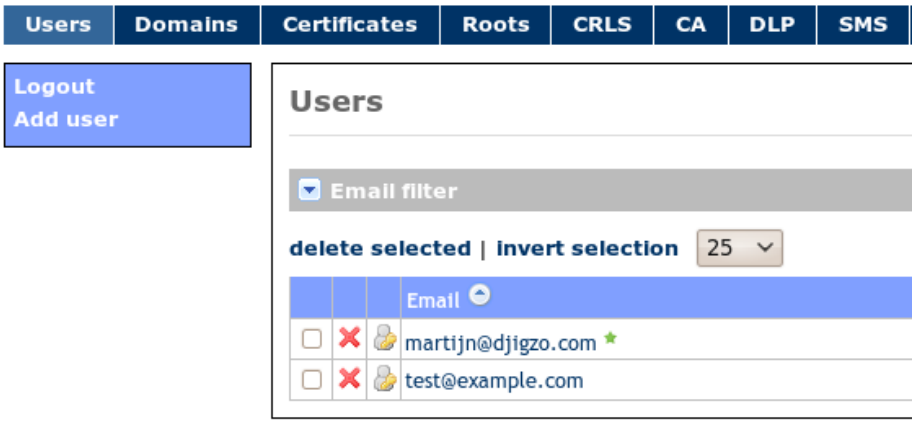


Figure 5: Users

therefore not be manually changed because they can be overwritten by the *MTA config* page. DJIGZO specific settings are used by other Postfix settings (they are referenced as *${djigzo_...}*).

**Note:** The gateway does not validate any changes made to the *MTA raw config* so care must be taken when modifying the Postfix config file directly. If the Postfix configuration file contains errors, Postfix might not function properly.

# 4 Users

A user is a sender or receiver of email and is uniquely identified by it's email address. Every user has a set of preferences that determine how email is handled for that particular user. The users page gives an overview of all the users (see figure 5).

Specific users can be searched for using the filter. Users can be removed by clicking the 'cross' icon or by selecting the users and clicking *delete selected*. New users can be added by clicking *Add user* on the left hand side menu. Clicking a user opens the user preferences page (see next section). Internal users are marked with a green star icon (see *Locality property* on page 13 for more info on the difference between internal and external users). Clicking the user certificate icon opens the certificate selection page for the user. If the user is an internal user, the *Select signing certificate* page is opened. If the user is an external user, the *select encryption certificates* page is opened.

## 4.1 User preferences

Every user has a set of preferences which determine how email is handled for that particular user. User preferences can inherit some or all of the preferences from higher level preferences.

Users inherit their preferences from domain preferences, domain preference inherit from the global preferences and the global preferences inherit from the factory preferences:

user ← domain ← wildcard domain[2] ← global ← factory.

The preferences for a user can be edited by clicking on the users email address (see figure 6). The user preference page links to sub-pages *select encryption certificates*, *select signing certificate*, *templates* and *global preferences* which can be used to edit additional preferences. Whether a preference is inherited or not is determined by the associated *inherit* checkbox. If checked, the preference is inherited.

Some preferences are only relevant for the sender of a message and some preferences are only relevant for the receiver of a message. Most preferences however are relevant for both the sender and receiver. The MPA mail flow in Appendix E shows exactly when and how preferences are used by the MPA.

**Message originator** DJIGZO uses the *From* header as the identity of the sender and not the "envelope sender". It is important to understand the difference between the from and the envelope sender because the identity of the sender determines which settings and certificates will be used by the sender. Because sender is such a confusing term, DJIGZO will use the term *originator* when referring to the identity of the sender (i.e., the from header).

The rationale for using the *From* for the identity of the sender is that the envelope sender is more or less similar to the postman and is therefore only responsible for the message delivery and not for the message content whereas the *From* identifies the actual originator of the email. The user identified by the *From* is responsible for the message content. In most cases however, the envelope sender is the same as the *From*.

Another important reason why the from header is used, is that S/MIME uses the from header as the identity of the sender, i.e., an S/MIME client compares the email address of the signing certificate with the from header of the message. While the identity of the sender is determined by the from header, the identity of the recipient is determined by the recipient of the SMTP envelope and not the *To* header.

Next, a brief overview of the all the user preferences will be given.

### 4.1.1 General

**Locality** The *locality* of a user can be external or internal. The *locality* preference is a very important preference because it determines whether email coming into the gateway should be encrypted or decrypted. If the recipient of an email is an internal user, the email should be decrypted. If the recipient of an email is an external user, the message should be encrypted (whether the message will actually be encrypted depends on other user settings). Typically users for which the gateway receives email will be internal users (i.e., users belonging to one of the relay domains) and all other users will be external users.

[2] An example of a wildcard domain is *.example which matches test.example.com.

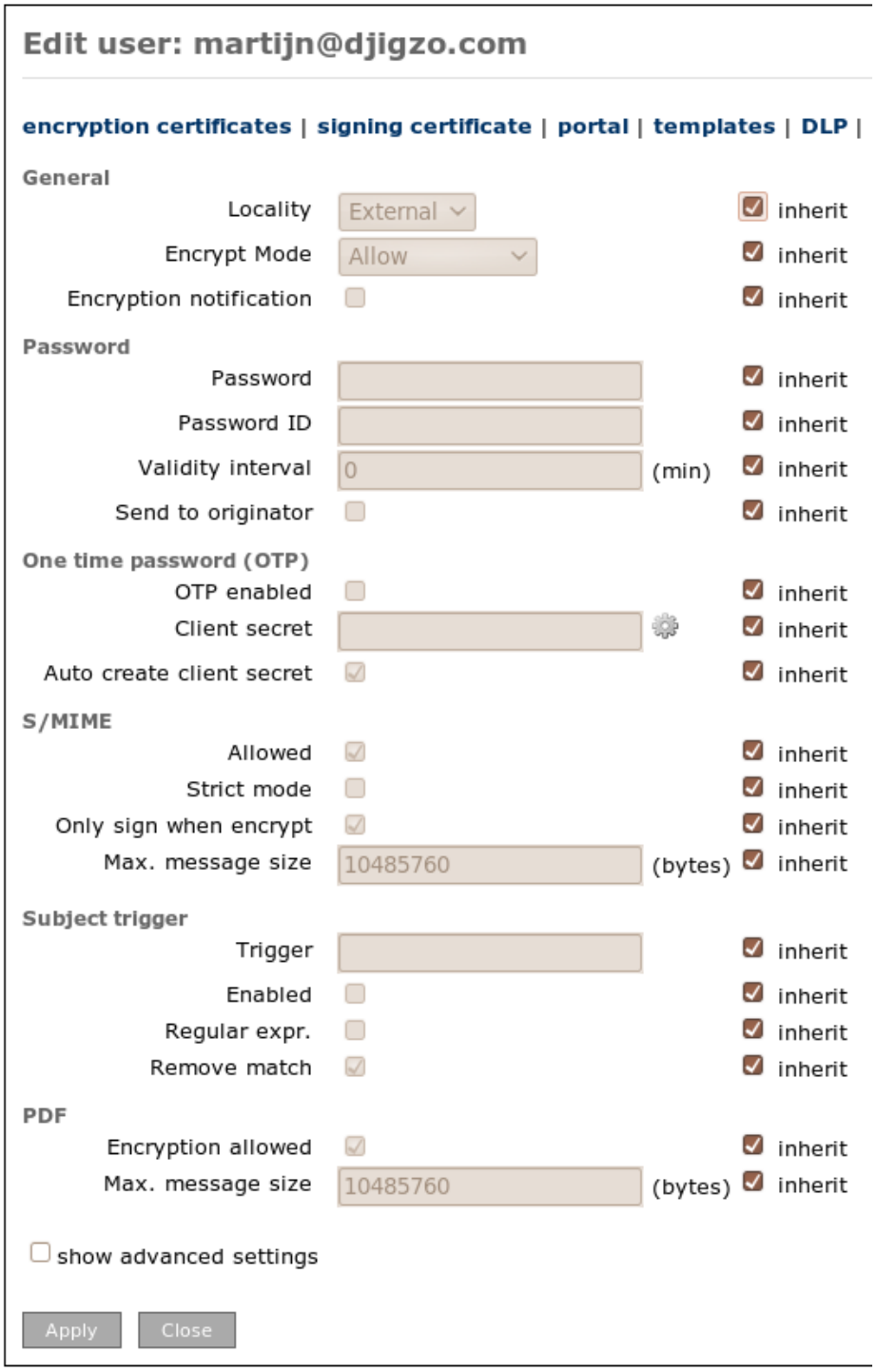


Figure 6: User preferences

The domain of a recipient is based on the SMTP envelope recipient. The domain of the sender is based on the *From* header[3].

**Note:** because the locality is such an important preference, it should be setup correctly. In most installations all domains for which the gateway receives email (i.e., the relay domains) should be internal domains. By default all users and domains are external.

**Encrypt mode** Encrypt mode determines whether a message sent to an external user (i.e., a user with external locality) should be encrypted or not. The possible encrypt modes are: *No Encryption*, *Allow*, *Mandatory* and *Force*. Encrypt mode is used for the sender and receiver. If encrypt mode is *No encryption*, the message will not be encrypted by default (unless being overruled by the *subject trigger*). If mode is *Allow* the message is only encrypted if it is possible to encrypt the message (i.e., a valid recipient certificate is available or PDF encryption is setup for the recipient). With *Mandatory* mode, the message must be encrypted and if it is not possible to encrypt the message, the message will not be sent and the sender will be notified.

Encrypt mode is a sender and receiver preference. This means that the settings for both the sender and receiver must allow encryption. If for example the sender has encrypt mode *Mandatory* and the recipient has encrypt mode *No Encryption*, the message will not be encrypted and will therefore not be sent (because the sender encrypt mode was *Mandatory*). If the sender (or recipient) has encrypt mode *Force*, the other encrypt mode is ignored and encryption is forced. *Force* encrypt mode is for example used when all email sent to an external recipient should always be encrypted regardless of the *Encrypt mode* of the sender.

**Encryption notification** If set, the sender of the message will be notified (with an email) when the message is encrypted (see template *successful encryption* on page 32 for the notification message template).

### 4.1.2 Password

**Password** The password for the user. Currently this is only used for PDF encryption. The password can be set by the administrator or can be randomly generated. If the current password has expired (see *Validity interval*) a new password will be generated when the password is used. If a static password should be used, password expiration should be disabled by setting *Validity interval* to -1.

**Password ID** The *Password ID* identifies the password used for PDF encryption. The *Password ID* is required when the "One Time Password" (OTP) mode is used or when the password is delivered by SMS Text.

With the OTP encryption mode, a password will be generated based on a secret key and on a unique identifier (the *client secret* and the *Password ID*).

[3] If there is no *From* header, the *Sender* header will be used. If the *Sender* header is also missing, the envelope sender will be used.

The *Password ID* is used by the recipient of the encrypted PDF message to regenerate the password (for more information about the OTP mode, see the PDF encryption guide). With the SMS Text mode, the randomly generated password will be delivered to the recipient via an SMS text message. Because the recipient can receive multiple passwords via SMS Text, the recipient has to know which password belongs to which encrypted PDF. The encrypted PDF messages therefore contain a password identifier which can be used to find the matching password. Everytime a new password is generated, a new unique password ID is generated. The *Password ID* property shows the last generated password ID for the user.

**Validity interval** The time (in minutes) the password is valid. If the password is no longer valid (expired) a new password will be generated when the password is used. If the *Validity interval* is 0 a new password will be generated every time a message is PDF encrypted. If *Validity interval* is -1 the password never expires.

**Send to originator** If checked, generated passwords will be sent to the originator of the message (i.e., the sender) as a notification message. This is currently only used when a password is generated for PDF encryption. The notification message will contain the generated passwords (see template *passwords* on page 32 for the notification message template). The originator is responsible for delivering the generated passwords securely to the recipients of the encrypted message.

### 4.1.3 One time password (OTP)

With the one time password mode, passwords for PDF encryption will be generated based on the *Client Secret* and on the *Password ID*. For more information on the OTP mode, see the PDF encryption guide).

**OTP enabled** If selected, the one time password mode is enabled for the user.

**Client secret** The *Client secret* is used to generate the one time password for the recipient.

**Auto create client secret** If OTP mode is enabled and the recipient does not yet have a *Client secret* and *Auto create client secret* is enabled, a new randomly generated client secret will be automatically created for the recipient.

### 4.1.4 S/MIME

**Allowed** If checked, digital signing and encryption using S/MIME is allowed.

**Strict** By default, the gateway tries to decrypt every incoming S/MIME encrypted message even if the recipient of the message does not have an associated private key with which the message can be decrypted. In other words, the gateway tries to decrypt the message with any suitable key ("decrypt if possible"). There are a couple of advantages when decrypting every incoming email irrespective of the recipient: *a*) it makes it easier to manage domain to domain encryption; *b*) forwarded email can be decrypted and, *c*) email handling with multiple recipients is faster because only one key is required for decryption.

Even though non-strict mode is easier from a management perspective, it is not as secure as *strict* mode. In non-strict mode, if an external attacker gets hold of an encrypted message, the attacker can resend the message to an internal accomplice, i.e., someone from inside the company who has access to an internal mail box and who works closely with the attacker. Because the message will decrypted with any available key, the message will be delivered decrypted to the insider even though the insider was not the original recipient.

In *strict* mode, additional checks will be done to make sure that the message will only be decrypted if the recipient has a valid decryption key. A message will only be decrypted for a recipient if the certificate associated with the private key for decryption is valid, trusted, not revoked and if one of the following is true:

**(a)** the recipient has a certificate and private key with a matching email address and the message can be decrypted with this private key or,

**(b)** the recipient has a certificate and private key and the certificate is explicitly associated with the user and the message can be decrypted with the private key or,

**(c)** the recipient is from a domain and the domain has an explicitly associated certificate and private key and the message can be decrypted with this private key.

If in strict mode, every recipient for which none of the above rules apply, will receive the message in encrypted form.

Whether or not to use strict mode depends mostly on whether you trust your internal users. If you do not trust all internal users, it's better to enable strict mode. If all internal users can be trusted, running in non-strict mode might be somewhat easier to manage.

**Note:** strict mode can be enabled and/or disabled per domain and per recipient. Although it's advised to only change the global strict settings, there are situations where it can be helpful to enable or disable strict mode per recipient. For example, suppose the global strict mode is enabled. However, because of email archiving purposes, the front-end SMTP server sends a copy (bcc) of every incoming email to the email archiver. Since the gateway is in strict mode, encrypted message won't be decrypted by the gateway when delivered to the email archiver. By disabling strict mode for the email archiver recipient, incoming email delivered to the email archiver will be decrypted.

**Only sign when encrypt** If checked, messages will only be digitally signed when they are S/MIME encrypted. If not checked, all messages will be digitally signed. The sender of a message must have a valid signing certificate before a message can be digitally signed.

**Max. message size** If the email message is larger than the specified maximum message size (in bytes) the message will not be S/MIME signed or encrypted. Large S/MIME messages can sometimes not be handled by S/MIME email clients. Another reason for limiting the size of S/MIME messages is that encrypting and signing of large email messages can be resource intensive.

### 4.1.5 Subject trigger

A subject trigger can be used to force encryption when the subject contains a certain keyword. This is useful when the default setting for a sender is *No encryption* while you want to force encryption for a particular message ("on demand encryption").

**Trigger** If the subject contains the provided trigger keyword and the subject trigger is enabled, encryption is forced for this message. Whether the message is really encrypted depends on the availability of certificates etc. If encryption is triggered but the message cannot be encrypted, the message will not be sent and the sender will be notified.

**Enabled** The subject trigger functionality will only be functional if *Enabled* is checked.

**Regular expr.** If checked, *Trigger* is interpreted as a regular expression and the subject is matched against this regular expression. **Example regular expression trigger:** `(?i)(\[secure\]|\[encrypt\])`. This subject trigger will force encryption when the subject contains [secure] or [encrypt]. (?i) makes the check case insensitive.

**Remove match** If checked, the matching part will be removed from the subject. **Example:** Suppose the trigger equals "[encrypt]" and the subject of the incoming message is "your bank statement [encrypt]" the subject after encryption is "your bank statement".

### 4.1.6 PDF

**Encryption allowed** If checked, PDF encryption is allowed.

**Max. message size** PDF encryption not only encrypts the message body but also encrypts the message attachments. To prevent the PDF from becoming too large, the PDF is not encrypted if the total size of body text and attachments exceeds the maximum message size (in bytes).

## 4.2 Advanced settings

The advanced settings sub page contain settings which are only used in specialized setups (see figure 7). Some settings can only be set for the global settings. See 4.3 for more information on the global specific settings.

### 4.2.1 General

**Server secret** The server secret is used to protect external resources against tampering (using the HMAC algorithm). For example the reply link in an encrypted PDF message is protected to make sure that a recipient can only reply to a message that was generated by the server. A global server secret will be automatically generated the first time the server starts. The server secret is a required setting. In most setups there is no need to override the inherited server secret.

**Force encrypt allowed** If checked, senders are allowed to trigger encryption of messages with a specific header (see *Force encrypt trigger*).

**Force encrypt trigger** The *Force encrypt trigger* can be used to trigger encryption (either S/MIME or PDF) of a message using a specific email header. All headers of an outgoing email are matched against the *Force encrypt trigger* and if there is a match, encryption is forced. If the header is present but the message cannot be encrypted, the message will be bounced back to the sender to notify that the message could not be encrypted.

Force encrypt trigger, for example, can be used when an automated system sends email to external recipients and some, but not all, emails should be encrypted. By adding a header to an outgoing email, the external system can specify whether the email should be encrypted or not.

The trigger is specified as: `HEADER[:REG_EXPR]`, where *HEADER* is the name of the header and *REG_EXPR* is the optional header value specified as a regular expression. If *REG_EXPR* is not specified, all header values are accepted. If *REG_EXPR* is specified, only those header values that match the regular expression will trigger encryption.

**Examples:** The following examples trigger encryption when the messages contains the `X-Djigzo-Encrypt` header. The header values are ignored (i.e., all header values are accepted).

```
X-Djigzo-Encrypt
X-Djigzo-Encrypt:
X-Djigzo-Encrypt:*.
```

The following example triggers only when the message contains a header named `X-Djigzo-Encrypt` with a header value of *true* (whitespace is ignored and checks are case insensitive).

```
X-Djigzo-Encrypt:(?i)^\s*true\s*$
```

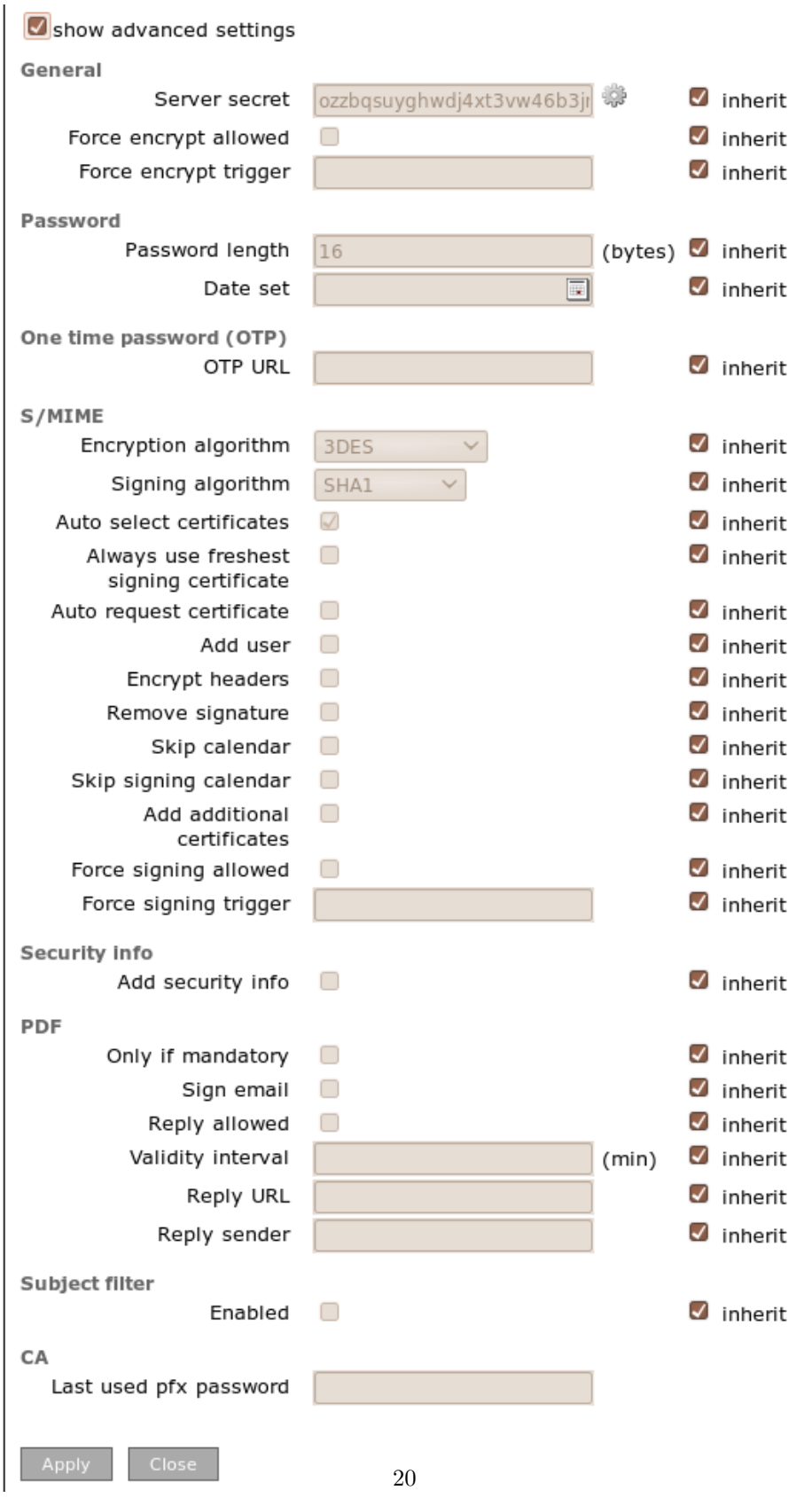




Figure 7: Advanced user preferences

### 4.2.2 Password

**Password length** The length (in bytes) of the randomly generated passwords. This is used when a new password for PDF encryption is automatically generated.

**Date set** The date at which the password was set. This is used in combination with the *validity interval* to determine whether the password is still valid. If *Date set* is empty, the password will never expire.

### 4.2.3 One time password (OTP)

**OTP URL** The recipient of an OTP encrypted PDF, needs to access the portal to generate the password for the PDF. The default external URL for the OTP password generator is based on the portal *Base URL* (see 4.6). It's advised to change the *Base URL* of the portal and only change the *OTP URL* if the OTP generator runs separately from the portal.

### 4.2.4 S/MIME

**Encryption algorithm** The encryption algorithm to use when encrypting the message. The following encryption algorithms can be selected: *AES256*, *AES192*, *AES128*, *3DES*, *RC2*, *CAST5*, *CAMELLIA256*, *CAMELLIA192*, *CAMELLIA128* and *SEED*.

**Note:** some S/MIME clients only support a subset of the available algorithms. For example Outlook only supports *AES256*, *AES192*, *AES128*, *3DES* and *RC2*. *3DES* is supported by all S/MIME clients.

**Signing algorithm** The signing algorithm to use when signing the message. The following signing algorithms can be selected: *SHA1*, *SHA256*, *SHA512* and *RIPEMD160*.

**Note:** some S/MIME clients only support a subset of the available algorithms. In order to validate SHA2 (SHA256 and SHA512) messages, Windows Vista with Outlook 2003 (or newer) is needed. In order to both sign and validate SHA2 messages, Windows Vista or 7 with Outlook 2007 or 2010 is needed (see http://blogs.technet.com/b/pki/archive/2010/09/30/sha2-and-windows.aspx).

**Auto select certificates** If checked, encryption certificates will be automatically selected for the recipient.

**Always use freshest signing certificate** The first time a message must be signed, the gateway automatically searches for a valid signing certificate for the sender. Once a signing certificate has been selected, the signing certificate will be used for all signing operations until the certificate is no longer valid.

If however *Always use freshest signing certificate* is selected, every time a message is signed, the newest signing certificate (i.e., a valid certificate with the latest "not before") is used.

**Auto request certificate** If checked and the sender does not yet have a valid signing certificate, a new certificate and private key will be automatically requested for the sender using the default Certificate Authority (see CA Settings on page 48).

**Add user** If checked and a certificate is available for the recipient, a user object will be created if a message is S/MIME encrypted.

**Encrypt headers** If checked, certain headers (*Subject*, *To*, *Cc*, *Reply-To* and *From*) are added to the encrypted message. This option is normally only used when encrypting email to a DJIGZO for Android user (it provides access to all the relevant headers from the smime.p7m attachment).

**Note:** the headers are added to the encrypted binary blob and are *not* removed from the message. Do not select this option if the recipient uses Outlook because Outlook does not support encrypted headers.

**Remove signature** If checked and an incoming message is signed, the signature will be removed from the message. This can be helpful when the email client used by internal users or some email handling software cannot handle digitally signed messages.

**Skip calendar** If checked, calendar messages[4] (for example Outlook meeting requests) are not digitally signed or encrypted. Some email clients, for example Outlook, cannot handle meeting requests if the meeting requests are digitally signed or encrypted.

**Skip signing calendar** If checked, calendar messages (for example Outlook meeting requests) are not digitally signed. Some email clients, for example Outlook, cannot handle meeting requests if the meeting requests are digitally signed. The difference between *Skip signing calendar* and *Skip calendar* is that when *Skip signing calendar* is checked but *Skip calendar* is not, messages can still be encrypted. This can be helpful when all email, including calendar messages, sent to a specific domain must be encrypted with a domain certificate.

**Add additional certificates** If checked and the message is S/MIME encrypted, the message will also be encrypted with the additional certificates. See 9.3 for more information.

[4] Messages with the content-type “text/calendar”

**Force signing allowed** If checked, senders are allowed to trigger signing of messages with a specific header (see *Force signing trigger*).

**Force signing trigger** The *Force signing trigger* can be used to trigger S/MIME signing of a message using a specific email header. All headers of an outgoing email are matched against the *Force signing trigger* and if there is a match, S/MIME signing is forced.

This can for example be used when an automated system sends email to external recipients and some, but not all, emails should be digitally signed. By adding a header to an outgoing email, the external system can specify whether the email should be signed or not.

The trigger is specified as: `HEADER[:REG_EXPR]`, where *HEADER* is the name of the header and *REG_EXPR* is the optional header value specified as a regular expression. If *REG_EXPR* is not specified, all header values are accepted. If *REG_EXPR* is specified, only those header values that match the regular expression will trigger signing the message.

**Examples:** The following examples trigger signing when the messages contains the `X-Djigzo-Sign` header. The header values are ignored (i.e., all header values are accepted).

```
X-Djigzo-Sign
X-Djigzo-Sign:
X-Djigzo-Sign:*.
```

The following example triggers only when the message contains a header named `X-Djigzo-Sign` with a header value of *true* (whitespace is ignored and checks are case insensitive).

```
X-Djigzo-Sign:(?i)^\s*true\s*$
```

### 4.2.5 Security info

**Add security info** If checked and an incoming email is S/MIME encrypted or signed, information about the encryption or signature will be added to the subject. For more information see 4.3.1.

### 4.2.6 PDF

A brief explanation of the advanced PDF preferences will be given. See the *PDF Encryption Guide* for more information on how to setup PDF encryption.

**Only if mandatory** If checked, PDF encryption will only be enabled if encryption is mandatory (for example, if encrypt mode is mandatory, or encryption is triggered using the subject trigger).

**Sign email** If checked and the sender has a valid signing certificate, the email containing the encrypted PDF will be S/MIME digitally signed[5].

**Reply allowed** If checked, the encrypted PDF will contain a *Reply* link which can be used by the recipient of the encrypted PDF to securely reply to the message using the built-in portal.

**Validity interval** If checked, the *Validity interval* determines how long (in minutes) a reply link is valid.

**Reply URL** A recipient can securely reply to the PDF by clicking the reply link in the PDF. The reply link opens the reply page of the built-in portal using the default web browser. The reply URL should be setup to link to the external URL of the PDF reply page. The default reply URL is based on the portal *Base URL* (see 4.6). It is therefore advised to change the *Base URL* of the portal and only change the *Reply URL* if the PDF reply page runs separately from the portal.

**Reply sender** The envelope sender of the PDF reply message will be set to *Reply sender*. The local name part of the *From* header of the reply message will be set to the email address of the replying user prefixed with "in name of".

**Example:**If the user martijn@djigzo.com replies to the encrypted PDF using the reply portal page, the reply will contain the following from:

```
"in name of martijn@djigzo.com" reply@example.com
```

(where reply@example.com is the *Reply sender* address and martijn@djigzo.com is the email address of the user that replies).

The *Reply-To* header is set to the email address of the replier to make sure that a reply is sent to the correct recipient.

A reply using the reply portal page is always sent using the same sender because:

**(a)** The reply sender is always a known address. The encryption rules for the reply sender can therefore be specified. For example, it's possible to force all PDF reply messages to be encrypted.

**(b)** If for some reason the reply message is bounced, the bounce will not be sent to the original sender but to the *Reply sender*. This prevents the bounced message from accidentally being sent over the Internet without encryption.

**(c)** Using the real email address of the replying user as the envelope sender requires the gateway to "spoof" the sender address. If the sender domain of the replying user has defined any SPF records, the reply can be flagged as a forgery and therefore blocked by spam filters.

[5]The email containing the PDF is signed, not the PDF itself.


Figure 8: Global advanced user preferences

#### 4.2.7 Subject filter

If enabled, the subject of incoming email will be filtered with the filter setup on the global settings (see 4.3.2 for more information).

#### 4.2.8 CA

**Last used pfx password** If a pfx file was generated, *Store password* was selected when generating the pfx and the pfx was sent by email to the user, the password for the pfx file will be stored in the *Last used pfx password* setting (see 12.4 for more information).

The user preference sub-pages *select encryption certificates*, *select signing certificate*, *templates* and *global preferences* will be explained in later paragraphs.

### 4.3 Global advanced settings

Certain settings can only be set for the global settings (see figure 8 for the global specific settings).

#### 4.3.1 Security info

**Add security info** If checked and an incoming email is S/MIME encrypted or signed, information about the encryption or signature will be added to the subject. The actual text that will be added to the subject depends on whether the message is encrypted or signed and whether the signature is valid.

**Decrypted tag** If an incoming message is encrypted, the *Decrypted tag* will be added to the subject.

**Signed tag** If an incoming message is signed and the signature is valid, the *Signed tag* will be added to the subject.

**Signed by tag** If an incoming message is signed and the signature is valid but the email address of the sender (the from header) is not the same as the email address of the signing certificate, the *Signed by tag* will be added to the subject with %s replaced by the email address of the signing certificate.

**Invalid signature tag** If an incoming email is signed but the signature is not valid (for example the signing certificate is not trusted), the *Invalid signature tag* will be added to the subject.

**Note:** since an external sender can add these tags to an existing message (i.e., "spoof" that the message was protected), the existence of any of these security info tags should not be used as a proof that the message was encrypted and/or signed. Whether or not the message was really signed and/or encrypted can only be checked with 100% certainty by looking at the `X-Djigzo-Info` headers (see Appendix A of the DJIGZO S/MIME setup guide for more information on the `X-Djigzo-Info` headers). The *Subject filter* (see next section) can be used to remove all of the security info tags of incoming email to make sure that an external sender cannot "spoof" that the message was encrypted and/or signed.

### 4.3.2 Subject filter

**Enabled** If checked, the subject of incoming email will be filtered.

**Filter** The filter with which the subject will be filtered. The filter should be specified as:

```
/REG-EXP/ VALUE
```

where REG-EXP is the regular expression that will be used to match part of the subject and VALUE is the string that will replace the matched part. If VALUE is not set, the matched part will be removed from the subject (i.e., it will be replaced with an empty value)

**Example:** the following subject filter can be used to remove the default security info tags from incoming email:

```
/\[(decrypted|signed|signed by:.*|invalid signature!)\]$/
```

## 4.4 Mobile

The mobile sub settings page contains settings which are only required when using *DJIGZO for BlackBerry*. See *DJIGZO for BlackBerry administration guide* for more information.

## 4.5 SMS

**Phone number** The phone number of the recipient to which SMS Text messages will be sent. Passwords for the encrypted PDF or passwords for encrypted certificates can be sent via SMS Text messages. The phone number should be in international format (i.e., including the country code).

**Send SMS** If checked, the sender of the message is allowed to send SMS Text messages.

**Receive SMS** If checked, the recipient of the message is allowed to receive SMS Text messages.

**Phone number allowed** If checked, senders are allowed to specify a telephone phone number on the subject of an outgoing message. This telephone number is used by the PDF encryption functionality to send passwords via SMS Text messages. The telephone number is only used when the subject trigger is specified (see *Subject trigger* on page 18) and when the telephone number is at the end of the subject line. The telephone number can start with a + and may contain spaces, and the following characters (excluding the quotes) "-()".

**Examples:** Suppose that the subject trigger is [encrypt].

The following subjects contain valid telephone numbers:

**(a)** This is a subject with a phone number [encrypt] +31123456

**(b)** Encrypt this [encrypt] +31-(123)456

The following subjects do not contain valid telephone numbers:

**(a)** This is a subject with an invalid phone number [encrypt] 31=456

**(b)** Another example with an invalid phone number +31123456 [encrypt]

It should be noted that only one recipient (*To*, *Cc* or *Bcc*) at the same time is supported when the telephone number is in the subject. With multiple recipients it would be impossible to match the recipient with the correct telephone number. If the message has more than one recipient, the message will not be sent and the sender will be notified.

**Default country code** The telephone number in the subject should be in international format (i.e., including the country code). If the telephone number starts with a zero (0), which is not a country code, the server will add the default country code to the telephone number to make it a complete international telephone number. The *default country code* is only used when the telephone number is specified on the subject. The *Default country code* is not used by the telephone number that has been explicitly set using the administration page (see *Phone number* on page 27) since that number should always be set in international format.

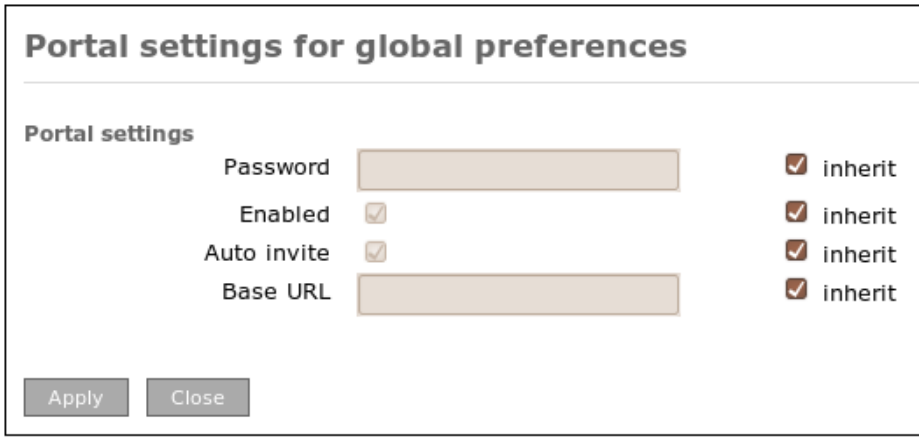


Figure 9: Portal settings

## 4.6 Portal

The portal can be used to provide the following functionality for external users:

- PDF reply
- Manage a DLP quarantined message
- Generate the one time password

The portal settings page can be used to setup certain aspects of the portal (see figure 9).

**Password** The password is used by the external user to login to the portal. If an external user receives a PDF encrypted email which was encrypted with a one time password (OTP), the external user can login to the portal to retrieve the password for the PDF. If no password is set for the user, the user cannot login.

**Enabled** If set, the user can login to the portal using the email address of the user as the login name and the portal password for the user. If not set, the user cannot login.

**Note:** the enabled setting is only used to specify whether the user can login. If not set, users can still reply to a PDF since replying to a PDF does not require the user to login.

**Auto invite** If the *Auto invite* setting is set and a one time password encrypted PDF gets sent to the user, the user is “invited” to select a new password. See the PDF encryption guide for more information.

**Base URL** To access the portal functionality, external users need to connect to the portal. The URLs to which external users need to connect to are written to the emails and encrypted PDFs (for example the reply link in the PDF). To make sure the URLs are externally accessible URLs, the gateway has to know what the correct external URL of the portal is[6]. The *Base URL* is not directly used, but is used as the base for the following URLs: PDF reply URL, OTP URL and DLP Quarantine URL. The *Base URL* can only be set for the global settings.

**Example:** In most setups, the base URL should look similar to*:

```
https://www.example.com/web/portal
```

* replace `www.example.com` with the domain name or IP address of the real server.

**Note:** since all other URLs used by the gateway are based on the *Base URL*, it's advised to only set the *Base URL* and not set the other URLs. The other URLs only need to be explicitly set if some specific functionality uses a different URL than the portal base URL.

# 5 Domains

The domains page gives an overview of all the domains that have been explicitly added by the administrator (see figure 10). A domain can be used to setup preferences for all users of that domain. For example, a domain can be added to create a secure S/MIME tunnel between two organizations. Because all users from a specific domain inherit the preferences and certificates from that domain, every email sent to a user in that domain will be encrypted with the domain certificate. Normally a *virtual private network* (for example a TLS connection) is used for a secure tunnel between email servers. However, the problem with a VPN is that each intermediate email server must support encrypted connections and each intermediate server needs to be fully trusted (the email is stored unencrypted on the email server until forwarded to the next hop). When email is sent to domains which cannot be guaranteed to be secure (like for example Hotmail or Yahoo) use of an encrypted channel cannot be enforced. With S/MIME tunneling, the message itself is encrypted and not just the connection. Because the message itself is protected, it can be sent over an unsecured connection.

Wild-card domains are also supported. For example user *test@example.com* inherits the preferences and certificates from the wild-card domain **.example.com* and from *example.com*. If a domain is in use (i.e., there is a user in the users list from that domain) the domain can no longer be removed (indicated by the missing red "cross") until all users from that domain are removed.

[6] In most typical setups, the gateways internal IP address is different from the external IP address (NAT).

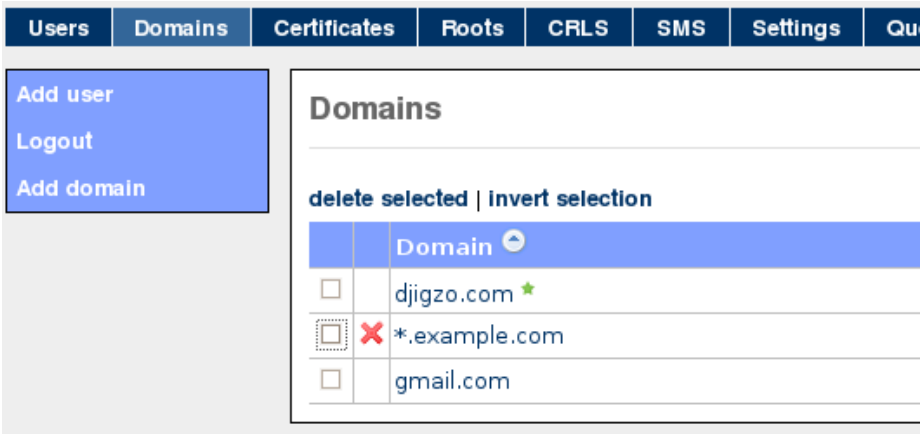


Figure 10: Domains

# 6 Templates

Some actions require the gateway to send email or SMS Text messages. These messages are created from message templates which can be modified by the administrator[7]. The templates are MIME encoded messages. When modifying the templates, care should be taken that the templates are valid MIME messages. The following templates can be edited (see figure 11):

- encrypted PDF
- Encrypted PDF via SMS
- Encrypted PDF OTP
- Encrypted PDF OTP invite
- Encryption failed notification
- Encryption notification
- Passwords notification
- SMS with password
- BlackBerry S/MIME
- SMS PFX password
- PFX email
- DLP warning
- DLP quarantine
- DLP block

[7]The templates are processed using the Freemarker template engine (see http://freemarker.sourceforge.net)

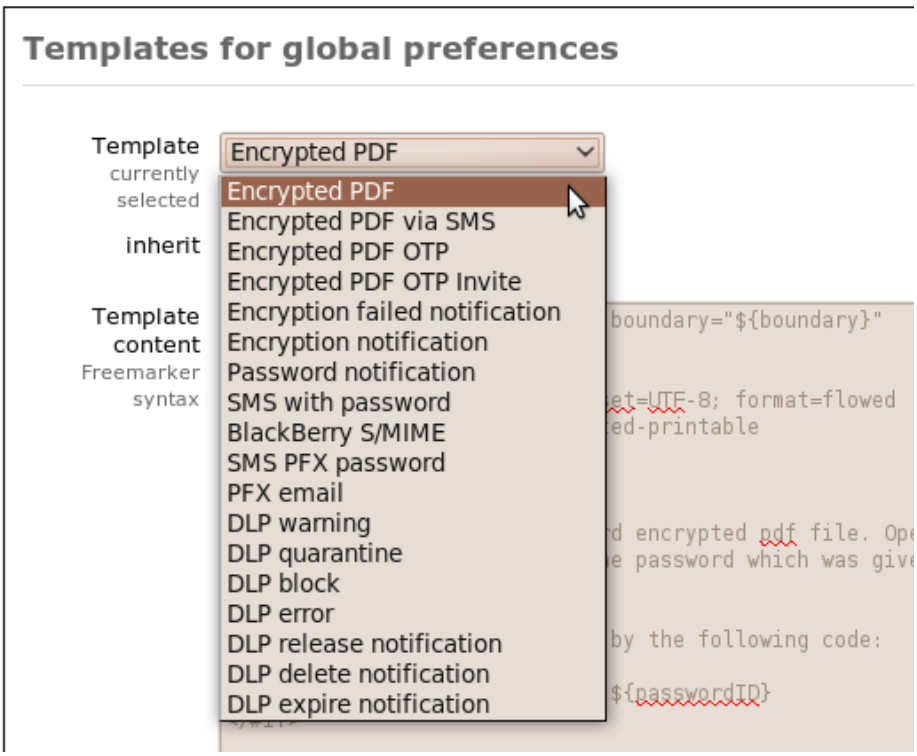


Figure 11: Message templates

- DLP error
- DLP release notification
- DLP delete notification
- DLP expire notification

**Note:** The *SMS PFX password* and *PFX email* templates can only be edited for the global settings.

**encrypted PDF** The template for the final message of an encrypted PDF message. When a message is PDF encrypted the actual message content, including the attachments, is converted to an encrypted PDF. This encrypted PDF is then attached to a message and the message, with the encrypted PDF, is sent to the final recipient. The PDF attachment in the template is just a "dummy" PDF which will be replaced by the real encrypted PDF. This template is used when the PDF password was not newly generated (i.e., the PDF password was a static password or was still valid).

**Encrypted PDF via SMS** This template is similar to the *encrypted PDF* template. The only difference is that this template is used when the PDF password

is newly generated and the password was sent via an SMS Text message to the recipient.

**Encrypted PDF OTP** This template is similar to the *encrypted PDF* template. The only difference is that this template is used when the PDF password is generated using the one time password (OTP) functionality.

**Encrypted PDF OTP invite** This template is similar to the *Encrypted PDF OTP* template. The only difference is that this template is used when the recipient does not yet have a password set.

**Encryption failed notification** Template used for the notification message that the message could not be encrypted but encryption was mandatory.

**Encryption notification** Template used for the notification message that the message was successfully encrypted. This template is only used when the sender of the message has *Encryption notification* enabled (see page 15).

**Passwords notification** Template used for the notification message containing newly generated passwords (see *Send to originator* on page 16 for more info).

**SMS with password** Template for the SMS Text message containing the generated password. The complete SMS Text message should fit in one SMS Text message (maximum 160 characters). The template should therefore not be too large.

**BlackBerry S/MIME** Template used for the S/MIME email message when the recipient preference *Recipient uses add-on* is enabled. Any S/MIME message sent to a recipient having *Recipient uses add-on* enabled, is converted to a message that can be read on a BlackBerry using the *DJIGZO for BlackBerry* add-on. See *DJIGZO for BlackBerry administration guide* for more information.

**SMS PFX password** Template for the SMS Text message containing the password for the encrypted private key file. The complete SMS Text message should fit in one SMS Text message (maximum 160 characters). The template should therefore not be too large. For more information see the CA section 12. Note that this template can only be edited for the global settings.

**PFX email** Template for the email containing the password protected private key file (.pfx). For more information see the CA section 12. Note that this template can only be edited for the global settings.

**DLP templates** For more information about the DLP specific templates, see the separate *DLP setup guide*.

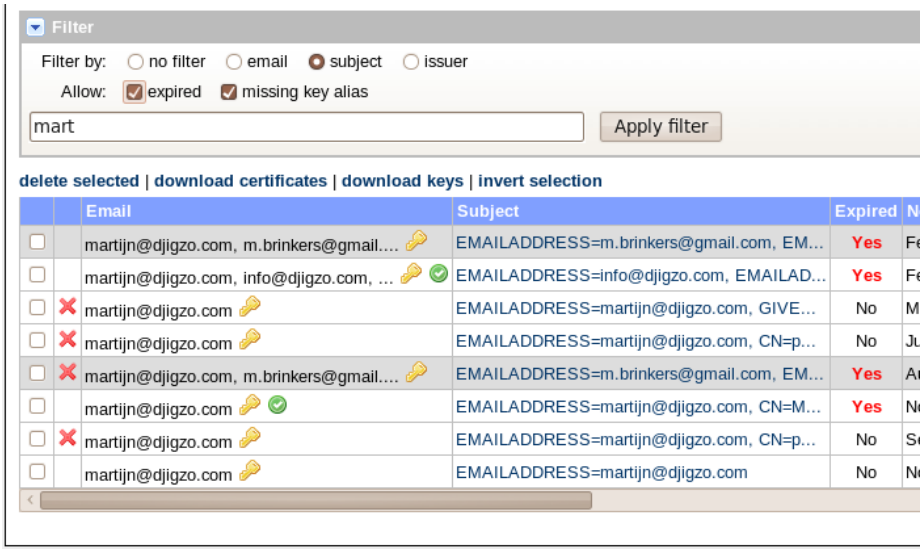


Figure 12: Certificate store

# 7 Certificates

DJIGZO supports S/MIME for encryption and digital signing of email messages. S/MIME is based on PKI and X.509 certificates[8]. The system has a built in X.509 certificate store. Certificates can be manually added and removed by the administrator. Certificates attached to incoming digitally signed messages are automatically extracted from the signature and are added to the *Certificates* store. The certificate store supports unlimited number of certificates[9]. Intermediate and end-entity certificates are stored in the *Certificates* store and root certificates are stored in the *Roots* certificate store. The Certificates page shows all the certificates in the *Certificates* store (see figure 12).

Specific certificates can be searched with the certificate filter. Certificates which are not valid (not signed by a trusted root, revoked, expired etc.) are shown in gray. Certificates with an associated private key contain the *key* icon. Certificates which are revoked are shown in red.

DJIGZO follows RFC 3280 (“Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List Profile”). Selected certificates can be downloaded or deleted. When a certificate is in use (by a user, domain or global settings) the certificate cannot be deleted (indicated by a missing “red cross”).

Certificate details can be opened by clicking on the certificate subject (see figure 13). By clicking the *usage* sub-menu it is shown which users or domains are using the certificate. The certificate details page can be used to get more info as to why a certificate is not valid. For example, figure 13 shows that the certificate is not trusted because the certificate is not trusted by a root

[8]For more information on S/MIME and X.509 see http://en.wikipedia.org/wiki/SMIME and http://en.wikipedia.org/wiki/X.509

[9]Limited by the size of the database. If a HSM is used, the HSM can impose a limit on the number of certificates.

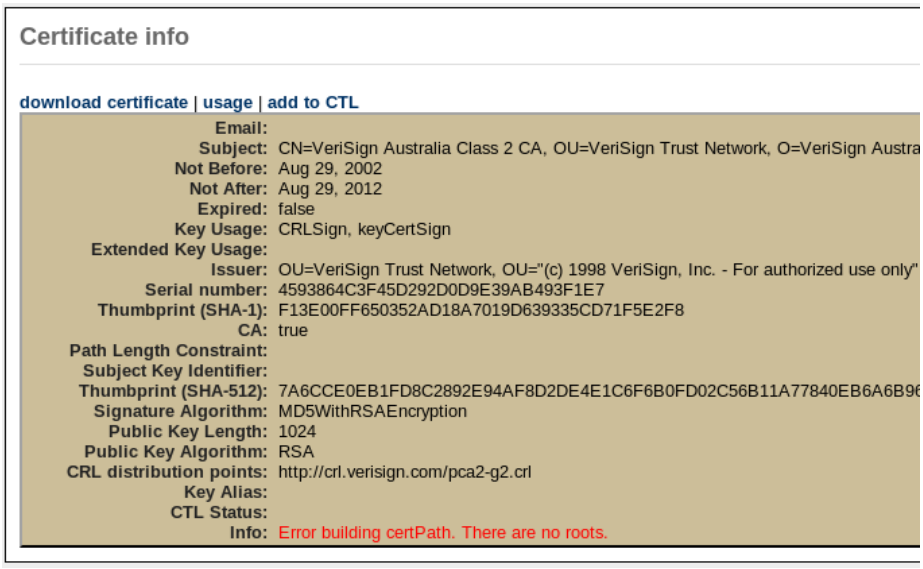


Figure 13: Certificate details

certificate (the certificate chain is incomplete).

The *Certificates* store can also contain private keys associated with the X.509 certificate. Private keys can be used for decrypting S/MIME encrypted messages or for digital signing of messages. An entry with an associated private key has a non-empty *Key alias* (see for example figure 13). By unchecking the *allow missing key alias* checkbox only certificates for which there is a private key available will be shown.

The *Roots* store contains certificates which are fully trusted by the administrator (i.e., trust is explicit and not inferred from other certificates). The *Roots* store normally only contains certificates (i.e., the *Key alias* is always empty).

Certificates can be manually imported by the administrator. Certificates can also be added to the certificate store when an incoming S/MIME protected email has attached certificates. Any attached certificates are extracted from the message and are stored in the certificates store. Most S/MIME signed messages contain at least the signing certificate.

The certificate can be added to the *Certificate Trust List* by clicking *add to CTL*. *Certificate Trust List* will be explained in section 11.

## 7.1 Importing Certificates

Certificates can be imported into a store with the *Import certificates* page (see figure 14). A certificate file, a store to import to and additional import parameters should be selected. Files with just one certificate (DER or PEM encoded) and files with multiple certificates (.p7b) are supported. Importing a large number of certificates can take some time. After the import the *Import certificates* page will show how many certificates were imported.

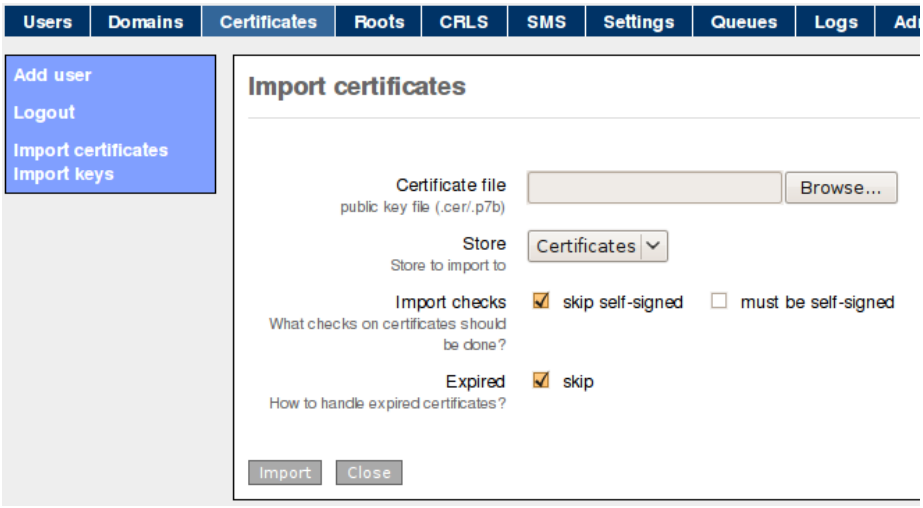


Figure 14: Certificate import

## 7.2 Importing keys

Certificates with associated private keys can be imported into the *Certificates* store using the *Import keys* page (see figure 15). A password protected *PKCS#12* private key file (.p12 or .pfx) must be selected and uploaded.

## 7.3 Download certificates and keys

Certificates and associated private keys can be downloaded from the gateway. Select the certificates that need to be downloaded and click *download certificates* or *download keys* from the sub-menu. When downloading private keys, a password used for encrypting the private key file (.p12 file) must be entered.

# 8 S/MIME

In this section a brief introduction of S/MIME will be given. S/MIME is based on *Public Key Infrastructure* (PKI) and uses X.509 certificates.

## 8.1 PKI

Public Key Infrastructure is a technology which can be used to securely exchange information over insecure networks using public key cryptography. PKI uses X.509 certificates to bind a public key to an identity. The main advantage of PKI is that there is no need to directly trust everyone involved because trust can be inferred. Roughly speaking there are two trust models in use today: hierarchical (via trusted CAs) or “Web Of Trust”.

With the hierarchical trust model, trust is inferred bottom-up. The root (the bottom) is blindly trusted (that makes it by definition a root) and all leaf nodes and branches (the end-user and intermediate certificates) are trusted because


Figure 15: Key import

they are child's of the trusted root (to be precise the intermediate certificates are issued by the root certificate). S/MIME uses a hierarchical trust model.

In a "Web of Trust" model, trust is inferred from trusted neighbors in a mesh like structure (a web). **For example:** *Alice* trusts *Bob* and *Ted* trusts *Alice* and therefore *Ted* now also trusts *Bob* (through *Allice*). The hierarchical model can be viewed as a "Web of Trust" model with additional constraints.

Because trust is inferred from other entities, it is possible to securely check whether one entity trusts another entity and that it is not possible to "spoof" any trust. Trust checking is done using *Public Key Cryptography*. An intermediate certificate is digitally signed by the issuer of the certificate using the issuers private key. With the public key of the issuer, it can be checked whether the certificate was really issued by the issuer. The public key together with some extra information forms an X.509 certificate.

## 8.2 X.509 certificate

A typical X.509 certificate contains the following elements (this is a non-exhaustive list):

- Public Key
- Subject
- Email address
- Issuer
- Serial Number
- Not Before
- Not After
- Key Usage

- Extended Key Usage

An X.509 certificate is digitally signed by the issuer of the certificate. By digitally signing the certificate, any changes done after signing will break the signature. Any changes to the certificate will therefore be noticed. A brief introduction of some of the main elements of an X.509 now follows.

**Public Key** The public key, like the name already implies, is the key that everyone is allowed to know. If a message must be encrypted, the public key of the recipient is used for encryption. The public key is used to verify a digital signature (the digital signature is created with the associated private key).

**Subject** The subject of a certificate contains the name of the "owner" and optionally an email address (or sometimes multiple email addresses).

**Email address** A certificate can contain multiple email addresses. X.509 certificates for S/MIME should normally contain the email address for which the certificate was issued.

**Issuer** The issuer contains the name of the issuer of this certificate (i.e., the issuer element should be equal to the subject of the issuer). If the subject of a certificate is equal to the issuer of a certificate the certificate is most likely a self-signed certificate. Root certificates are almost always self-signed.

**Serial Number** Every certificate should have a serial number. The serial number should be unique for the issuer (i.e., an issuer should use the serial number only once).

**Not Before** This is the date at which the certificate becomes valid. If the current date is before the *Not Before* date, the certificate is not yet valid.

**Not After** This is the date at which the certificate is no longer valid. If the current date is after the *Not After* date, the certificate is no longer valid.

**Key Usage** The public key of the certificate can be used for multiple purposes. Sometimes however the issuer of the certificate wants to restrict the key usage to only certain types. The following key usage types can be identified:

- digitalSignature
- nonRepudiation
- keyEncipherment
- dataEncipherment
- keyAgreement

- keyCertSign
- CRLSign
- encipherOnly
- decipherOnly

If the key usage is not specified it implies that the key may be used for all purposes. 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*.

**Extended Key Usage** The extended key usage, if specified, further specifies for what purposes the certificate has been issued. The following extended key usages can be identified:

- anyKeyUsage
- serverAuth
- clientAuth
- codeSigning
- emailProtection
- timeStamping
- OCSPSigning
- IPSecEndSystem
- IPSecUser
- IPSecTunnel
- smartcardLogin

If the extended key usage is not specified it implies that the key may be used for all purposes. For S/MIME, if an extended key usage is specified, it should at least contain *anyKeyUsage* or *emailProtection*.

**Thumbprint** The thumbprint is strictly speaking not part of an X.509 certificate. The thumbprint is the *cryptographic hash*[10] calculated over the bytes of the encoded certificate. The thumbprint uniquely identifies a certificate. The default algorithm used by DJIGZO for calculating the thumbprint is *SHA-512*.

[10] See http://en.wikipedia.org/wiki/Cryptographic_hash_function for more info on cryptographic hash functions.

### 8.3 Revocation checking

Sometimes it can happen that a certificate should no longer be used. For example because 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 by a certificate authority (CA) and is periodically updated. A revoked certificate should no longer be used. When a CRL is not available or when the administrator would like to "black list" a specific certificate, the certificate can be added to the *Certificate Trust List* (CTL). For more info on CTL see section 11.

## 9 Certificate selection

When required DJIGZO will automatically select the correct certificates for signing and encryption. Only certificates that are valid (i.e., trusted, not expired, not revoked) are automatically used. This implies that certificates should be trusted by a root certificate. The root certificate store by default does not contain any certificates[11]. The administrator should therefore add all the root certificates trusted by the organization to the root store. If a root certificate is not available a certificate can be "white listed" by adding the certificate to the *Certificate Trust List*.

### 9.1 Encryption certificate selection

Encryption certificates can be selected for a user, a domain or for the global settings. The *select encryption certificates* page can be opened by clicking the *select encryption certificates* sub-menu on the preference page (see figure 6). The *select encryption certificates* page shows all the certificates that have been selected for the user (or domain or global settings). When the certificate selection page has been opened, by default, only certificates with matching email addresses will be shown (see figure 16).

Each user can have an unlimited number of associated certificates. The system tries to automatically select the certificates for a user based on strict PKI rules. The certificate will only be automatically selected when the email addresses match. If a certificate is not automatically selected (for example the email address in the certificate does not match the email address of the user) the administrator can force the usage of a certificate by manually selecting the certificate for this particular user. Figure 16 Shows that multiple certificates are selected for user *test@example.com*.

When a message is S/MIME encrypted all of the selected certificates for the recipient are used. This allows the recipient to open the message with one the private keys associated with one the public keys used for encryption. The main advantage of using all of the selected certificates is that it allows the recipient to use different keys for decryption. For example, the key stored on the recipients home computer can be used when the message is read at home and the key on the office computer can be used when the message is read at the office.

[11] A collection of some well known CA certificates can be downloaded from the DJIGZO website.


Figure 16: Select encryption certificates

The sender does not known at which location the recipient will open the email so it's better to encrypt the message with both certificates.

**Color coding** The selected certificates are color coded based on validity and inheritance of the certificates (see figure 17):

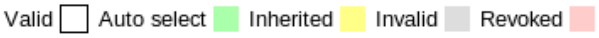


Figure 17: Color coding

Certificates can be manually selected and deselected by selecting the certificate checkbox and applying the settings. Automatically selected certificates cannot be deselected (the certificate can be completely removed if the certificate is no longer required). Uncheck *Auto select certificates* for the user if automatic selection of certificates for the user is not required.

Even when a certificate is manually selected it does not automatically mean that the certificate will be used for encryption. If a certificate is not valid it's not used for encryption. For example figure 16 shows that two certificates are manually selected. One certificate is however not valid (*gray color*) because it has expired. This certificate is therefore not used when a message is encrypted for this user. If a manually selected but invalid certificate must be used, the certificate must be made valid (add for example the certificate to the CTL to make it valid).

Besides selecting certificates, the *Select encryption certificates* page supports the creation of new end-user certificates for external users with the built-in CA server. A certificate can also be securely transported via email to an exter-

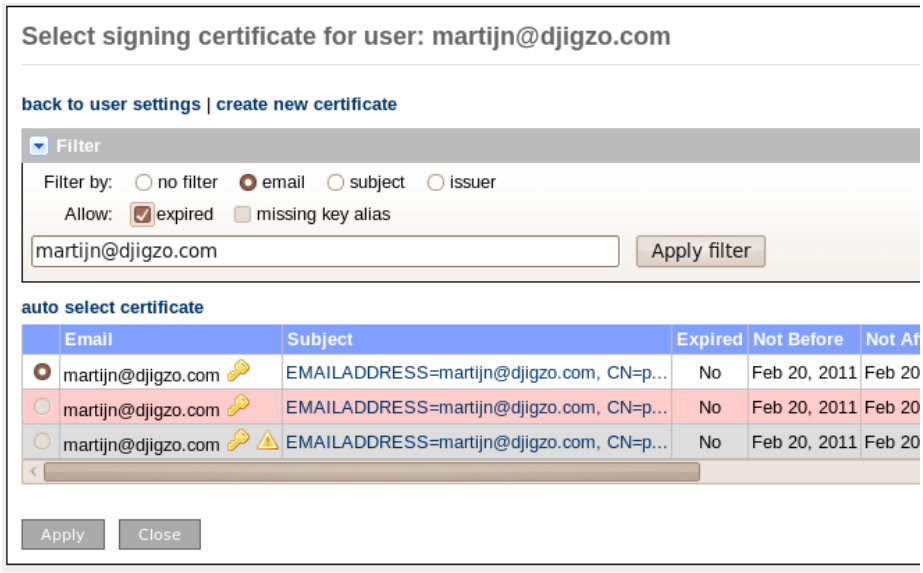


Figure 18: Select signing certificate

nal end-user. For more information on the built-in CA functionality see section 12.

## 9.2 Signing certificate selection

A signing certificate can be selected for a user, a domain or for the global settings. The *select signing certificate* page can be opened by clicking the *select signing certificate* sub-menu on the preference page (see figure 6). The *select signing certificate* page shows the signing certificate that has been selected for the user (or domain or global settings). When the certificate selection page has been opened, only certificates with matching email addresses will be shown by default (see figure 18). Only certificates with an associated private key can be selected and only one signing certificate per user can be selected at the same time. The system tries to automatically select a signing certificate by searching for a valid certificate with a matching email address. If there are multiple certificates suitable for signing, the first certificate found will be selected. The administrator can override the automatically selected certificate by manually selecting another certificate. If a certificate was manually selected the selected certificate can be reverted back to an automatically selected certificate by pressing *auto select certificate*.

## 9.3 Additional certificates

If a message is S/MIME encrypted, the message can also be encrypted with additional certificates. For example, a company policy might dictate that all encrypted data should be readable even when the sender and or recipient no longer have access to the private key (key escrow). Another reason to encrypt the data with an additional encryption certificate is that it allows a centralized

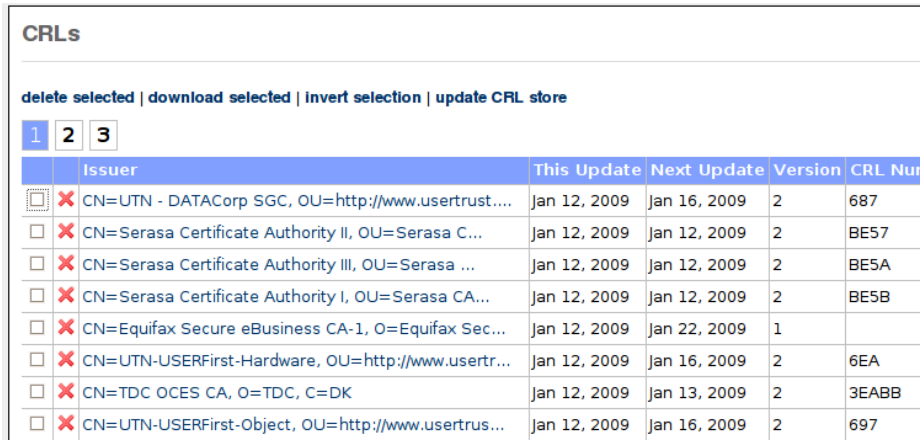


Figure 19: CRL store

virus scanner to scan the email. Additional certificates can be selected using the *additional certificates* sub-menu (see figure 16). Additional certificates are inherited just like regular encryption certificates and can be selected for the global settings, the domain settings or for a user.

**Note:** Anyone with access to the private key of the additional certificate(s) can in principle decrypt all encrypted email. The private keys of the additional certificates should therefore be stored in a safe place and only be used when required.

# 10 Certificate Revocation List

A certificate revocation list (CRL) is a list of certificates[12] which have been revoked and which should therefore no longer be used. Certificates can contain *CRL distribution points*. A *CRL distribution point* contains URLs from which the latest CRL can be downloaded. Periodically all the URLs from all the *CRL distribution points* for all the trusted certificates[13] are collected and the latest CRLs are then downloaded from these URLs[14]. The downloaded CRLs are stored in the CRL store (see figure 19).

CRL details can be viewed by clicking the CRL *Issuer* link (see figure 20). CRLs which are not valid (incorrectly signed, no path to a trusted root etc.) are shown in gray. The details page provides more information on why a CRL is not valid. By default the CRL store is periodically updated every 12 hours. A CRL update can be forced by clicking *update CRL store* in the sub-menu. The CRL entries (the serial numbers of the revoked certificates and optionally a revocation reason) can be downloaded as a text file by clicking *download CRL entries* (see figure 20).

[12] to be precise it's actually a list of serial numbers issued by the CA
[13] by default DJIGZO only downloads CRLs from valid certificates
[14] CRL distribution over HTTP(s) and LDAP is supported.

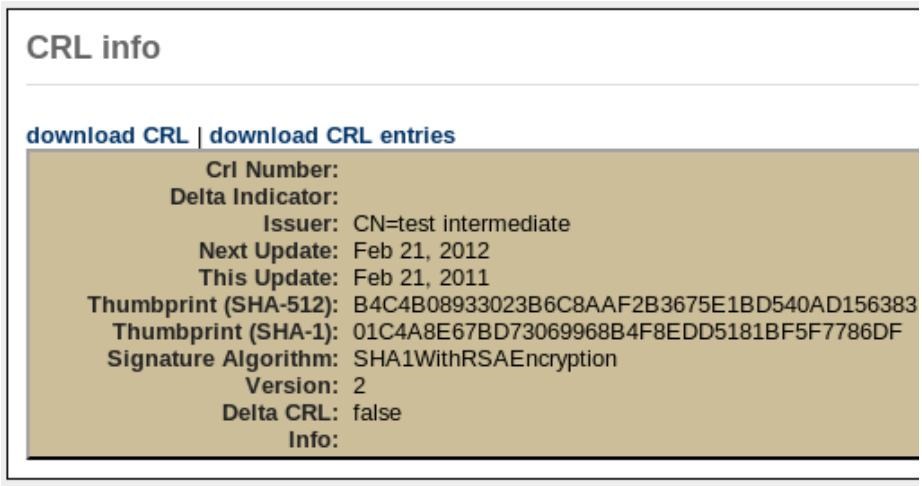


Figure 20: CRL details

**Note:** Some CRLs contain an extremely large number of revoked certificates. Downloading the entries of these large CRLs is therefore not advised.

# 11 Certificate Trust List

A Certificate Trust List (CTL) is a list of certificates (to be precise, a list of certificate thumbprints) which are explicitly trusted (*white listed*) or explicitly distrusted (*black listed*). The administrator can manually add or remove certificates to the Certificate Trust List.

In most cases PKI is sufficient for deciding whether or not a certificate is valid. Sometimes however, the administrator needs more control over this automatic process. Some examples when a CTL can be helpful:

**(a)** A certificate should no longer be used because it was compromised but the certificate issuer does not have a CRL. In this case the administrator can *black list* the certificate.

**(b)** 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). After *white listing* the certificate the administrator can manually select the certificate for a user.

**(c)** 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 and checking the *Allow expired* checkbox the certificate can now be manually selected.

By clicking *Certificate Trust List* on the left hand side menu of the certificates page (see figure 21) the *Certificate Trust List* can be opened. The *Certificate Trust List* contains the thumbprints of all the certificates that have been added

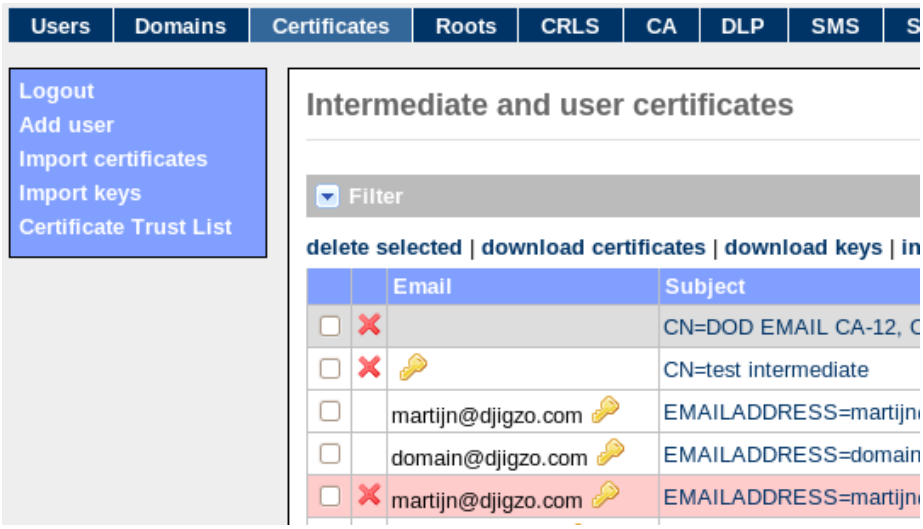


Figure 21: Open Certificate Trust List

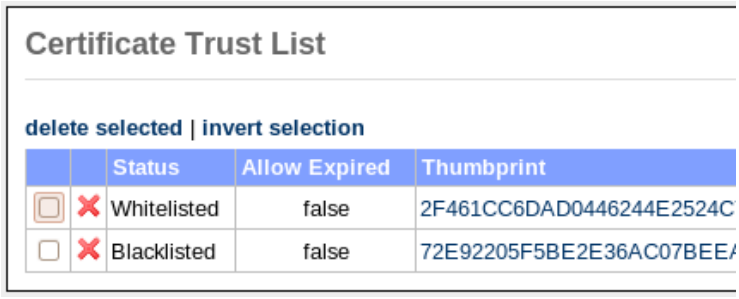


Figure 22: Certificate Trust List

to the CTL (see figure 22). A new entry can be added to the CTL by clicking *Add CTL entry* on the left hand side menu. A CTL entry does not directly contain a certificate, it contains the thumbprint of a certificate. The reason for this is that it should be possible to *white list* or *black list* a certificate even if the certificate is not available in the certificate store. An expired certificate can be made valid by by checking *Allow expired*. *Allow expired* is only applicable when *white listing* a certificate.

**Trust inheritance** *Black listing* a certificate is inherited by certificates issued by that certificate. This means that if an intermediate certificate is *black listed* all certificates, directly or indirectly, issued by that intermediate certificate are also *black listed*. *White listing* is not inherited. If an intermediate certificate is *white listed*, certificates issued by that intermediate certificate are not automatically *white listed*. For a certificate to be *white listed* it has to be explicitly added to the *Certificate Trust List*.

**Example:** Suppose a trusted root has issued multiple intermediate certificates. Normally all the intermediate certificates issued by that root are trusted. The administrator however does not trust one of the intermediate certificates. Removing the root from the root store won't help because the result would be that none of intermediate certificates are trusted. By *black listing* one of the intermediate certificates all certificates issued by that intermediate certificate will also be (implicitly) *black listed*.

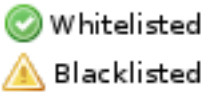


Figure 23: Certificate Trust List Icons

**CTL icons** When a certificate is *white listed* or *black listed* an icon is added next to the email address of the certificate (see figure 24). The green icon is shown when the certificate is *white listed* and the yellow icon is shown when the certificate is *black listed* (see figure 23). Clicking one of the icons will open the CTL entry page for the certificate.

**Final note:** the *Certificate Trust List* should normally only be used as a "work around" for when the standard PKI rules are not sufficient. Using the *Certificate Trust List* to manage trust is much more time consuming than managing trust using PKI.

# 12 Certificate Authority (CA)

The gateway contains a built-in CA server which can be used to create end-user certificates for internal and external users. This helps to quickly setup an S/MIME infrastructure without having to resort to external CAs for certificates and keys. Certificates and private keys can be securely transported to external recipients using a password encrypted certificate store (.pfx). The external recipients can use the certificate with any S/MIME capable email client like *Outlook*, *Outlook express*, *Lotus Notes* and start receiving and sending S/MIME encrypted email without having to install additional software.

The DJIGZO gateway contains a pluggable framework which allows new certificate request handlers to be registered. A certificate request handler is responsible for creating and/or retrieving a certificate and private key from internal or external CAs. By default, there are currently four certificate request handlers available: *built-in*, *delayed built-in*, *EJBCA* and *Comodo*.

A brief explanation of the CA functionality will now follow. For a more thorough overview on how to setup the S/MIME infrastructure see the separate *S/MIME setup guide*.

**Note:** the built-in CA has limited functionality. If support for multiple CA profiles, OCSP, CRLs for intermediate and root certificates is required a dedicated external CA should be used instead (for example EJBCA).

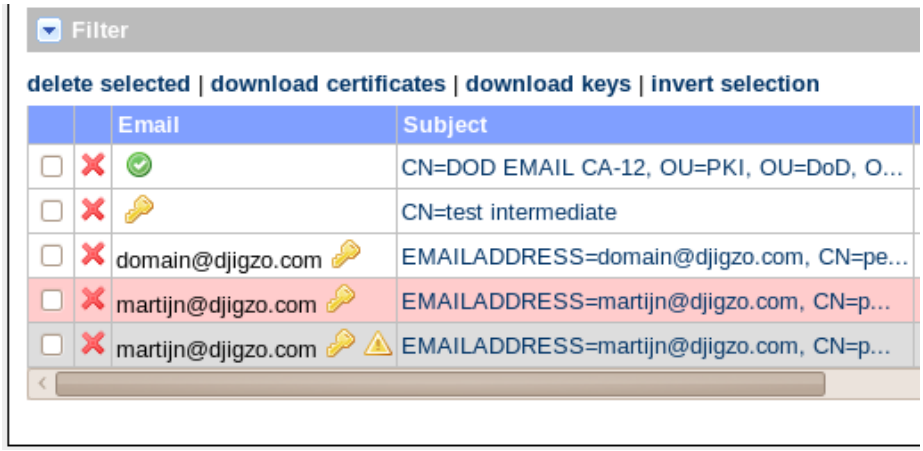


Figure 24: *white listed* and *black listed* certificates

## 12.1 Create new CA

Before starting to create end-user certificates, a root and intermediate certificate should be created[15]. The *Create new CA* page (see figure 25) can be used to create a new CA (i.e., create a new intermediate and root certificate).

Some details about the root and intermediate certificates should be specified before the certificates can be created (see figure 25).

**Validity** The number of days the root or intermediate certificate is valid (starting from the day it was created). This is a mandatory property.

**Key length** The length of the public key in bits (1024, 2048 and 4096). A 2048 bits key is sufficient is most cases. This is a mandatory property.

**Email** The email address that will be added to the certificate. Leave it empty (unless your policy requires an email address for your CA).

**Common name** The *common name* of the certificate is the main identifier of the certificate. The common name of the root certificate must be different from the common name of the intermediate certificate. Choose a unique common name and do not reuse a common name.

**More** Selecting *more* enables the following advanced settings for the subject: *organization*, *first name* and *last name*. These settings are only used to make it easier for end users to identify the CA certificates.

**Make default CA** If checked, the newly created CA will be the default CA.

[15] If a root and intermediate certificate is already available import them into the certificate and root store.

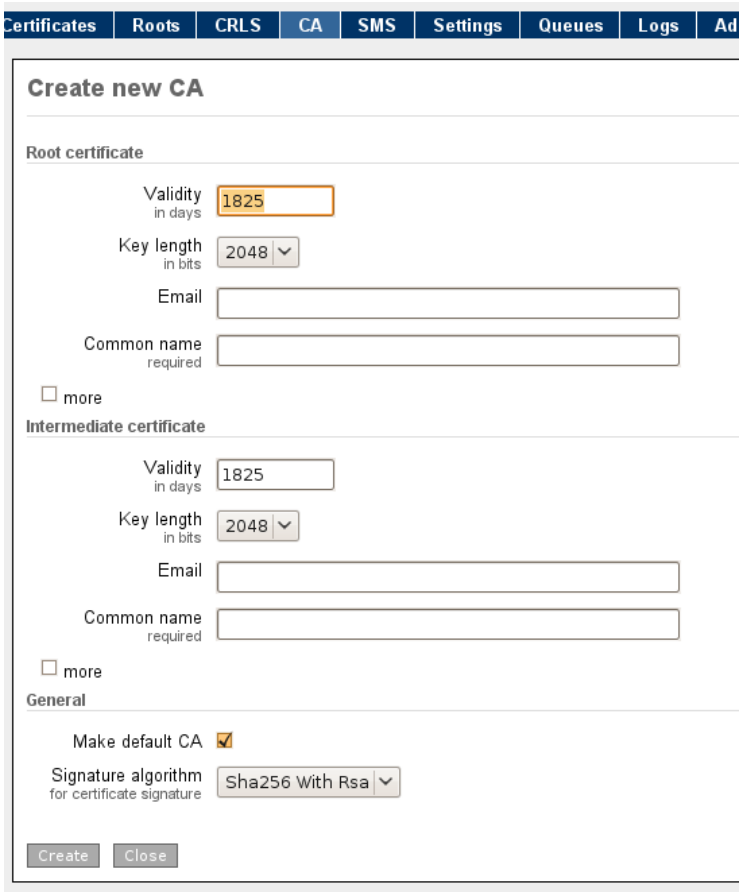


Figure 25: Create new CA

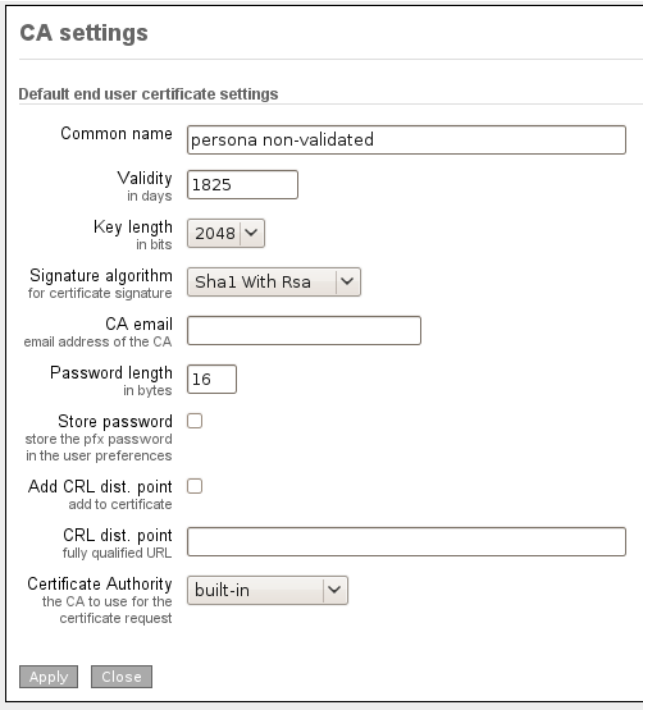


Figure 26: CA settings

**Signature algorithm** The algorithm used for signing the root and intermediate certificate. Windows versions prior to *XP-sp3* do not support *SHA256 With RSA* or better. If older Windows versions should be supported you are advised to use *SHA1 With RSA*. If support for older Windows versions is not required you are advised to select *SHA256 With RSA*.

## 12.2 CA settings

When the *Create new end-user certificate* page is opened the default dialog values are taken from the *CA settings*. The *CA settings* can be opened from the CA settings sub-menu (see figure 26). The settings *Common name*, *Validity*, *Key length* and *Signature algorithm* were already explained (see page 46)

**CA email** The sender email address used when sending certificates to end-users by email. Make sure that the CA email address is a valid email address. Because the email containing the encrypted certificate is sent by the gateway, the settings for the CA email user should be such that the email is not encrypted by the gateway (i.e., set encrypt mode of the CA user to *No Encryption*). If a certificate and key must be sent to an external recipient the CA email address must be set.

**Note:** don't forget to set *encrypt mode* of the CA user to *No Encryption*!

**Password length** The certificate creation page can automatically generate a password for the encrypted private key container (.pfx). The number of random bytes used to generate the password is set with the password length.

**Store password** This will be the default value for *Store password* for the *Create new end-user certificate* page. If checked the password for the last generated pfx file for a user will be stored in the *Last used pfx password* preferences of the user (see 4.2.8).

**Add CRL dist. Point** This is the default value for *Add CRL dist. Point* for the *Create new end-user certificate* page. If checked, and the CRL distribution point is set, the CRL distribution point value will be added to the newly generated end-user certificate.

**CRL dist. Point** The CRL distribution point added to the end-user certificate. The CRL distribution point is only added if *Add CRL dist. Point* is set. This is the default value for the *CRL distribution point* setting for the *Create new end-user certificate* page.

**Certificate Authority** The default CA used when a certificate is requested. The default *Certificate Authority* is for example used when a certificate is automatic requested for a sender when the sender does not yet have a valid signing certificate and *Auto request certificate* is enabled for the sender. See section 12.3 for more information on *Certificate Request Handlers* and Certificate Authorities.

## 12.3 Certificate Request Handlers

A certificate request handler is responsible for creating and/or retrieving a certificate and private key from internal or external CAs. The DJIGZO gateway contains a pluggable framework which allows new certificate request handlers to be registered. By default, there are currently three certificate request handlers available: *built-in*, *delayed built-in* and *Comodo*.

Some certificate request handlers should be setup before they can be used. Certificate request handlers can register a configuration page which can be used to setup the certificate request handler. The available certificate request handler configuration pages can be accessed using the *Request handlers* left-hand side menu on the *CA* page. The *Registered Certificate Request Handler configuration pages* page shows all registered certificate request handler configuration pages (see figure 27).

### 12.3.1 built-in certificate request handler

The built-in certificate request handler uses the built-in CA for creating new certificates. Certificates created with the built-in certificate request handler will be issued by the default selected CA (see section 12.5 for more information on selecting the default CA). The built-in certificate request handler creates


Figure 27: Registered Certificate Request Handler configuration pages

a private key and certificate instantly without any delay (i.e., a request for a certificate is synchronously handled).

### 12.3.2 delayed built-in certificate request handler

The delayed built-in certificate request handler is similar to the built-in certificate request handler. The only difference is that with the delayed certificate request handler, the request will be handled asynchronously by the background request handler thread.

If *Auto request certificate* (see page 22) is enabled and message throuhput should not be impacted when a new certificate is requested, it's better to use the delayed built-in certificate request handler because all certificate requests will then be handled asynchronously. If however a certificate should be used immediately when requested, the built-in certificate request handler should be used.

### 12.3.3 Comodo certificate request handler

The Comodo certificate request handler requests certificates from Comodo's Enterprise Public Key Infrastructure (EPKI). Comodo's EPKI is an outsourced Certificate Authority managed by Comodo. The main advantage of using certificates issued by Comodo is that these certificates are by default trusted by most systems (like Windows, Mac OS, Ubuntu). Comodo certificates are however not free. A valid EPKI account is required and the Comodo certificate request handler should be configured before Comodo certificates can be requested. See appendix F for more information on enabling the Comodo certificate request handler.

## 12.4 Create new end-user certificate

With the CA page a new end-user certificate can be created (see figure 28). Before an end-user certificate can be created a CA should be available. A warning will be shown if no CA is available or if a default CA is not selected. The general and Certificate subject settings have already been discussed.

**Email delivery** The email delivery settings are required when the newly created certificate and private key should be securely sent to an external recipient.

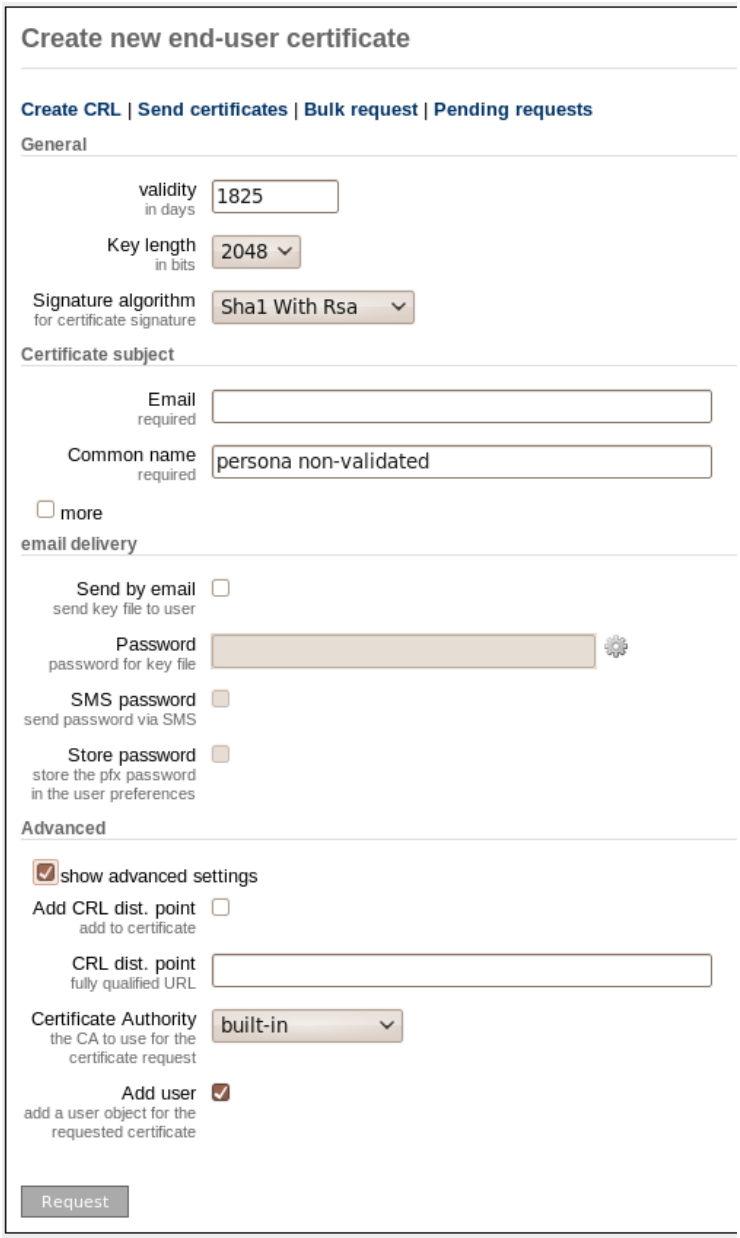


Figure 28: Create end-user certificate

If the *Send by email* checkbox is checked a password used for the protection of the certificate and private key should be set.

A password can be randomly generated by pressing the “gear” icon on the right hand side of the password edit field. If a password is manually set make sure that the password is strong enough. The password should be handed out to the recipient in a secure way i.e., it should not be emailed. For example send it by regular post or give the password in person. Alternatively the gateway can send the password via an SMS Text message.

**SMS password** If the *SMS password* checkbox is checked the password for the protected certificate and private key file will be sent to the recipient via an SMS Text message. This requires that the SMS gateway is correctly setup (see section 15) and that the recipients telephone number is added to the user settings of the recipient.

**Store password** If checked the password for the last generated pfx file for a user will be stored in the *Last used pfx password* preferences of the user (see 4.2.8).

**Advanced settings** With the advanced settings page the CRL distribution point for the certificate can be specified. A CRL distribution point should be a fully qualified URL pointing to the location where the latest CRL for the CA can be downloaded. If a CRL for the CA should be created and published make sure that the correct URL is specified. The URL cannot be changed after the certificate has been issued. The default value for the CRL distribution point is taken from the CA settings.

When the create button is clicked, and only if all the settings are valid, a new end-user certificate is created. If *Send by email* was checked the certificate and key will be password protected with the password and sent to the recipient by email. If *SMS password* was checked the password will be sent via an SMS Text message to the recipients telephone number. For a more thorough explanation of this procedure see the *S/MIME administration guide*.

## 12.5 Select default CA

There can be multiple CAs but only one can be active at the same time. Select the default CA with the *Select default CA* page (see figure 29).

## 12.6 Pending requests

Some certificate request handlers do not immediately issue a certificate (i.e., the certificate is asynchronously issued). For example the Comodo certificate request process requires several steps (see appendix F for more information). Certificate requests which are not immediately handled are stored in the *pending requests* store and handled asynchronously by a background thread (see figure 30).

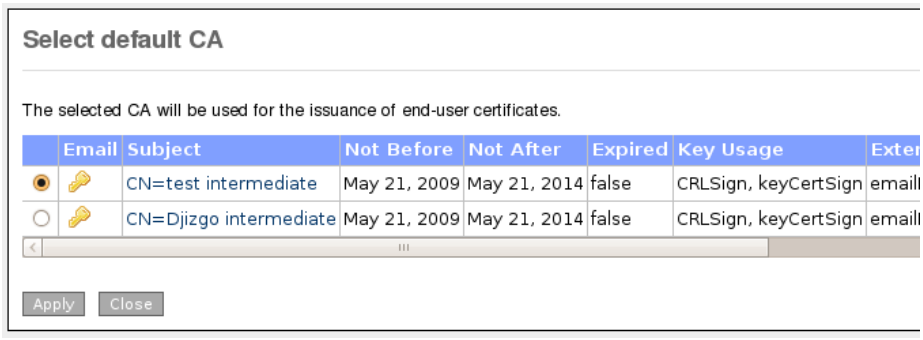


Figure 29: Select default CA

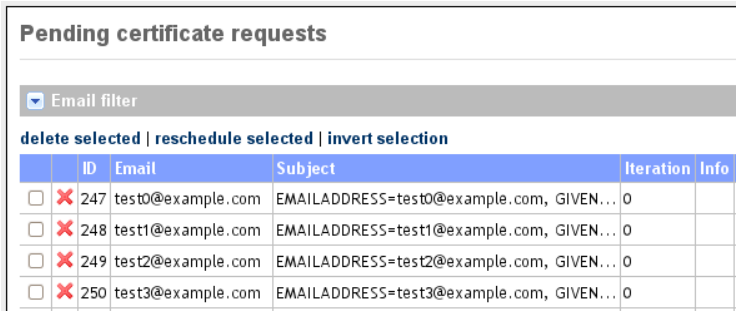


Figure 30: Pending certificate requests

## 12.7 Bulk request

With the bulk request option, multiple certificate requests can be issued at the same time. A comma separated text file containing the request details can be uploaded (see figure 31). The certificates will be requested using the selected certificate request handler. To make sure that the request details are correctly imported, a preview of the imported certificate requests will be shown after pressing *Request preview* (see figure 32). See appendix G for details of the comma separated file format.

**Note:** Certificate requests for email addresses for which there already is a valid signing certificate, will be skipped.

## 12.8 Create CRL

Sometimes a certificate should no longer be used even when it is not yet expired. A certificate revocation list (CRL) is used to revoke a specific certificate issued by a CA. With the *Create CRL* page a CRL for the internal CAs can be created or updated. Before the CRL can be created the CA, which will issue the CRL, must be selected (see figure 33).

After selecting the CA the *Create CRL* page is opened on which the certificates that should be revoked can be added to the list of revoked certificates. A brief explanation of the dialog fields:

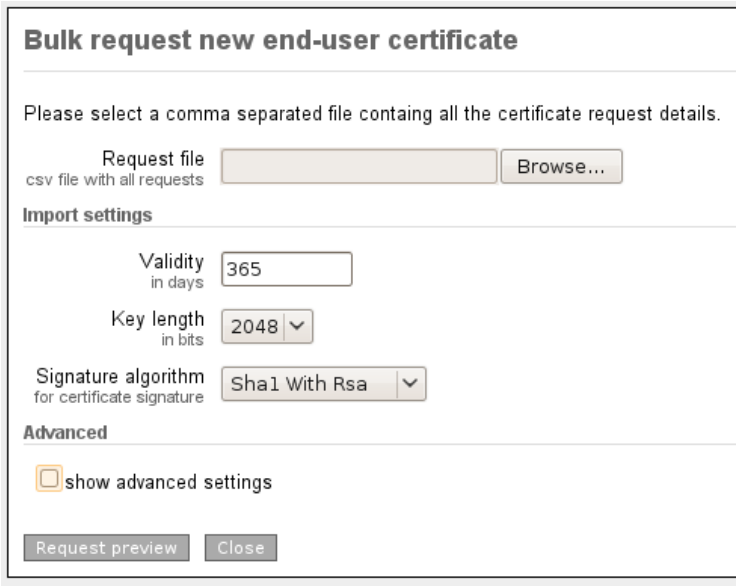


Figure 31: Bulk request

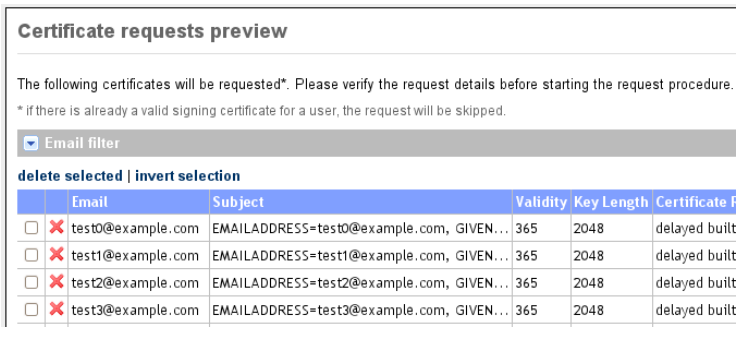


Figure 32: Certificate request preview

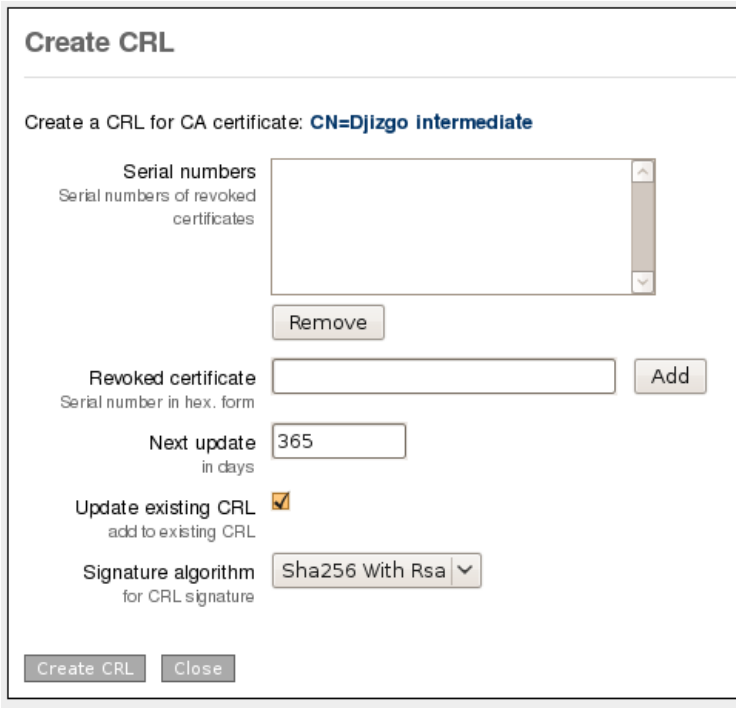


Figure 33: Create CRL

**Serial numbers** A certificate issued by a CA is uniquely identified by the serial number. The serial numbers list contains all the serial numbers that are about to be revoked.

**Revoked certificate** Add a new certificate to the list of certificates to be revoked by entering the serial number of the certificate (in hex form) in the *Revoked certificate* edit box and clicking the *Add* button.

**Next update** The *next update* is the date at which the CA claims it will issue a new CRL[16]. If the CA contains a CRL distribution point (see section 12.1) make sure that a new CRL is available and download-able from the CRL distribution point before the CRL expires. The next update is specified in days from the date of the CRL creation.

**Update existing CRL** If *update existing CRL* is selected an existing CRL is updated with the new serial numbers. The new CRL will contain the serial numbers of the old CRL and the new serial numbers. If *update existing CRL* is not selected a completely new CRL will be created with only the new serial numbers. It's best to always update an existing CRL because certificates that are previously revoked should remain revoked.

[16]The next update is the date at which a new CRL must be available. A CA is allowed to issue a new CRL before this date.

**Signature algorithm** The CRL will be signed by the issuing CA. A CRL should be signed to make it possible for external parties to check whether the CRL is a valid CRL and is issued the CA. Windows versions prior to *XP-sp3* do not support *SHA256 With RSA* or better. If older Windows versions should be supported you are advised to use *SHA1 With RSA*. If support for older Windows versions is not required you are advised to select *SHA256 With RSA*.

Clicking the *Create CRL* button will create the new CRL. The new CRL will be automatically added to the CRL store (see section 10). If the CA specifies a CRL distribution point the CRL should be published. Download the CRL from the CRL store and upload it to the CRL distribution point URL.

## 12.9 Send certificates

Sometimes end-users require a copy of their certificates (and private keys). For example they experienced a system crash and had to completely reinstall the system (and forgot to make a backup).

The *Send certificates* page can be used to send a new copy of the certificate and private key to an external user. This is also known as *key escrow*. Clicking *Send certificates* opens the *Send selected certificates to recipient* page (see figure 34). Sending CA certificates by email is not allowed. This is done to prevent accidental leakage of CA certificates.

**Password** This has already been explained. See section 12.4.

**SMS password** This has already been explained. See section 12.4.

**Email** This is the email address of the recipient to which the certificate(s) and key(s) will be sent.

**Allow mismatch** By default, email addresses in the certificate must match the email address of the recipient. However, there are situations where the certificate and private key should be sent to to a different email address. By checking *Allow mismatch* the certificates and keys can be sent to non matching email addresses. The *Allow mismatch* check is added to prevent any leakage of certificate(s) and key(s) because of an accidental mistype of the recipients email address.

# 13 PDF encryption

The problem with S/MIME is that it requires the recipient to use an S/MIME capable email client[17] 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

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

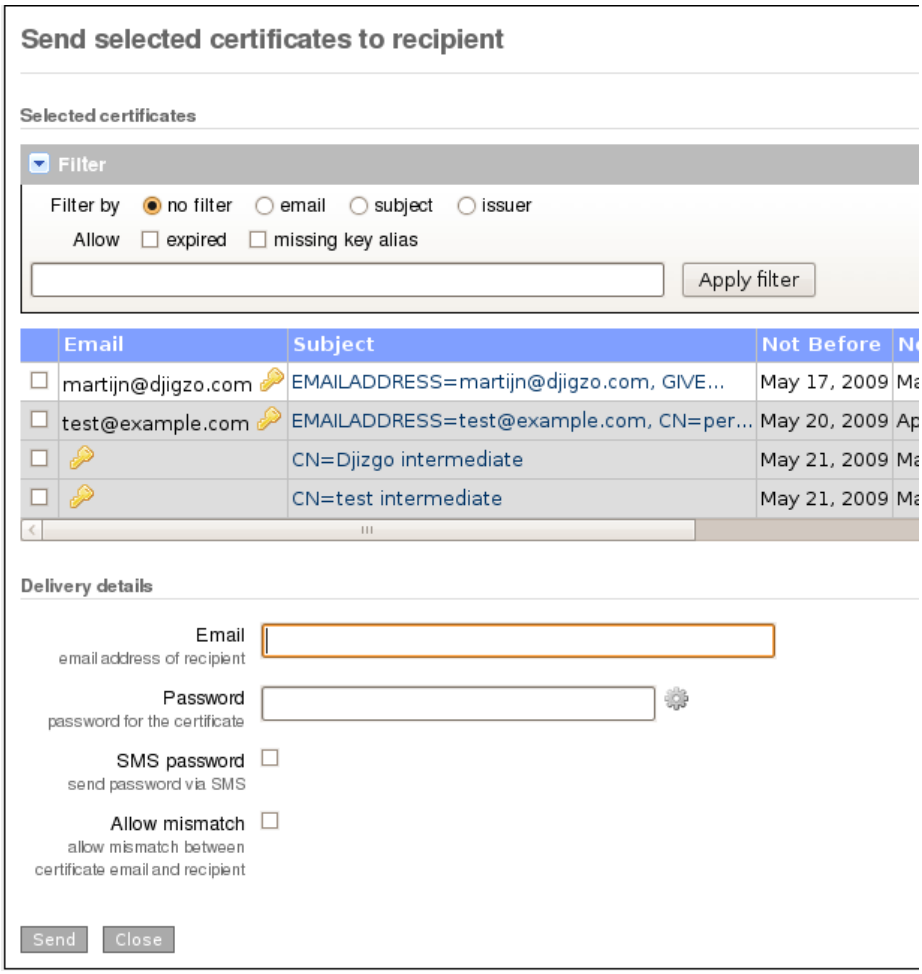


Figure 34: Send selected certificates to recipient

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[18]. 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 (see section 6 for the templates). There are different options on how to password encrypt the PDF:

**(a)** The PDF can be encrypted using a pre-defined static password.

**(b)** The PDF can be encrypted using randomly generated password. The password will then be sent by SMS Text to the recipient.

**(c)** The PDF can be encrypted using randomly generated password. The password will be sent back to the sender of the message.

**(d)** The PDF can be encrypted using a one time password (OTP) algorithm.

The four password options will be briefly explained. For more details see the PDF encryption guide.

**Static password** To use a pre-defined and static password for PDF encryption, the password for the recipient should be set. To make sure that the password will always be valid (i.e., that it will never expire), either set the *Validity interval* to -1 or make sure that the *Date set* password setting is not set.

**Send password by SMS Text** If setup correctly, the system can automatically send the generated password to the recipient via an SMS Text message. This requires that the SMS *Phone number* is set for the recipient. Alternatively, if the user is allowed to add a telephone number to the subject (see page 27), the mobile number can be specified on the subject of the email. Figure 35 shows the complete PDF encryption process when using the SMS option.

**Send to originator** If the *Send to originator* option is enabled, the generated password(s) will be sent back to the sender of the message. The sender is then responsible for securely delivering the passwords to the recipients.

**One time password (OTP)** If the one time password option is enabled, the PDF password will be generated using a one time password algorithm. The recipient can login to the portal to retrieve the PDF password.

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

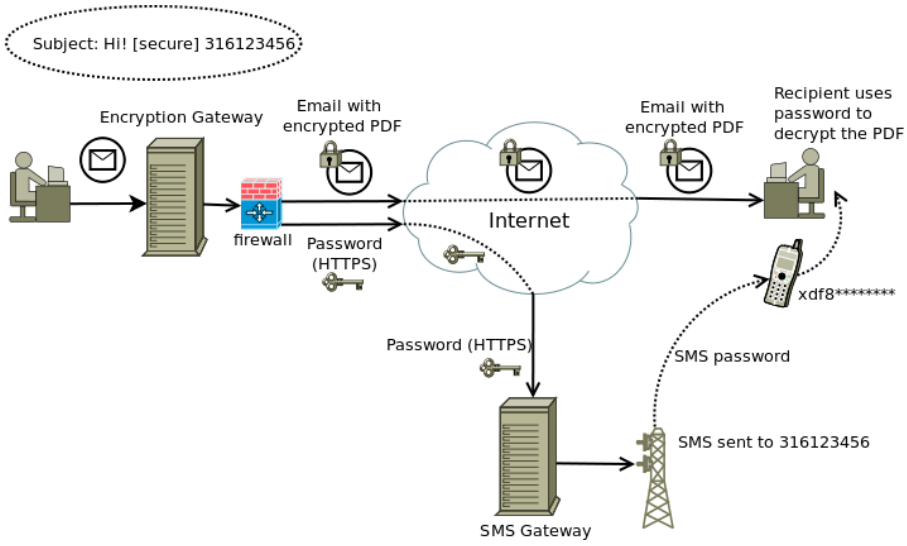


Figure 35: PDF encryption

## 13.1 Encrypted PDF message

The recipients receives a message with a standard message (based on a template) with the encrypted PDF attached (see figure 36 for an example opened in Gmail). When the PDF is opened, the PDF reader asks for the password (see figure 37). Only after entering the correct password will the PDF content be shown. The PDF is formatted to make it look like a normal email message. The attachments can be accessed from the attachment pane at the bottom (see figure 38).

## 13.2 Replying

A recipient can reply to the encrypted PDF message by clicking the *Reply* link (see figure 36). An on-line portal, via a secure *HTTPS* connection, will be opened (see figure 39). The reply URL in the PDF is equal to the *Reply URL* parameter at the time the encrypted PDF was created (see paragraph 4.2.6). The user can now enter a message body and add attachments (by default maximum 3). The reply will be sent via the DJIGZO server. Because the reply is sent via the DJIGZO server it can be encrypted as well (see *Reply Sender* at page 24).

# 14 DLP

Data Leak Prevention (DLP) is a feature that prevents certain information to leave the organization via email. What information this is, is defined in the configuration of the DLP system. Typically, it includes credit card numbers, bank account numbers, excessive amounts of email addresses or other personal information in one email message, etc. DLP is implemented as a filter on out-


Figure 36: PDF encrypted message

going email.

DLP can monitor email at various levels:

- email body content
- email headers
- email attachments of various types
- nested attachments of various types

DJIGZO DLP currently filters email bodies, attachments and nested attachements of type text, html, xml and other text-based formats. Filtering attachments of type pdf, doc, xls etc. will be part of a future offering of DJIGZO DLP. For more information about setting up the DLP functionality, see the separate *DLP setup guide*.

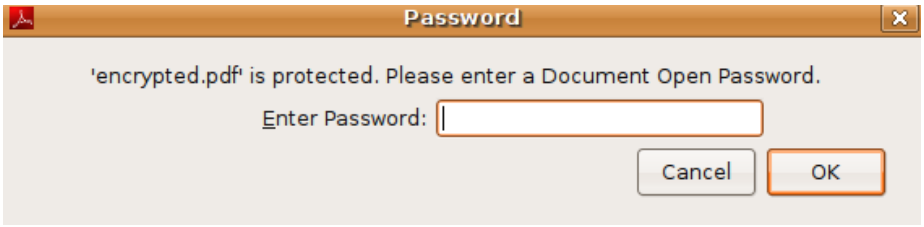


Figure 37: PDF encrypted message

# 15 SMS gateway

DJIGZO contains an SMS gateway which is used for sending generated passwords via SMS Text messages. The SMS gateway can use different SMS transports for the delivery of SMS Text messages[19]. The default SMS transport is set to Clickatell (see http://www.clickatell.com for more information). SMS Text messages are sent via a secure HTTPS connection to Clickatell. When an SMS Text message is sent, it is queued for delivery until the message has been delivered with the active SMS transport (see figure 40). To test the SMS gateway an SMS Text message can be manually added with *Add SMS*.

## 15.1 Clickatell transport

The default SMS transport is the *Clickatell transport*. This transport forwards all the SMS Text messages to an external SMS gateway (using a secure HTTPS connection). A Clickatell account must be created and configured before any SMS Text messages can be sent. See see http://www.clickatell.com for more information about the sign-up process.

During the sign up process a HTTP connection must be added [20] (leave the *Callback* parameters empty). The connection has an associated *API ID* which is required for the Clickatell transport. Open the Clickatell transport configuration page by opening the *SMS* page and clicking the *Clickatell settings* left-hand side sub-menu (see figure 41). The first three settings: *API id*, *User* and *Password* are mandatory. The *From* parameter can be set to the sender of the SMS Text message (i.e., the telephone number of the sender) but only after the telephone number has been approved by Clickatell.

Clickatell uses pre-paid message credits. To check how many credits are left (and for testing the login credentials), click *update balance*.

**Note:** newly entered transport settings are only used after the changes have been applied. Before clicking *Update balance*, make sure all changes are applied.

[19] Currently only Clickatell and Gnokii (direct connection to Nokia phones) are supported.
[20] See the Clickatell *HTTP API Specification v.2.x.x* document for more information

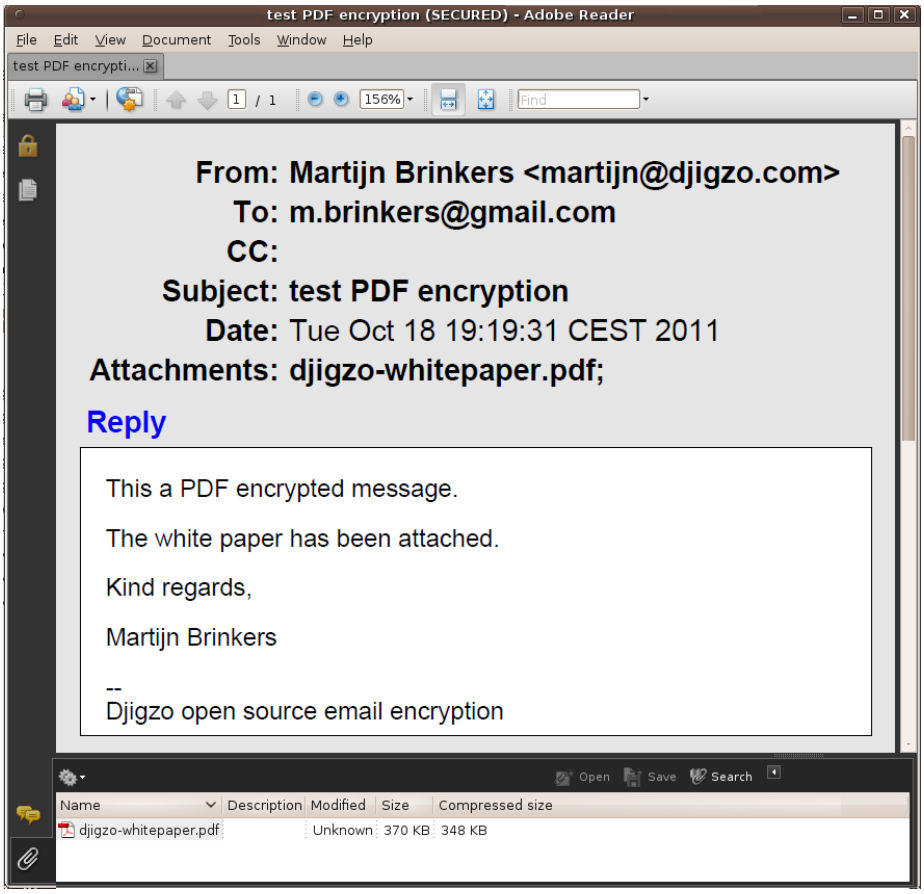


Figure 38: PDF decrypted


Figure 39: PDF reply

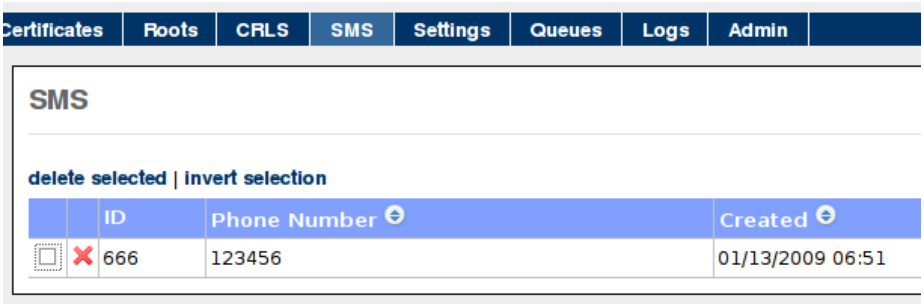


Figure 40: SMS gateway

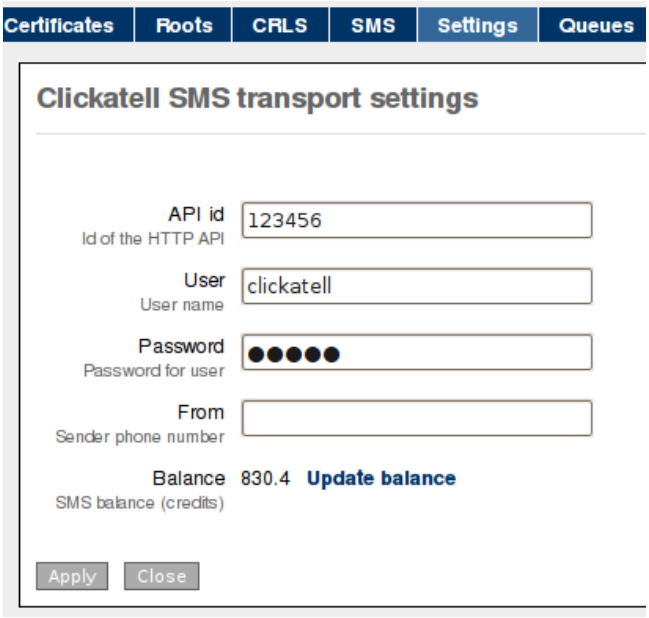


Figure 41: Clickatell settings

# 16 Mail Queues

Postfix is used as the MTA for sending and receiving of email to internal and external recipients. Internally a Java based SMTP server is used for message processing (the *Mail Processing Agent*). The MTA and MPA store all mail into different mail queues.

The mail queues of the MTA and MPA can be viewed and managed using the *Queues* page (see figure 42). There are five different mail queues: *MTA*, *MPA outgoing*, *MPA error*, *MPA spool* and *MPA respool*.

**MTA queue** With the MTA queue page messages on the MTA queue can be removed, put on hold, viewed etc.

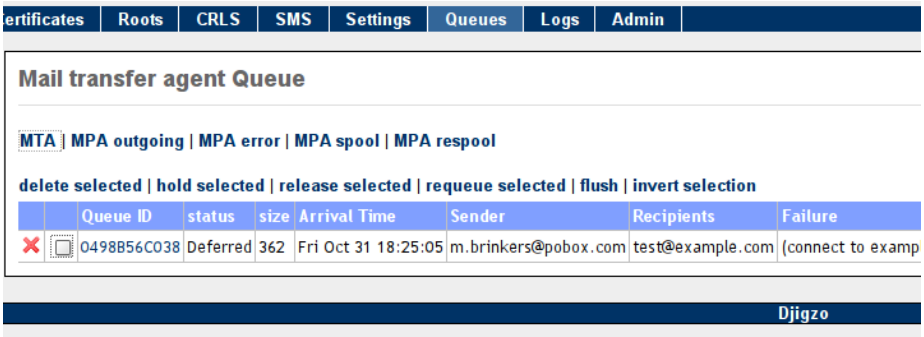


Figure 42: Mail Queues

**MPA queues** The MPA contains four queues: *MPA outgoing*, *MPA error*, *MPA spool* and *MPA respool*. Normally the error and respool queue should be empty. The other two queues should only contain email for a short period while the email is processed. Processed email is sent to the MTA for further delivery.

# 17 Logging

The *Logs* page is used to view the MTA and MPA logs. A keyword filter can be set to view only a subset of all the log entries (see figure 43). The search keyword is highlighted in yellow. Every email is tagged with a unique *Mail ID* (shown in a green color). This makes it easier to track an email while the email is being processed. Color coding of certain log elements is used to make it easier to spot certain details (for example an email address is shown in a blue color, an error in red).

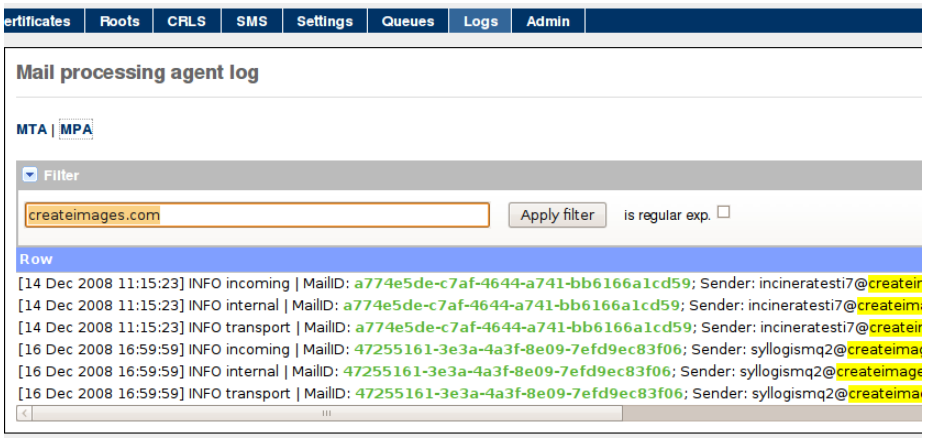


Figure 43: Logging

# 18 Administrators

Multiple administrators, each with a different set of roles, can be registered. The *Admin* page shows a list of all the registered administrators (see figure 44). A new administrator can be added by clicking *Add admin* on the left-hand side menu. This will open the *Adding new administrator* page (see figure 45).

## 18.1 Roles

An administrator can have any of the following roles:

- ROLE_LOGIN
- ROLE_ADMIN

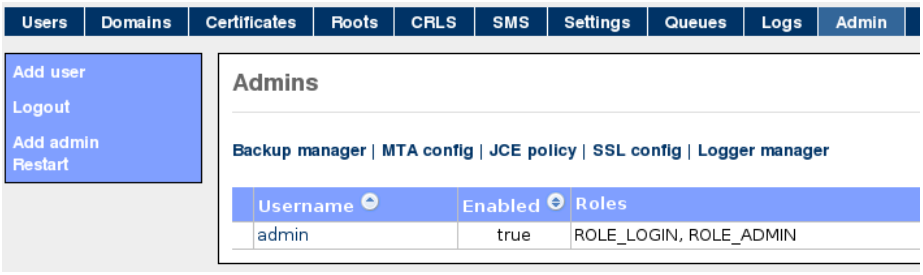


Figure 44: Administrators

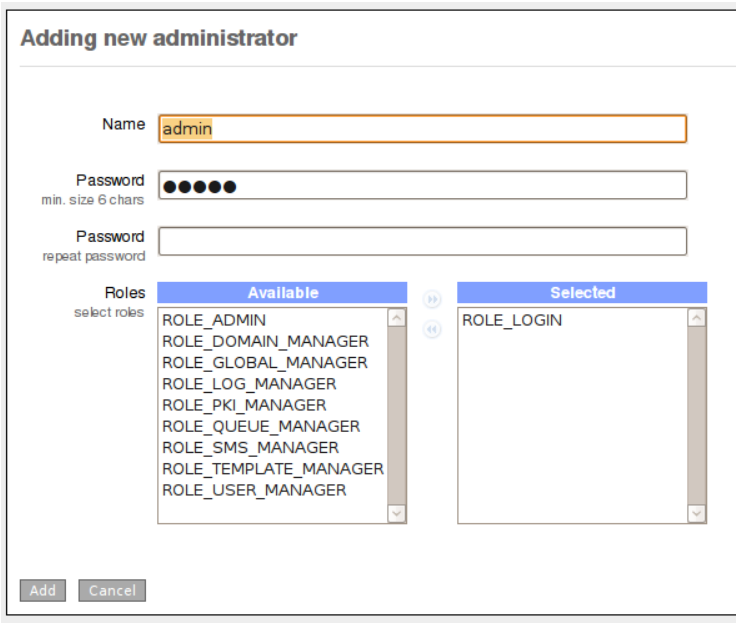


Figure 45: Add new administrator

- ROLE_USER_MANAGER
- ROLE_DOMAIN_MANAGER
- ROLE_GLOBAL_MANAGER
- ROLE_LOG_MANAGER
- ROLE_PKI_MANAGER
- ROLE_QUEUE_MANAGER
- ROLE_SMS_MANAGER
- ROLE_TEMPLATE_MANAGER
- ROLE_DLP_MANAGER
- ROLE_QUARANTINE_MANAGER
- ROLE_MOBILE_MANAGER
- ROLE_PORTAL_MANAGER

**ROLE_LOGIN** This is a required role. An administrator with just ROLE_LOGIN is only allowed to view a few basic settings.

**ROLE_ADMIN** This role is similar to having all roles (i.e., an administrator with ROLE_ADMIN is allowed to do anything).

**ROLE_USER_MANAGER** An administrator with this role is allowed to *add users*, *delete users*, *edit users*, *select user certificates* and *select user signing certificate*.

**ROLE_DOMAIN_MANAGER** An administrator with this role is allowed to *add domains*, *delete domains*, *edit domains*, *select domain certificates* and *select domain signing certificate*.

**ROLE_GLOBAL_MANAGER** An administrator with this role is allowed to edit the global settings.

**ROLE_LOG_MANAGER** An administrator with this role is allowed to view the log files.

**ROLE_PKI_MANAGER** An administrator with this role is allowed to *import certificates*, *delete certificates*, *import keys*, *download keys*, *import CRLs*, *delete CRLs*, *update CRL store* and *manage the CA*.

**ROLE_QUEUE_MANAGER** An administrator with this role is allowed to manage the queues (with the exception of the quarantine queue).

**ROLE_SMS_MANAGER** An administrator with this role is allowed to manage the SMS gateway.

**ROLE_TEMPLATE_MANAGER** An administrator with this role is allowed to edit templates.

**ROLE_DLP_MANAGER** An administrator with this role is allowed to manage all DLP settings like adding new DLP rules, removing DLP rules, managing the quarantine queue. In order to view the quarantine queue, the administrator also requires the ROLE_QUEUE_MANAGER role.

**ROLE_QUARANTINE_MANAGER** An administrator with this role is allowed to manage the quarantine queue. In order to view the quarantine queue, the administrator also requires the ROLE_QUEUE_MANAGER role.

# 19 Backup manager

## 19.1 System backup

The backup manager can be used to backup and restore all the relevant system settings (including the certificates, keys and MTA settings). A backup can be created and downloaded to the administrators computer or a backup can be stored on a remote SAMBA share (see figure 46). A backup can be automatically initiated at set intervals and stored (encrypted or non encrypted) on a remote SAMBA share. A backup can be password encrypted. If no password is specified the backup will not be encrypted.

**Warning:** restoring a backup will overwrite all local settings and cannot be undone. The system will be restarted after the restore.

## 19.2 Backup configuration

The backup configuration page is used to configure the remote SAMBA share and configure the automatic backup (see figure 47).

### 19.2.1 SMB share settings

The SMB share settings specify which remote SAMBA share should be used for remote backups (automatic backups can only be stored on a remote share). The remote share can be any server that supports the SMB protocol (for example Microsoft Windows Network or SAMBA). *Test connection* can be used to test whether the specified share can be accessed with the provided settings and credentials.

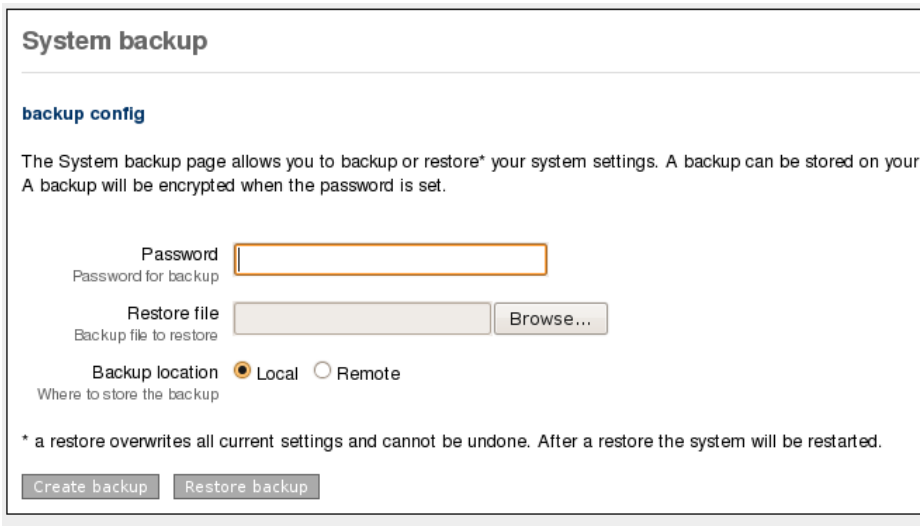


Figure 46: System Backup

### 19.2.2 Automatic backup

**Enabled** Remote backups can be automatically initiated at set intervals. To enable automatic backups the *enabled* checkbox should be checked.

**Cron expression** The cron expression[21] determines at which intervals a backup will be started. A restart is required after changing the cron expression. The default cron expression **0 0 2 * * ?** automatically starts a backup every night at 2 o'clock (see Appendix D for more cron expression examples).

**Password** The password with which the backup will be encrypted.

### 19.2.3 Other

**Strategy** The filename of the backup is determined by the *strategy*. Choose between *day of week*, *day of month*, *day of year* and *timestamp*. *Day of week* uses the day of the week as a filename postfix (1-7). *Day of month* uses the day number as a filename postfix (1-31). *Day of year* uses the day (1-365) as a filename postfix. *Timestamp* uses a filename based on the number of milliseconds since January 1, 1970 UTC.

## 20 SSL certificate manager

The gateway requires a HTTPS connection for the Web admin and PDF reply page. During installation, a default SSL certificate has been installed. It is therefore advised to install a new SSL certificate after installation. The *SSL*

[21] For more info on the cron trigger format see http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html

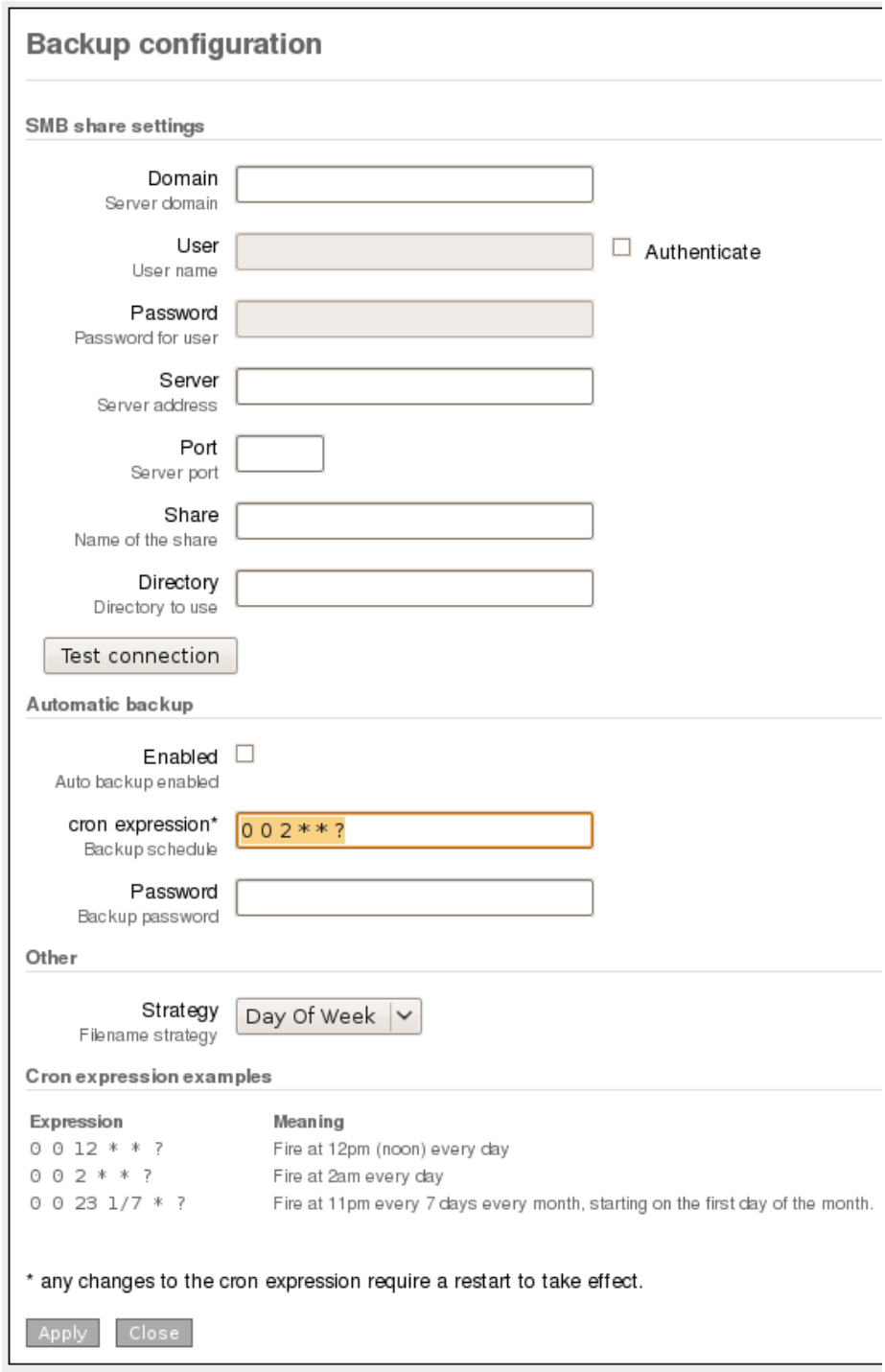


Figure 47: Backup configuration

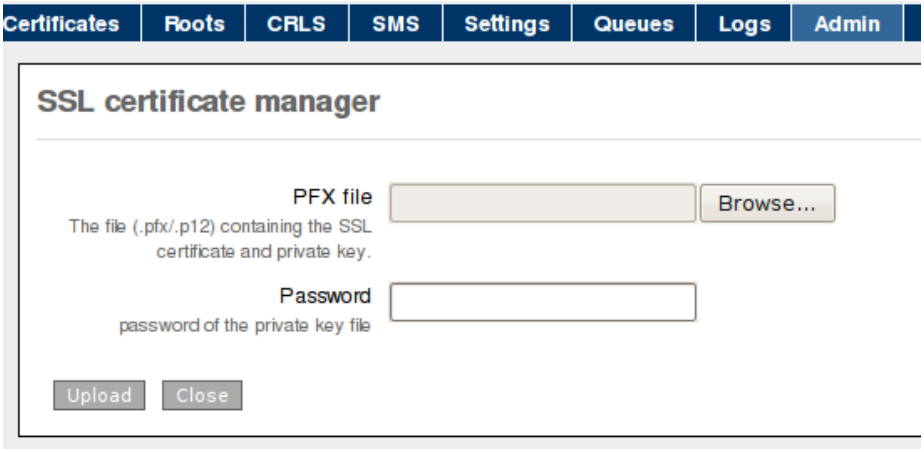


Figure 48: SSL certificate manager

*certificate manager* page can be used to install a new SSL certificate (see figure 48). A password protected *PKCS#12* file (.pfx or .p12) with the SSL certificate and private key suitable for SSL should be uploaded. After installation of the SSL certificate, the system should be restarted. The system can be restarted by clicking *Restart* on the *SSL certificate manager* page or by opening the *Admin* menu and selecting *Restart* on the left hand side menu[22].

**Note:** if the PDF reply functionality is used, and the PDF reply page is externally accessible you are advised to install an SSL certificate which is trusted by all browsers (for example use a Verisign or StartSSL certificate).

# 21 Proxy

For CRL downloads and sending of SMS Text messages, the gateway needs access to external resources. If external resources should be accessed via a proxy, the proxy should be configured. The DJIGZO proxy client only supports HTTP(s) and *NTLMv1* (*NTLMv2* is not supported). The proxy can be configured by opening the *Admin* page and selecting *Proxy config* (see figure 49).

## 21.1 Fetchmail

Fetchmail can be used to retrieve email from remote *POP3*, *IMAP* servers and forward the email to different email addresses via the DJIGZO gateway. DJIGZO allows the administrator to configure Fetchmail via a web based configuration page.

[22] Alternatively if the Virtual Appliance is used the system can be restarted by selecting *Restart services* from the Virtual Appliance Console.

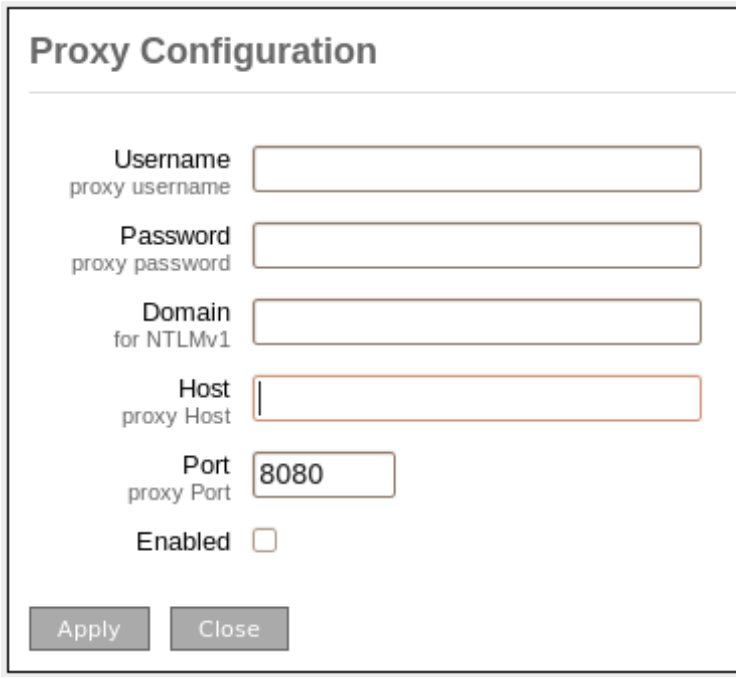


Figure 49: Proxy configuration

**Note:** Fetchmail support is only available on the Virtual Appliance[23]. Fetchmail is not enabled by default and should be manually enabled from the Virtual Appliance console. See the Virtual Appliance guide on how to enable Fetchmail support.

## 21.2 Fetchmail manager

When Fetchmail support is enabled a *Fetchmail manager* option is added to the admins menu (see figure 50). With the *Fetchmail manager* page new servers can be added which will be periodically polled for new messages (see figure 51).

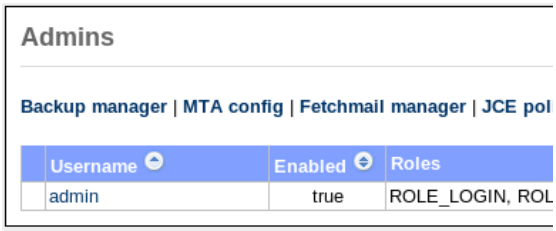


Figure 50: Admins fetchmail options

[23] Contact us for instructions on how to enable Fetchmail support for a non-appliance version of DJIGZO.

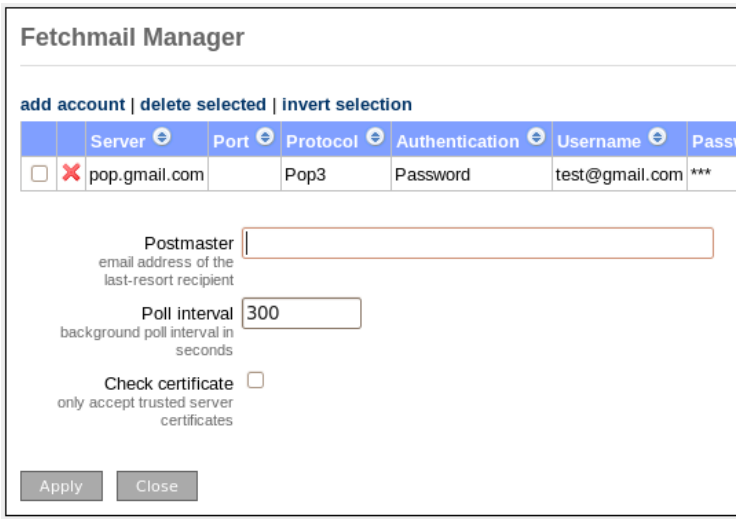


Figure 51: Fetchmail manager

### 21.2.1 Global settings

Fetchmail manager contains three global settings relevant for all polled servers: *Postmaster*, *Poll interval* and *Check certificate*.

**Postmaster** If email cannot be forwarded an error message will be sent to the postmaster email address.

**Poll interval** The number of seconds between consecutive checks for new email. For *IMAP* accounts with *IDLE* support a new message is instantly detected and forwarded (also known as push mail). The *Poll interval* should not be too low to prevent flooding of the remote server.

**Check certificate** If checked, Fetchmail checks whether the server certificate is trusted and is issued by a locally trusted CA.

### 21.2.2 Applying changes

When the *Apply* button is pressed the Fetchmail configuration will be updated and Fetchmail will be restarted.

### 21.2.3 Adding a new account

New accounts to be polled and forwarded can be added by clicking *add account* (see figure 51). The page *Fetchmail Add Account to Poll* will be opened (see figure 52). The account settings will be briefly explained.

**Server** The server that is being polled. This can be a fully qualified domain name or an IP address.

**Port** The port the server being polled listens on. If left empty the default port for the protocol will be used.

**Protocol** The protocol of the server being polled (*POP3*, *IMAP* etc.).

**Authentication** The authentication protocol the server being polled uses. With *Any*, Fetchmail tries each available method consecutively until a successful login.

**Principal** The Kerberos principal (only relevant for IMAP and kerberos).

**Username** The username of the remote account.

**Password** The password of to the user account.

**Folder** The remote folder to query.

**UIDL** Force client-side tracking of new messages. Should be used in conjunction with *keep*. This setting is only relevant for *POP3*.

**SSL** Connect to the remote server via SSL.

**Keep** If *Keep* is selected, seen messages are not deleted from the remote server (if *Keep* is used with *POP3*, *UIDL* should also be selected). If possible, seen message should be deleted to make sure a message is never delivered twice. It is therefore advised to leave *Keep* unchecked.

**Idle** If selected, Fetchmail waits for new messages after each poll (IMAP only). With *Idle* new messages are instantly forwarded.

**Forward To** The email address to forward newly polled messages to.

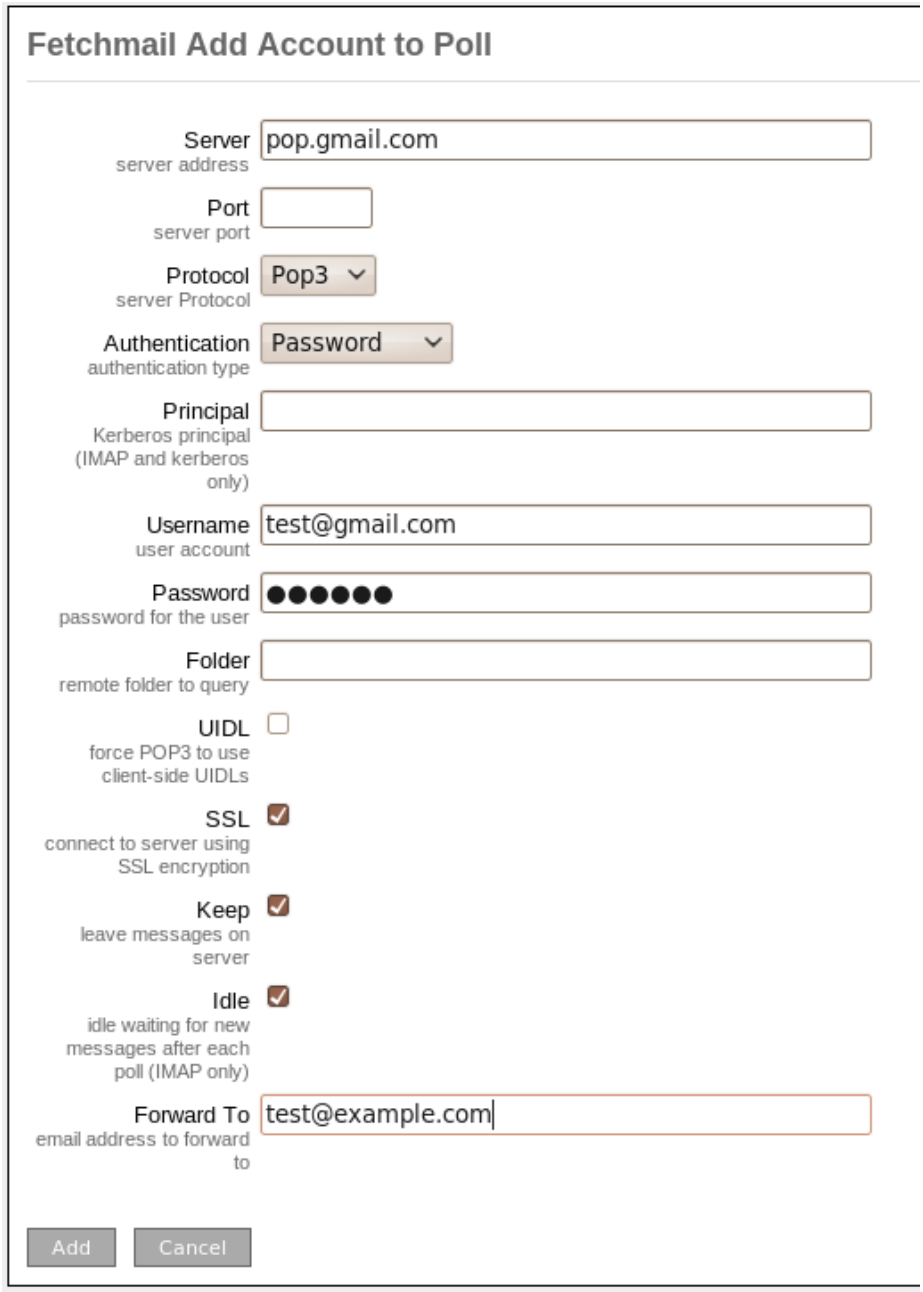


Figure 52: Fetchmail new account

# A SMTP HELO/EHLO name

The SMTP helo/ehlo name is the hostname the SMTP server sends with the SMTP EHLO or HELO command (the DJIGZO gateway uses the HELO or EHLO command when sending email to another email server). Some email servers check whether the helo/ehlo name is equal to the reverse IP lookup (with a reverse IP lookup the name is retrieved that belongs to the IP address) and if the names do not match they will flag the email as spam.

If the DJIGZO gateway is used to directly send email to external recipients (i.e., outgoing email is not relayed through an external relay host) the gateway should be setup with the correct helo/ehlo. The SMTP helo name should be equal to the reverse lookup of the external IP address.

If the external IP address is not known and the DJIGZO gateway uses the same IP address as the web browser, the external IP address and hostname (reverse IP) can be retrieved using on-line services like http://www.whatismyipaddress.com. The IP address shown is the external IP address. The shown hostname (the reverse IP lookup) should be used for the SMTP helo name. If the hostname of the DJIGZO gateway is set to the external hostname, the SMTP helo name can be left empty because the SMTP helo name will then be equal to the gateway hostname.

**Checking the HELO/EHLO name** whether the HELO/EHLO name is correctly setup can be checked using the helo check services from http://cbl.abuseat.org/helocheck.html by sending an email to *helocheck@cbl.abuseat.org*. The email will be immediately bounced. The bounce message contains the HELO name used by the gateway.

```
<helocheck@cbl.abuseat.org>: host mail-in.cbl.abuseat.org said:
    550 HELO for IP 82.94.189.170 was "secure.djigzo.com"
    (in reply to RCPT TO command)
```

Where 82.94.189.170 is the external IP address of the gateway (IP address will be different for every server) and *secure.djigzo.com* was the HELO name used by the gateway.

# B SASL authentication

SMTP client authentication is not enabled by default. SMTP client authentication can be enabled by adding the following lines to the postfix main config using the *MTA raw config* page (see 3.4):[24]

```
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/smtp_client_passwd
smtp_sasl_type = cyrus
```

New SASL credentials for an SMTP host can be added by clicking *add password*. This opens the *Add SASL password* page (see figure 53). If *mx* is

[24]The main config that comes with DJIGZO already contain these lines. They are however commented out.

selected the MX-records of the server are used instead of the IP address of the server (A-record). In most cases the IP address of the server should be used and *mx* should therefore not be selected.

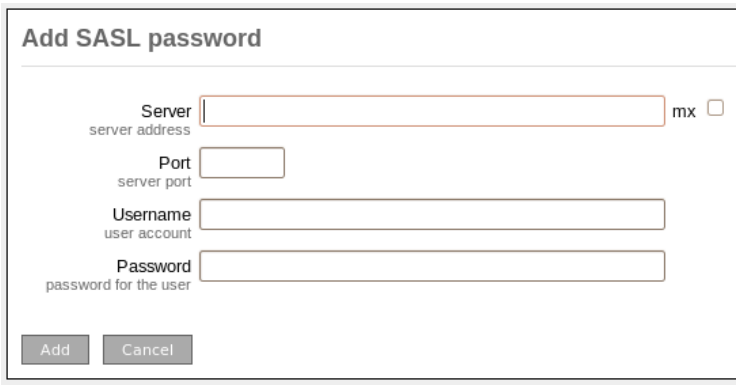


Figure 53: SASL add password

**Gmail example:** as an example the following part will explain how to use the Gmail SMTP servers as an external relay host (i.e., email sent to external recipients will be relayed via Gmail). For SMTP authentication Gmail requires a TLS protected connection. TLS and sasl authentication should therefore be enabled by adding the following lines to the postfix main config file using the *MTA raw config* page (see 3.4):

```
smtp_tls_security_level = may
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/smtp_client_passwd
smtp_sasl_type = cyrus
smtp_tls_CApath = /etc/postfix/certs/
smtp_sasl_security_options =
```

The *External relay host* should be set to *smtp.gmail.com* and the port to *587* (see figure 54).


Figure 54: Gmail external relay host

The SASL password for server *smtp.gmail.com* and port *587* should be set. The username should be set to the Gmail username (the username should include @gmail.com) and password (see figure 55).

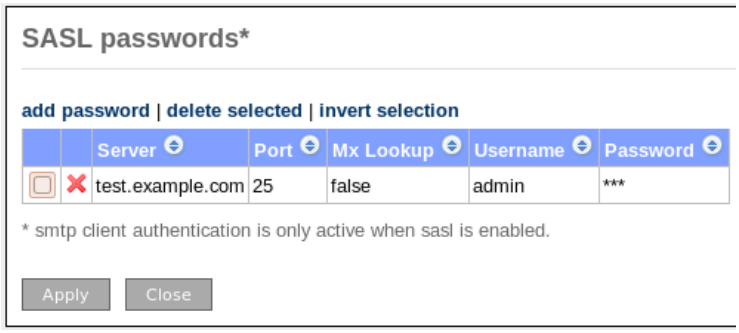


Figure 55: Gmail SASL password

# C Content and virus scanning

A content scanner can be used in combination with DJIGZO to selectively force encryption when a message contains certain keywords (for example a *Social Security Number*). A typical setup of a content scanner and an encryption gateway can be see in figure 56.

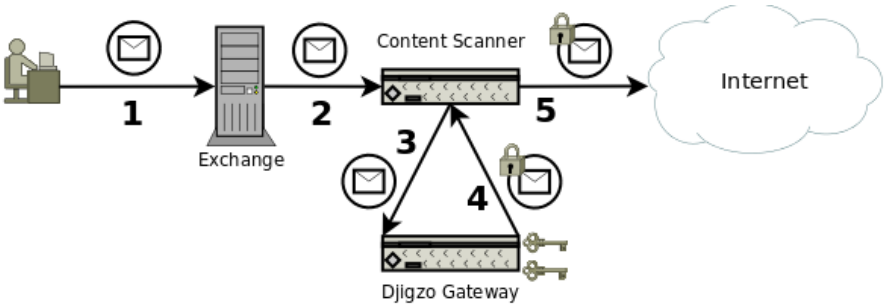


Figure 56: Content scanning

**DJIGZO with content scanner:**

1. User sends unencrypted message.
2. Exchange forwards message to content scanner.
3. Content scanner detects that the message must be encrypted (for example the message contains a SSN).
4. DJIGZO gateway encrypts the message with S/MIME or PDF.
5. Content scanner sends the encrypted message to the recipient.

Most organizations need to scan all incoming and outgoing email for viruses. A typical setup of an encryption gateway and a virus scanner can be see in figure 57.

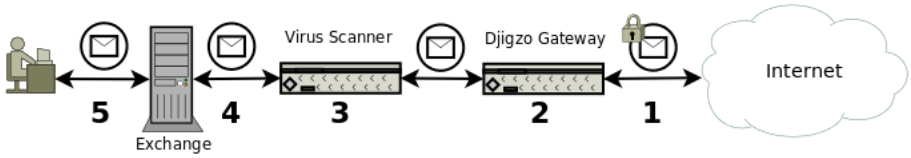


Figure 57: Virus scanning

**DJIGZO with virus scanning:**

1. S/MIME encrypted message is received from the Internet.
2. DJIGZO gateway decrypts the message.
3. The decrypted message is scanned for viruses.
4. After virus scanning the message is forwarded to Exchange.
5. User reads the message.

A more advanced setup is required when email must be encrypted on the desktop yet all outgoing email must be virus scanned because of corporate policies. Figure 57 shows how encrypted outgoing email can be virus scanned.

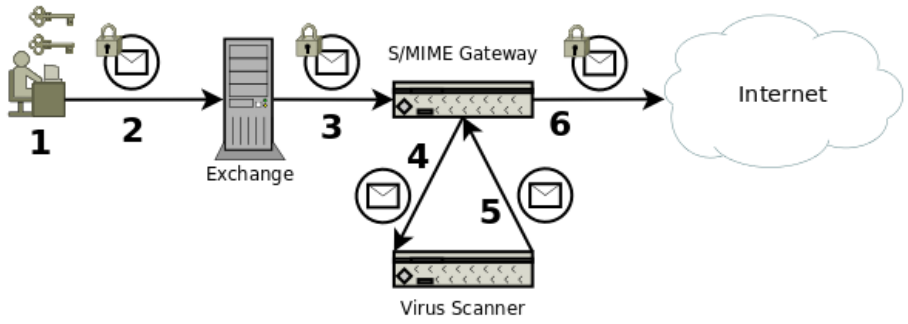


Figure 58: Virus scanning with desktop encryption

**DJIGZO with desktop encryption and virus scanning:**

1. User encrypts message with personal and receivers certificate.
2. S/MIME encrypted message is sent to Exchange.
3. Exchange sends S/MIME encrypted message to DJIGZO gateway.
4. DJIGZO gateway decrypts the message with the senders private key (the gateway stores a copy of the key) and sends the decrypted message to the virus scanner.
5. Virus scanner scans the message and if clean it will be sent back to the DJIGZO gateway.

6. The DJIGZO gateway re-encrypts the message and sends the message to the external recipient.

# D Cron Expressions

The following cron examples are taken from http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html.

| Expression | Meaning |
|---|---|
| 0 0 12 * * ? | Fire at 12pm (noon) every day |
| 0 15 10 ? * * | Fire at 10:15am every day |
| 0 10,44 14 ? 3 WED | Fire at 2:10pm and at 2:44pm every Wednesday in March. |
| 0 15 10 15 * ? | Fire at 10:15am on the 15th day of every month |
| 0 15 10 L * ? | Fire at 10:15am on the last day of every month |

For more cron examples see the *Quartz Scheduler* website.

# E MPA mail flow

The following flow-charts will show exactly how email is processed by DJIGZO.

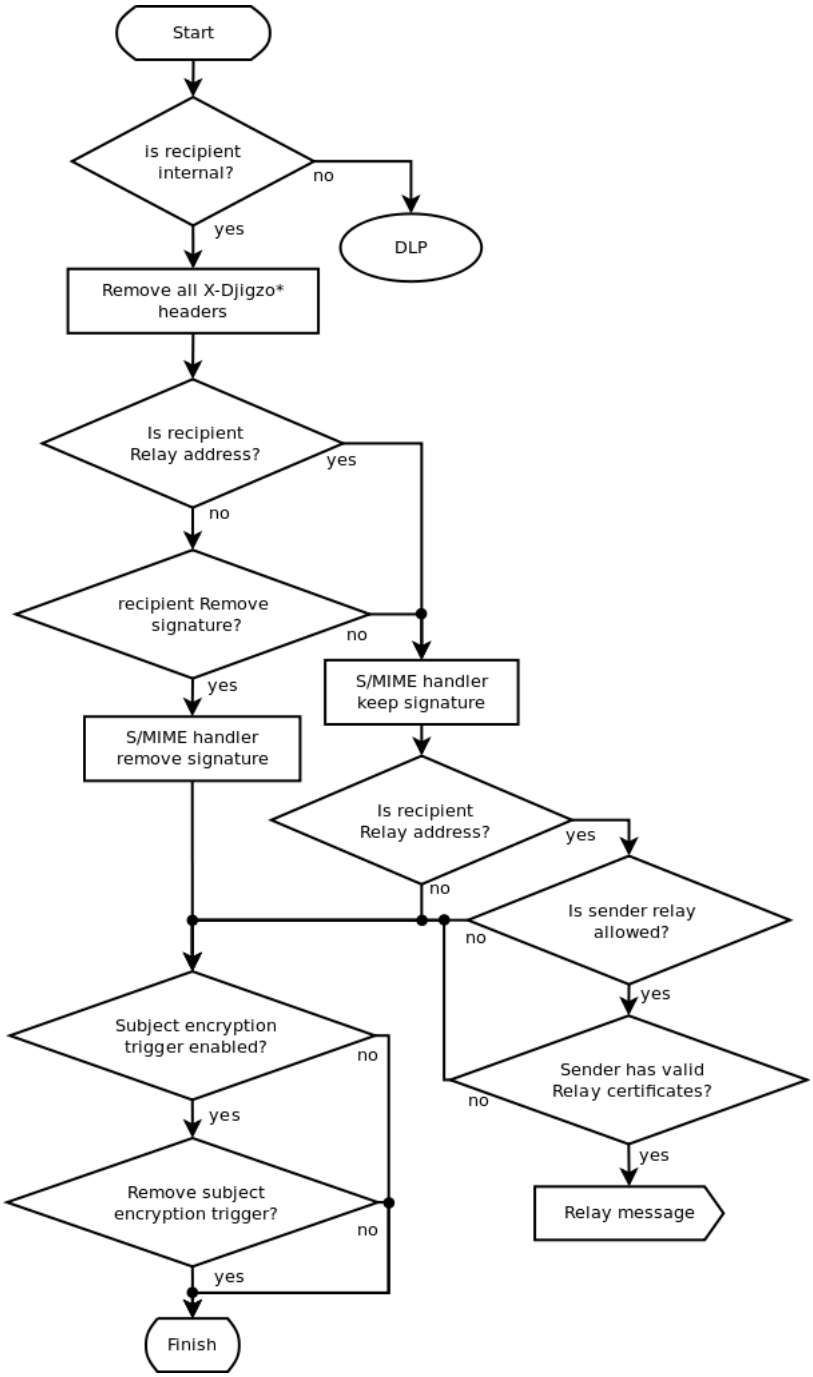


Figure 59: Start

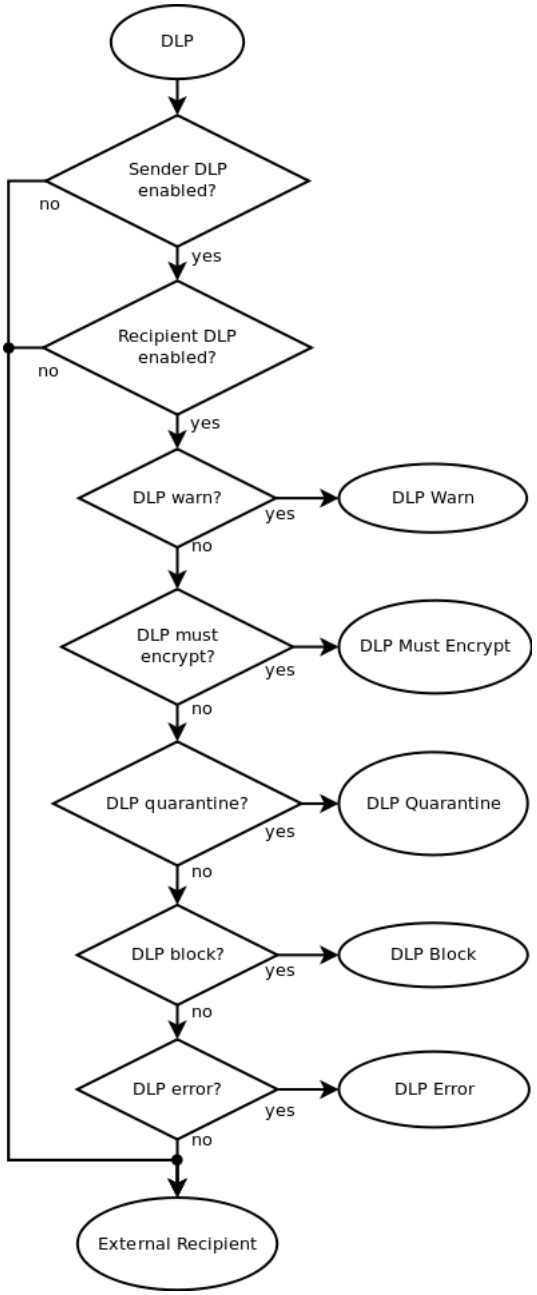


Figure 60: DLP

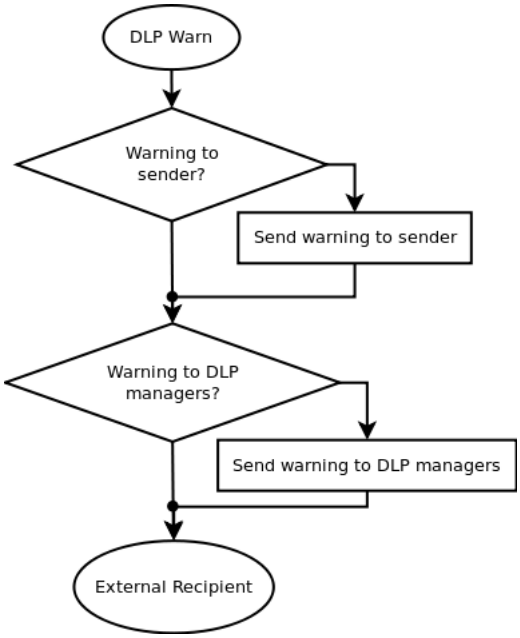


Figure 61: DLP Warn

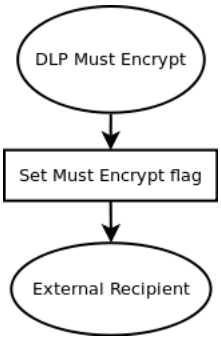


Figure 62: DLP Must Encrypt

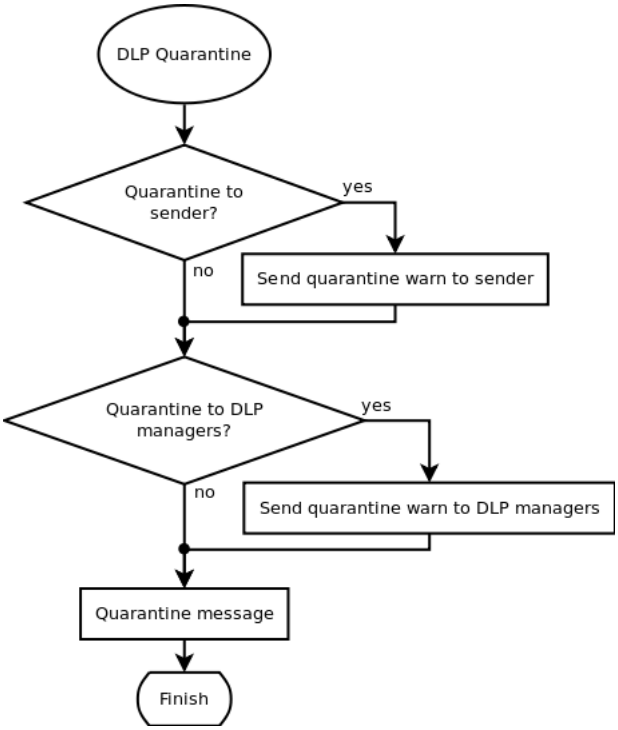


Figure 63: DLP Quarantine

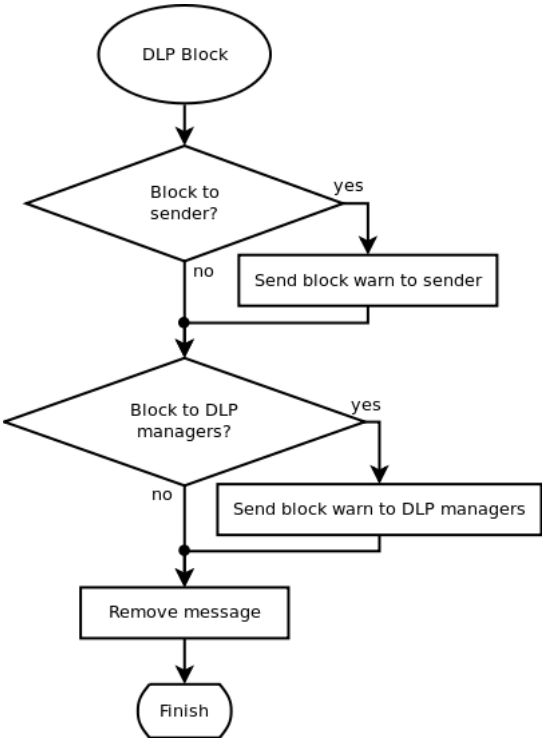


Figure 64: DLP Block

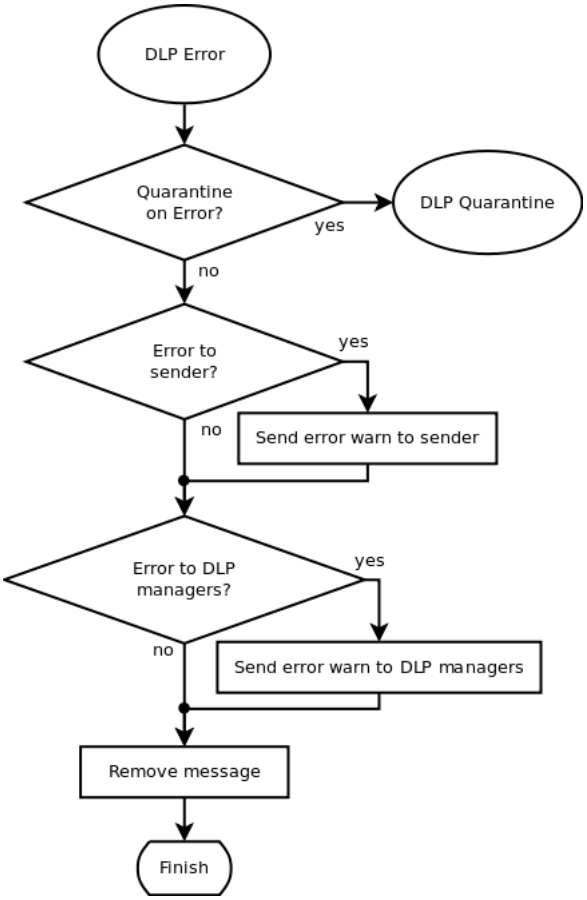


Figure 65: DLP Error

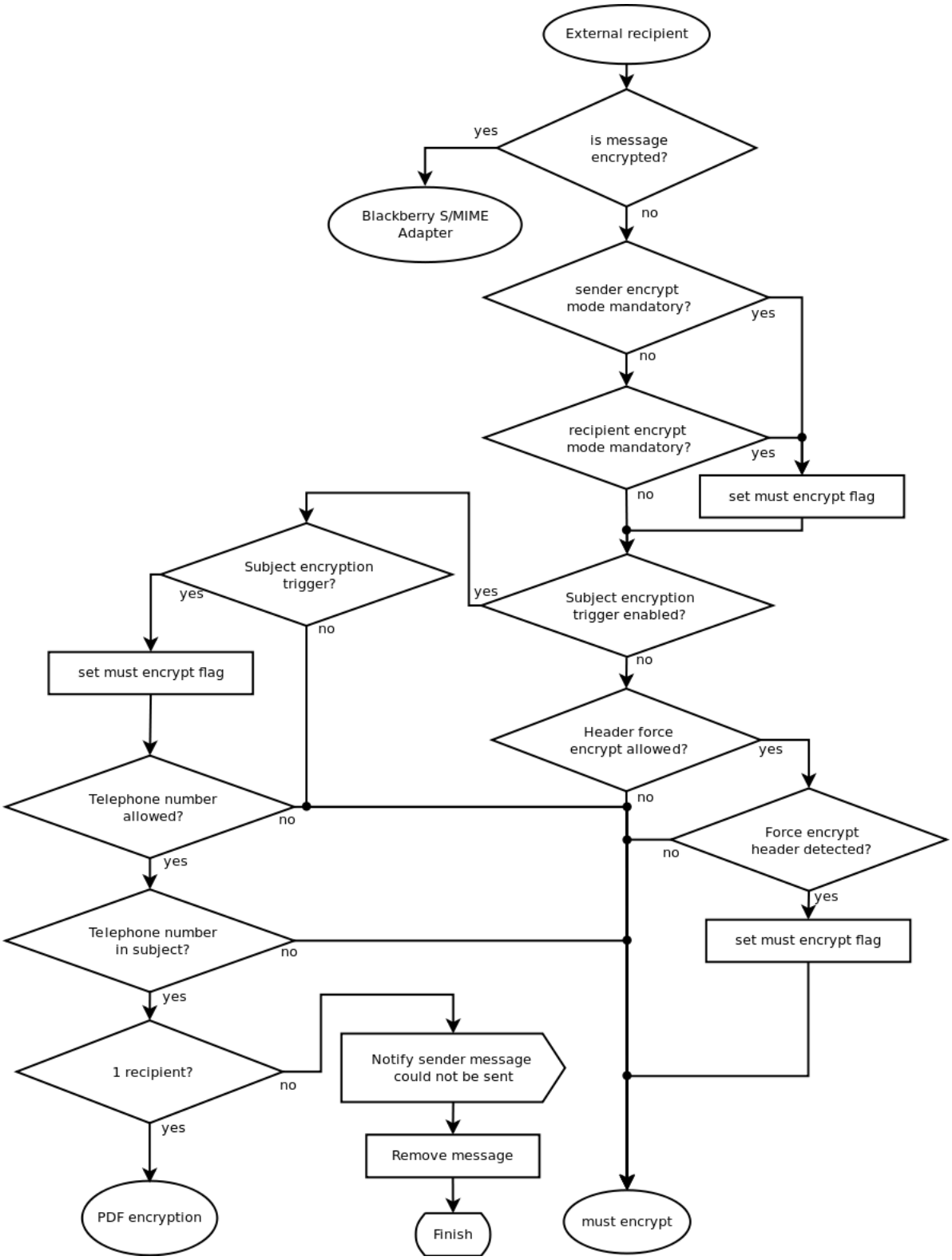


Figure 66: External recipient

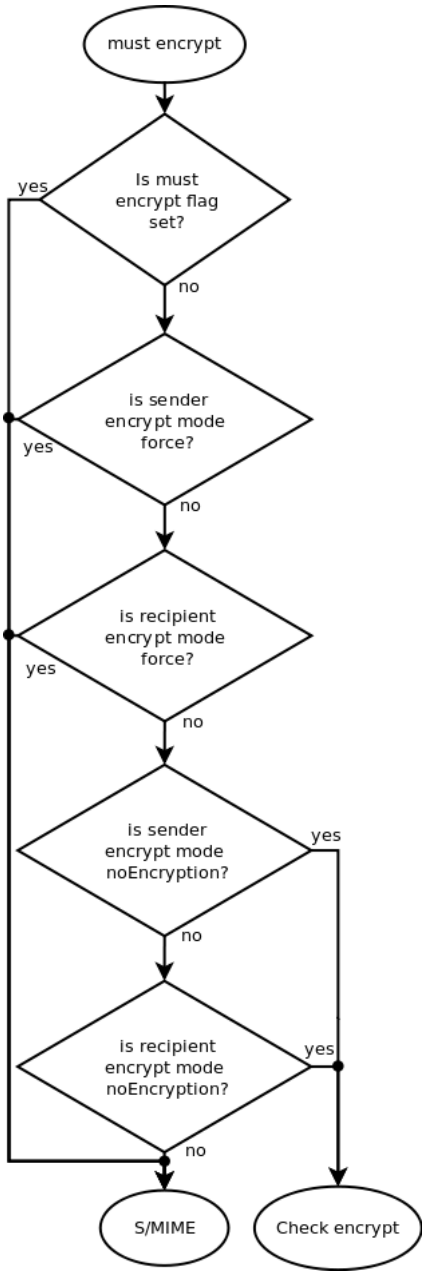


Figure 67: Must encrypt

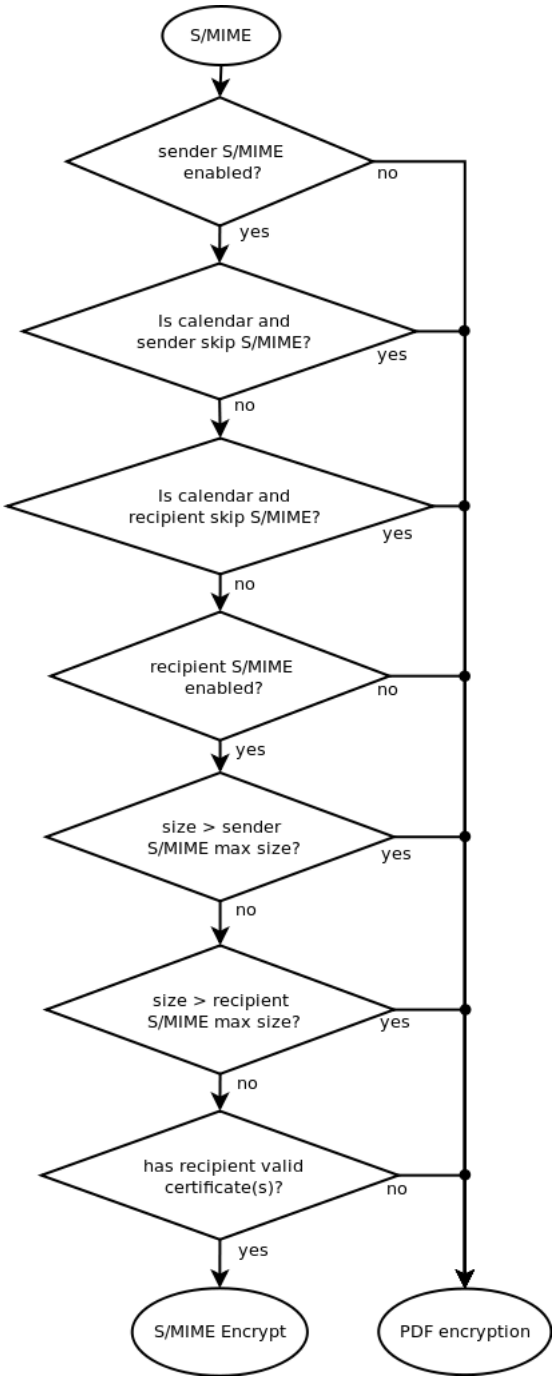


Figure 68: S/MIME

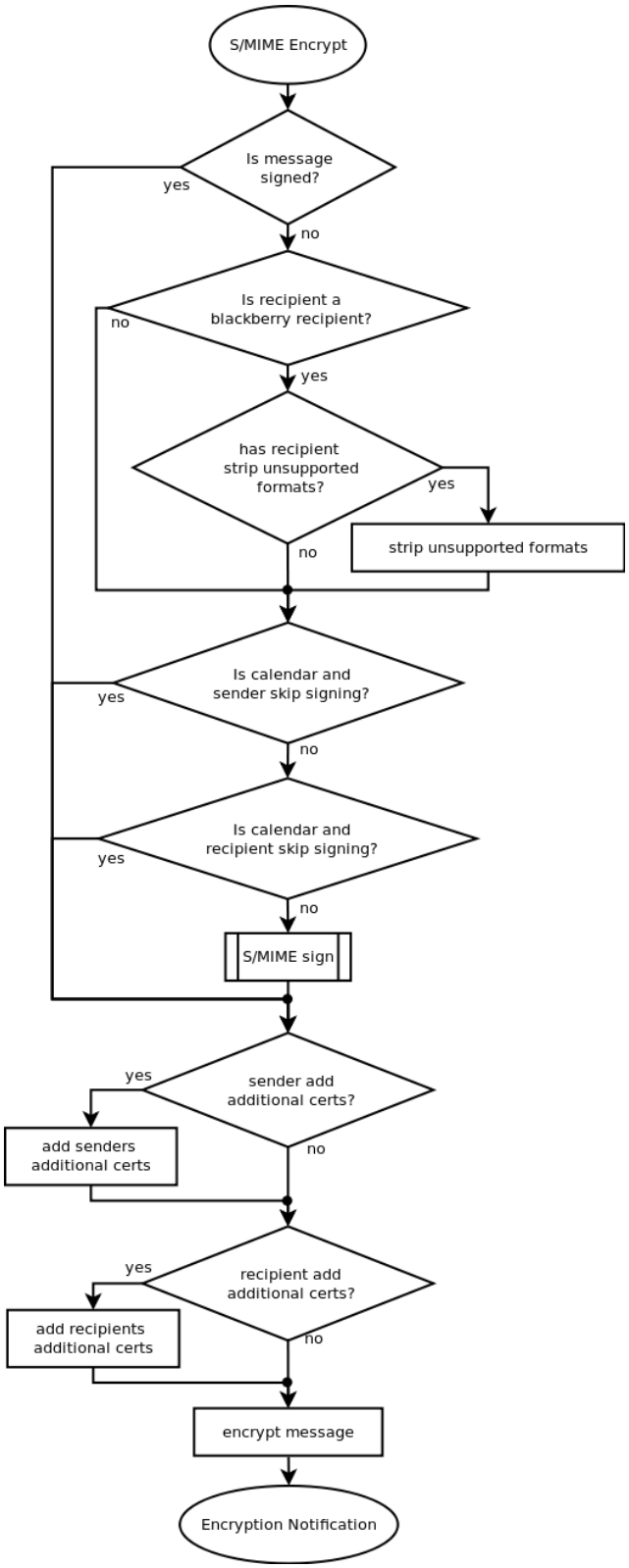


Figure 69: S/MIME encrypt

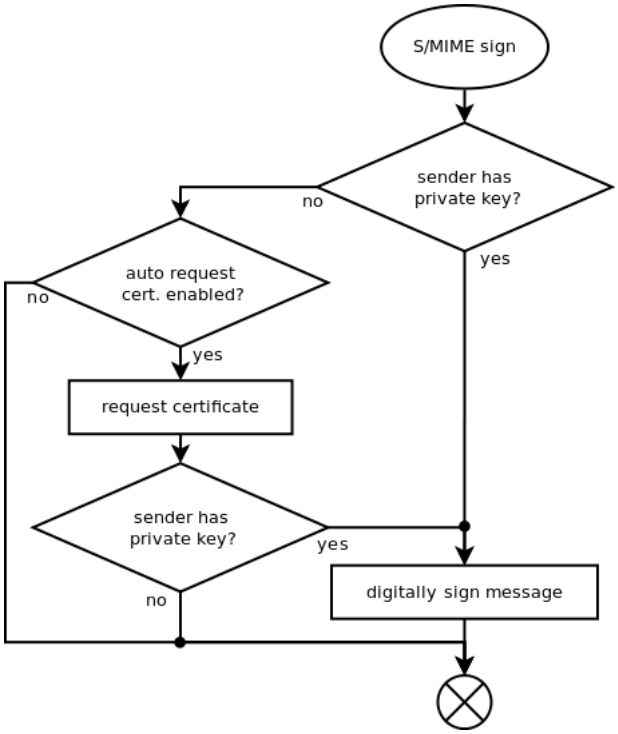


Figure 70: S/MIME sign

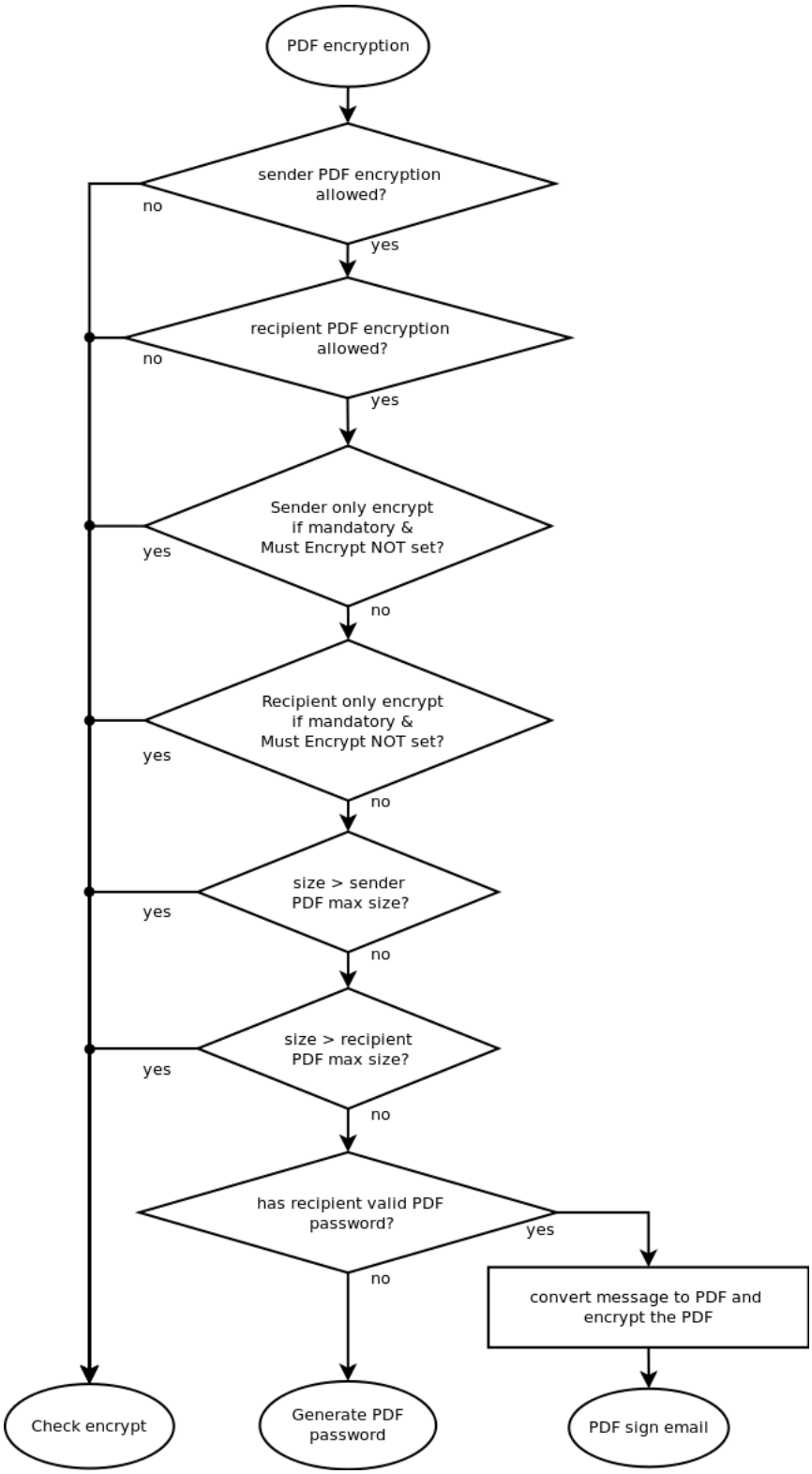


Figure 71: PDF encryption

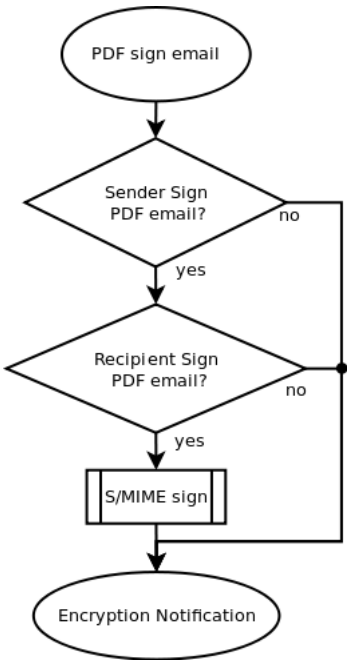


Figure 72: PDF sign email

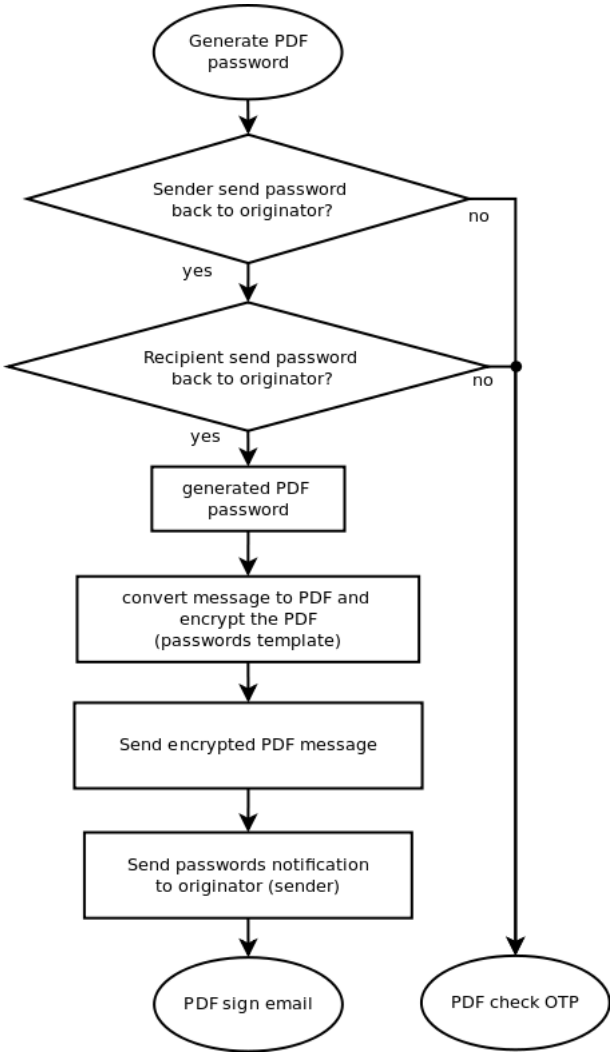


Figure 73: Generate PDF password

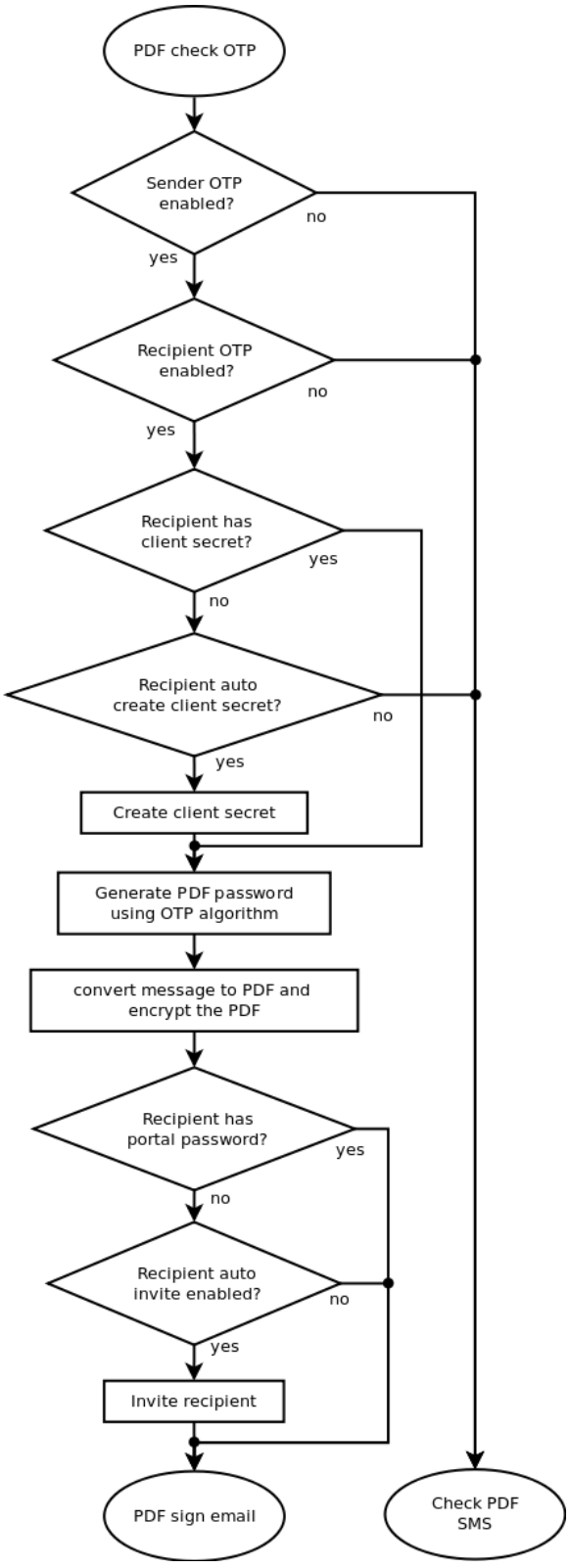


Figure 74: Check PDF OTP

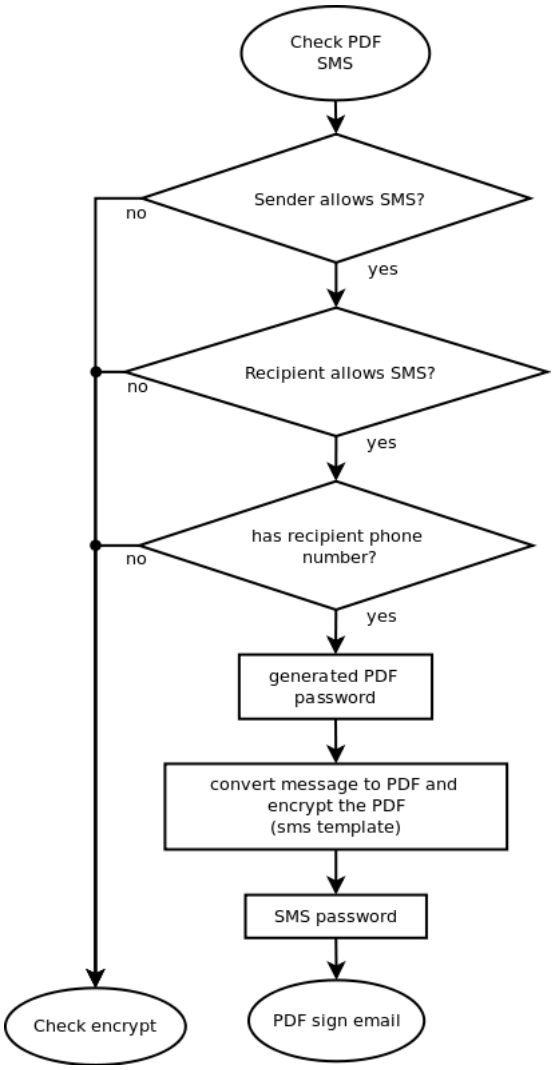


Figure 75: Check PDF SMS

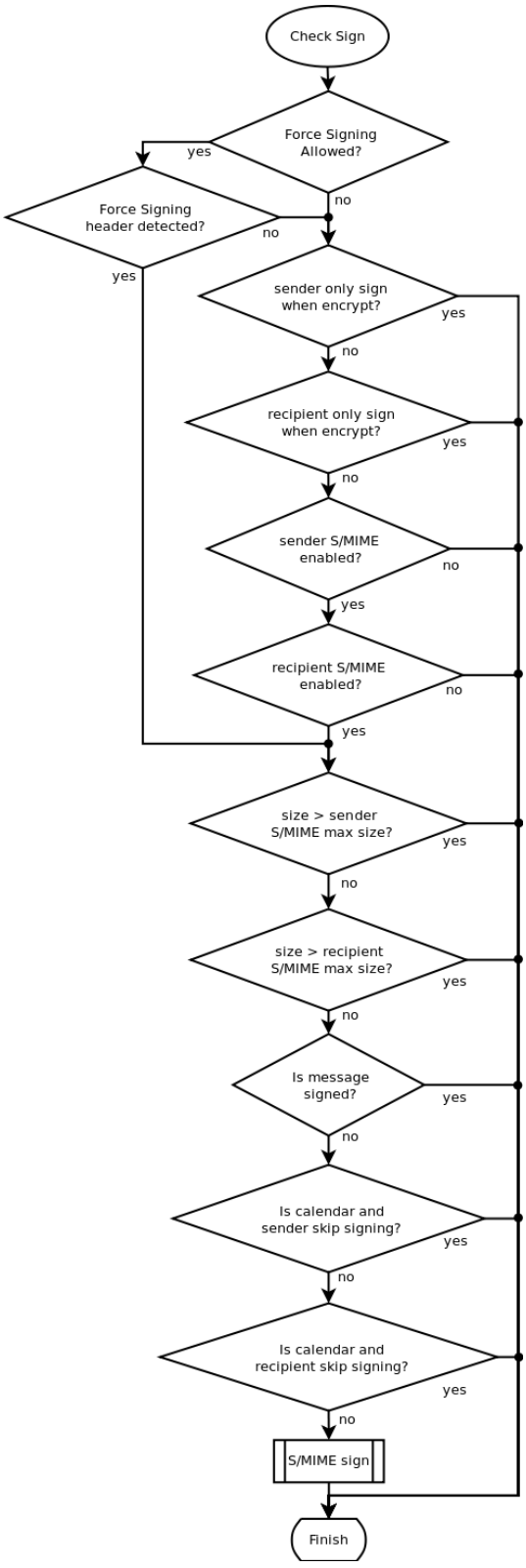


Figure 76: Check Sign

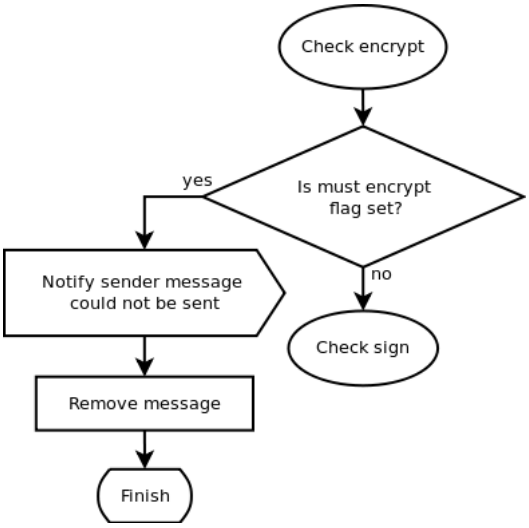


Figure 77: Check PDF SMS

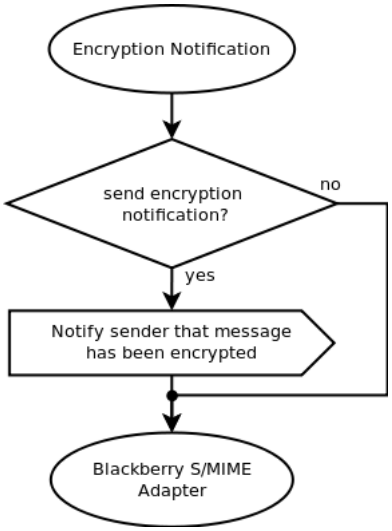


Figure 78: Encryption notification

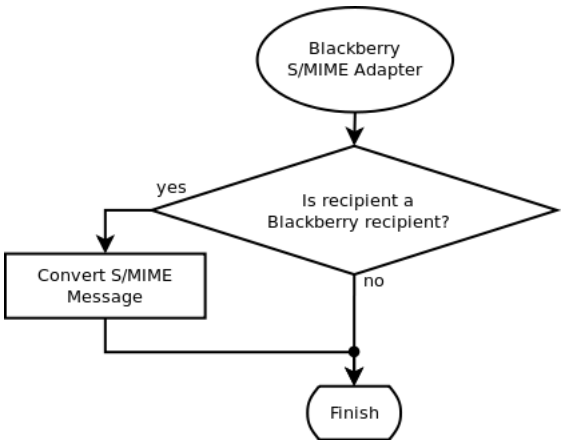


Figure 79: BlackBerry S/MIME adapter

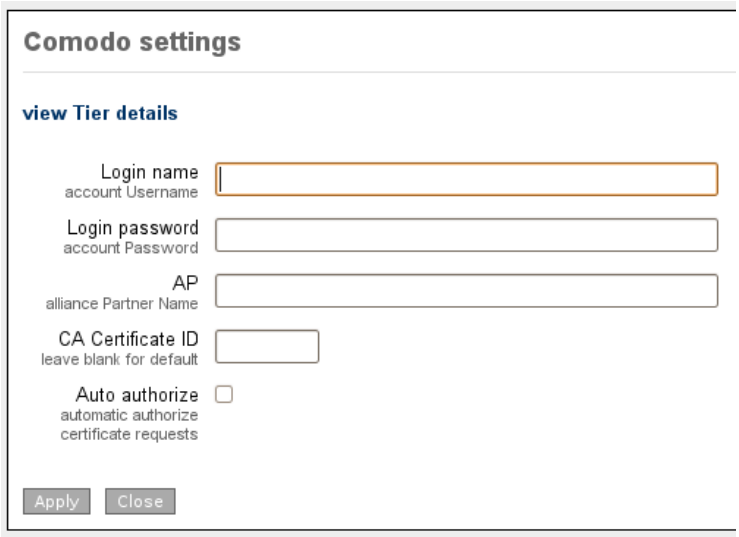


Figure 80: Comodo EPKI settings

# F Comodo certificate request handler

The Comodo certificate request handler requests certificates[25] from Comodo's Enterprise Public Key Infrastructure (EPKI). Comodo's EPKI is an outsourced Certificate Authority managed by Comodo. The main advantage of using certificates issued by Comodo is that these certificates are by default trusted by most systems (like Windows, Mac OS, Ubuntu).

The Comodo EPKI certificate request procedure is a three step procedure:

1. Apply for certificate.
2. Authorize request.
3. Collect certificate.

After every step, the EPKI manager sends an email to the registered EPKI manager. Because requesting a certificate using the Comodo certificate request handler takes several steps, a certificate is not immediately issued.

The Comodo certificate request handler requires a valid EPKI account. A Comodo EPKI account can be provided by DJIGZO or alternatively directly from Comodo. The following Comdoo EPKI account settings can be specified: *Login name*, *Login password*, *AP*, *CA Certificate ID* and *Auto authorize* (see figure 80).

The only required settings are: *Login name*, *Login password* and *AP*. Step two, *Authorize request*, will be done automatically by DJIGZO when *Auto authorize* is enabled. If *Auto authorize* is not enabled, the authorization step

[25] The private key will be generated on the gateway and not leave the gateway. The public key and some identifying information (like email address) will be sent to Comodo. Comodo will then generate, sign and return the certificate.

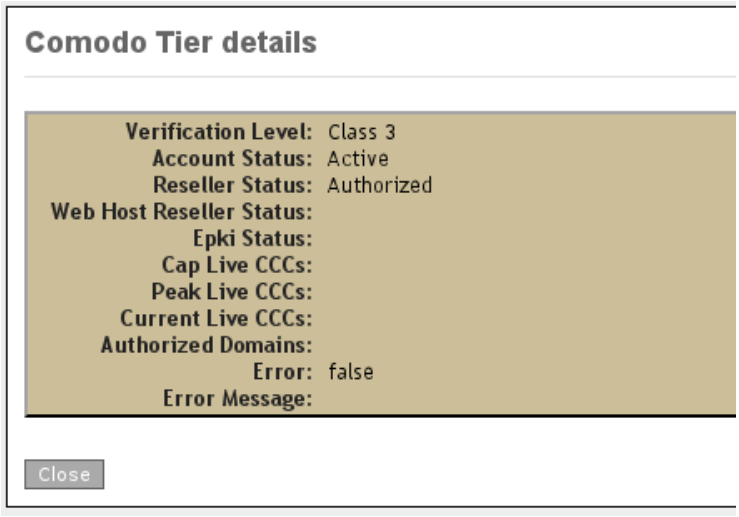


Figure 81: Comodo Tier details

should be done online using the EPKI portal. *CA Certificate ID* should only be specified when the issued certificate should be signed by a non-default CA certificate.

### F.1 Tier details

By clicking *view Tier details*, Comodo EPKI status information can be retrieved (see figure 81).

## G Bulk import

The file format of the file containing all the certificate requests used for the *bulk request* option is as follows:

- The file should be a comma separated file (CSV).
- Values containing commas should be double quoted.
- The first row should contain the column order.
- The CSV is considered to be US-ASCII encoded unless a Byte Ordering Mark (BOM) is used.
- The following columns are supported: *EMAIL*, *ORGANISATION*, *COMMONNAME*, *FIRSTNAME*, *LASTNAME*
- *EMAIL* and *COMMONNAME* are mandatory.
- By default, the maximum number of requests in a single CSV file is 10000.
- The maximum length of an individual entry is 256 characters.

Multiple aliases for the columns are available. Column names are case insensitive:

| Column | Aliases |
|---|---|
| EMAIL | email, e |
| ORGANISATION | organisation, org, o |
| COMMONNAME | commonname, cn |
| FIRSTNAME | firstname, fn, givenname, gn |
| LASTNAME | lastname, ln, surname, sn |

## G.1 Examples CSV

The following example shows how to import two requests. It uses a combination of column name aliases:

```
"e","org","COMMONNAME","fn","surname"
"test0@example.com","organisation 0","user 0","first name 0","last name 0"
"test1@example.com","organisation 1","user 1","first name 1","last name 1"
```

A similar example but this time the values are not quoted and only the required values email and organisation are specified:

```
e,cn
testA1@example.com, cn1
testA2@example.com, cn2
```

# H Unlimited Strength Policy Files

Due to import control restrictions by the governments of a few countries, the jurisdiction policy files shipped with some Java Runtime Environments (JRE) specify that "strong" but limited cryptography may be used. An "unlimited strength" version of these files indicating no restrictions on cryptographic strengths is available for those living in eligible countries (which is most countries). DJIGZO checks whether the "unlimited strength jurisdiction policy files" are installed and if not, a warning is given (see figure 82).

**Note:** most recent versions of OpenJDK (for example OpenJDK that ships with Red Hat 5.4 and Ubuntu 10.04) no longer limit the encryption strength. Installation of the "unlimited strength jurisdiction policy files" is therefore not required in those cases. The "unlimited strength jurisdiction policy files" should only be installed if the warning is shown.

In order to remove this restriction the "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files" must be downloaded from SUN's website (http://java.sun.com/javase/downloads/index.jsp "other downloads") and installed into Java. The DJIGZO Web admin *JCE policy manager* can be used to install the downloaded policy file. The *JCE policy manager* page can be opened by either clicking on the warning link (see figure 82) or by selecting the Admin menu and then selecting "JCE policy".

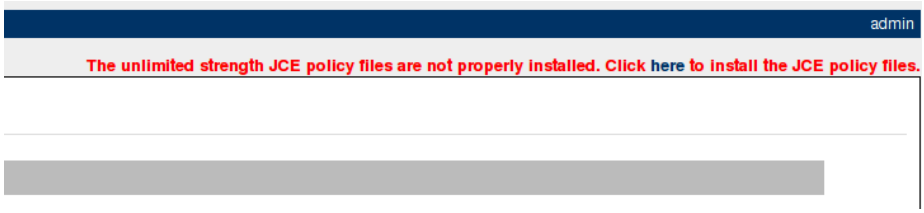


Figure 82: Limited strength encryption warning

**Note: DJIGZO REQUIRES UNLIMITED STRENGTH ENCRYPTION TO BE ENABLED.**

NOTE THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. IT'S YOUR RESPONSIBILITY TO MAKE SURE YOU ARE ALLOWED TO USE STRONG CRYPTOGRAPHY. THE AUTHORS OF DJIGZO ARE NOT RESPONSIBLE FOR ANY VIOLATIONS YOU MAKE.

## H.1 JCE policy manager

The JCE policy manager can be used to install the "Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files" (see figure 83). If the unlimited strength policy files are not yet installed a warning will be shown. To install the unlimited strength policy, the downloaded file `jce_policy-6.zip` should be selected and uploaded. After the file has been successfully uploaded, a message will be shown that the system should be restarted (see figure 84).

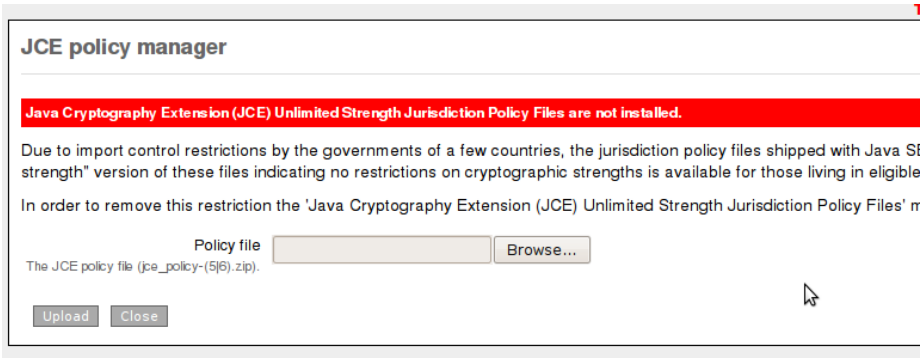


Figure 83: JCE policy manager

The system can be restarted by clicking *restart* (restarting takes about 45 seconds). The system can also be restarted by opening the *Admin* page and click *Restart* on the left hand side menu. Alternatively, if the *DJIGZO Virtual Appliance* is used the gateway services can be restarted by selecting *Restart services* in the virtual appliance console. After the restart the "Unlimited strength..." warning should no longer be shown. If the *JCE policy manager*

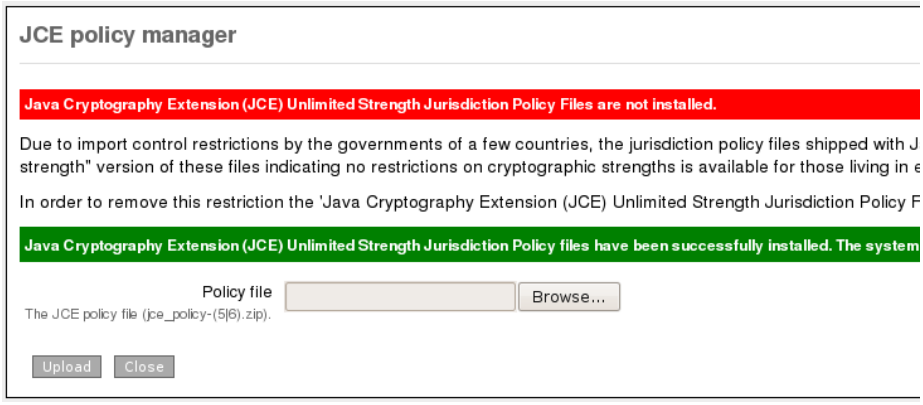


Figure 84: JCE policy uploaded

page is opened, after installing the policy files, a message should be shown that the policy files are already installed (see figure 85).

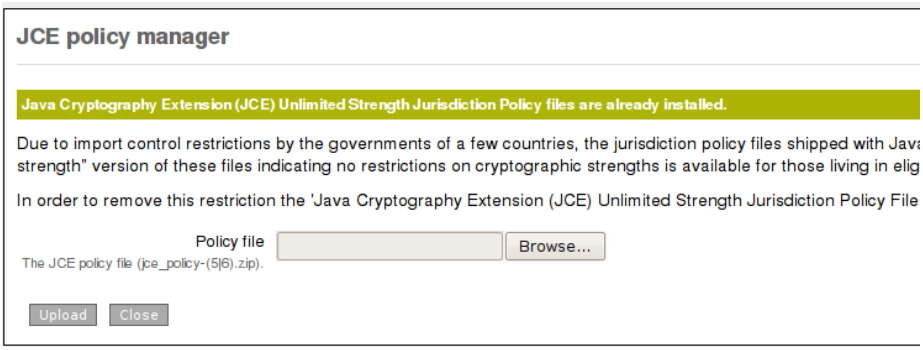


Figure 85: JCE policy already installed

**Note:** it can happen that when Java is updated, the unlimited policy files must be reinstalled[26].

[26] Upgrading Java when using the *Virtual Appliance*, Ubuntu or Red Hat/CentOS should not result in overwriting the policy files.