# WebApp S/MIME Manual

*Release 7.2.1*

Kopano

Jul 27, 2017

# Contents

**Edition 1.0 - The Kopano Team**

This document, the Kopano WebApp S/MIME Manual, describes how to install, upgrade, configure and maintain the S/MIME Plugin with Kopano WebApp on your system. In addition also the usage of the plugin is covered.

# CHAPTER 1

# Introduction

S/MIME (for Secure MIME or Secure Multipurpose Mail Extension) is a security process to guarantee the confidentiality and non-repudiation of e-mail. With the S/MIME plugin for WebApp, users now can make use of this functionality.

**Note:** Please keep in mind that this manual assumes that readers has already knowledge regarding S/MIME and secure e-mail.

If you require more information regarding this subject, Kopano would like to refer you to the following pages:

- https://en.wikipedia.org/wiki/S/MIME
- https://en.wikipedia.org/wiki/Email_encryption

This manual describes installing and configuring the S/MIME-plugin and also on how to sign and encrypt e-mails.

# CHAPTER 2

# Installation

The S/MIME Plugin requires just a few simple installation steps to get it up and running. Follow the steps below to set-up the S/MIME Plugin for all the users on your WebApp server.

## 2.1 RPM based distributions

Use the following command to install the `kopano-webapp-plugin-smime_<version>_all.rpm` package on RPM based distributions:

```
rpm -Uvh kopano-webapp-plugin-smime-<version>.noarch.rpm
```

Replace `<version>` with the correct version.

## 2.2 DEB based distributions

On Debian based distributions use:

```
dpkg -i kopano-webapp-plugin-smime_<version>_all.deb
```

Replace `<version>` with the correct version.

In the next chapter we will go into details on how to configure the S/MIME-plugin.

# CHAPTER 3

# Configuration

To configure the S/MIME-plugin, please edit the following file:

```
/etc/kopano/webapp/config-smime.php
```

This file contain multiple configuration options.

## 3.1 PLUGIN_SMIME_USER_DEFAULT_ENABLE_SMIME

This option allows you to enable or disable the plugin by default for a user.

## 3.2 PLUGIN_SMIME_CACERTS

This option specifies which path's the OpenSSL library should check for CA Certificates, usually this is /etc/ssl/certs.

**Note:** Using /etc/ssl/certs (or your default, central distribution SSL certifices) path is recommended.

## 3.3 PLUGIN_SMIME_CIPHER

This option defines which cipher OpenSSL uses to encrypt an S/MIME message, for PHP < 5.4 the greatest cipher option is 3DES, PHP >= 5.4 has added AES ciphers with a key length of up to 256 bit.

A list of supported ciphers are available at php.net: http://php.net/manual/en/openssl.ciphers.php

**Note:** For security reasons, Using the strongest cipher available is recommended.

## 3.4 PLUGIN_SMIME_ENABLE_OCSP

This option either enables or disables the OCSP validation of public certificates. OCSP is an Internet protocol used for obtaining the revocation status of an digital certificate.

## 3.5 PLUGIN_SMIME_PASSPHRASE_REMEMBER_BROWSER

With this option enabled the browser will offer to save passphrases used in s/mime signing functionality. This functionality is disabled by default.

CHAPTER 4

# Install the Certificate Authority Certificate

In this chapter we will explain how to find the Certificate Authority (CA) and how to install it on the server

## 4.1 Locate the CA certificate

If you are unable to find your CA certificate or your issuer did not provide one, below are the steps to locate and convert your CA.

Convert your .p12 certificate to a .pem certificate.

```
openssl pkcs12 -in certificate.p12 -out user.pem -clcerts -nokeys
```

Now open this .pem file and search for the CA certificate within this file.

```
openssl x509 -in user1.pem -text | grep CA
```

Use wget to download this certificate.

```
wget <certificate URL>
```

If the .crt is in binary format, convert it.

```
openssl x509 -inform DER -outform PEM -in binary.crt -out txt.crt
```

Now copy this certificate to the correct folder and update certificates.

## 4.2 Extract the CA certificate from PKCS #12

Another possibility is extracting CA certificate from your pkcs file. If the pkcs does not contain a CA certificate the output of your file will be empty.

```
openssl pkcs12 -in certificate.p12 -cacerts -nokeys -out ca.crt
```

**Note:** We recommend using the CA certificate your certificate issuer provides.

## 4.3 Install the CA certificate

### 4.3.1 DEB based

Copy your CA certificate to the ca-certificate folder:

```
cp ca.crt /usr/local/share/ca-certificates
```

Update the certificates

```
update-ca-certificates -f -v
```

### 4.3.2 RPM based

Install the ca-certificates package:

```
yum install ca-certificates
```

Enable the dynamic CA configuration feature:

```
update-ca-trust enable
```

Convert your ca.crt to .pem format:

```
mv ca.crt ca.pem
```

Add it as a new file to /etc/pki/tls/certs/:

```
cp ca.pem /etc/pki/tls/certs/
```

Now regenerate the hash links so the system easily finds it:

```
c_rehash
```

**Important:** Make sure your CA certificates can be read by the user which runs the web server, for example www-data on Debian and Ubuntu.

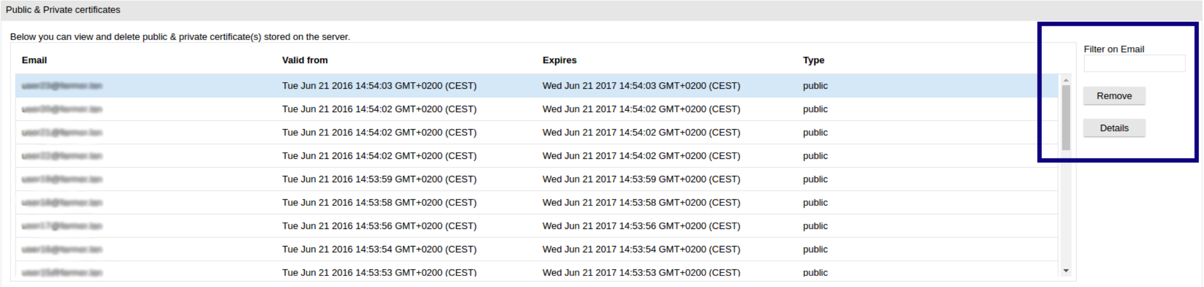
CHAPTER 5

# Certificate management

In this chapter we will describe how to manage the certificates stored on the server.

## 5.1 User certificate management

Certificates can be managed via the certificate manager in the WebApp. This manager can be accessed via the S/MIME settings tab located in the settings menu. Within the certificate manager the user can search for certificates, view the details of the certificate and remove certificates.



Public & Private certificates

Below you can view and delete public & private certificate(s) stored on the server.

| Email | Valid from | Expires | Type |
|---|---|---|---|
| [redacted] | Tue Jun 21 2016 14:54:03 GMT+0200 (CEST) | Wed Jun 21 2017 14:54:03 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:54:02 GMT+0200 (CEST) | Wed Jun 21 2017 14:54:02 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:54:02 GMT+0200 (CEST) | Wed Jun 21 2017 14:54:02 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:54:02 GMT+0200 (CEST) | Wed Jun 21 2017 14:54:02 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:53:59 GMT+0200 (CEST) | Wed Jun 21 2017 14:53:59 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:53:58 GMT+0200 (CEST) | Wed Jun 21 2017 14:53:58 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:53:56 GMT+0200 (CEST) | Wed Jun 21 2017 14:53:56 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:53:54 GMT+0200 (CEST) | Wed Jun 21 2017 14:53:54 GMT+0200 (CEST) | public |
| [redacted] | Tue Jun 21 2016 14:53:53 GMT+0200 (CEST) | Wed Jun 21 2017 14:53:53 GMT+0200 (CEST) | public |

**Warning:** Removing certificates might cause problems with the S/MIME functionality such as mail encryption.

## 5.2 Use public keys from Global Address book users

### 5.2.1 OpenLDAP

The userCertificate attribute holds the X.509 certificates issued to the user by one or more certificate authorities. As required by this attribute type's syntax, values of this attribute are requested and transferred using the attribute description "userCertificate;binary".

The administrator should set the following property into the ldap.propmap:

```
PR_EMS_AB_X509_CERT (aka PR_EMS_AB_TAGGED_X509_CERT)
0x8C6A1102 = userCertificate;binary
```

After this change the kopano-server should be restarted and all the changes should be synced:

```
service kopano-server restart && kopano-admin --sync
```

In ldap config set the following:

```
ldap_user_certificate_attribute = userCertificate;binary
```

Mapped properties can be checked with:

```
kopano-admin --details <user>
```

### 5.2.2 AD

The userCertificate attribute (also referred to as X509-Cert) in Active Directory stores DER Encoded X.509 v3 certificates that are associated with a user.

## 5.3 Certificate management with Python-Kopano

Another possibility for administrators is using python-kopano MAPI bindings to control certificates stored on the server. This way, you can manage certificates without logging into an Kopano account.

Example: This python script will list all the certificates on the server.

```
#!/usr/bin/env python
import kopano
from MAPI.Tags import *

for username in kopano.Server().users(remote=True):

try:
        for item in username.store.root.associated.items():
                print username.name,item.prop(PR_MESSAGE_CLASS).value,item.subject
except:
        pass
```

CHAPTER 6

# Usage

This section describes how the S/MIME Plugin is integrated within the Kopano WebApp client.

## 6.1 Passphrase policy

Not all characters are allowed in the passphrase. Special characters like "<, >, ?, -. +. @, #, %, !, $, _" and even spaces are allowed. Umlauts, Euro Sign (€) and characters other than standard ASCII are not allowed, due OpenSSL restrictions.

There is no minimal passphrase length or maximum length restriction.

**Note:** You won't be able to decrypt the certificate if the passphrase contains non-allowed characters.

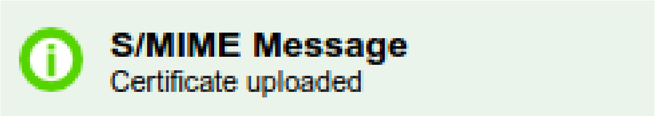
## 6.2 Importing Certificates

For being able to use the S/MIME functionality the private certificate needs to be uploaded to the server. The private certificate has to be of the PKCS#12 format for importing in the WebApp, this format can be obtained by exporting your certificate from Mozilla Firefox for example.

Please go to Option -> Advance -> Certificates -> View Certificates -> Your Certificates and select the certificate you want to use for signing and encrypting e-mails.

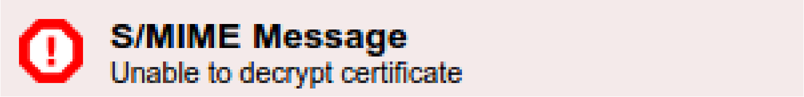
**Note:** The PKCS#12 format saves your private certificate encrypted, so it is recommended to use a strong passphrase. When generating the certificate we also recommend using a strong key-size (2048 bit), hash size (168 or 256 bits) and a strong hash algorithm (sha256 or higher). There are no character restriction regarding this passphrase.

Now that we have obtained a PKCS#12 file (usually called example.p12) we can import it in the WebApp, by navigating to Settings, click on S/MIME and click on *Select* and select your certificate and enter your passphrase and click on upload. If uploading the certificate was successful the WebApp will display a text notification.



S/MIME Message
Certificate uploaded

When the passphare is not correct we are not able to decrypt the certificate.



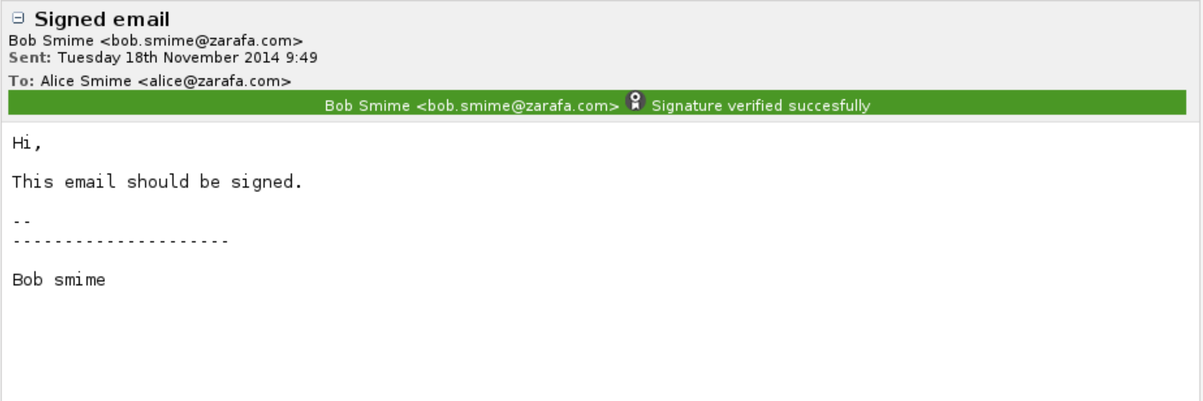
S/MIME Message
Unable to decrypt certificate

**Note:** The WebApp will only import a public certificate if the CA verification is successful and the certificate is not revoked. When OCSP is disabled the certificate is always imported.

## 6.3 Signing and verifying email

To verify email the WebApp only requires the public certificate of the recipient who has signed the email. WebApp will automatically import public certificates if they are newer then the current certificate of the recipient or if you don't have a public certificate of the recipient. Depending on the verification status and configuration the Webapp will display a different color in the S/MIME status bar:

1. **Green bar, when verification is successful. When OCSP is disabled the bar is also green.**

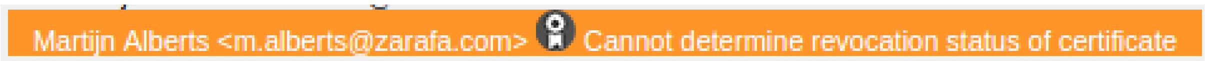


⊟ Signed email
Bob Smime <bob.smime@zarafa.com>
Sent: Tuesday 18th November 2014 9:49
To: Alice Smime <alice@zarafa.com>

Bob Smime <bob.smime@zarafa.com> Signature verified succesfully

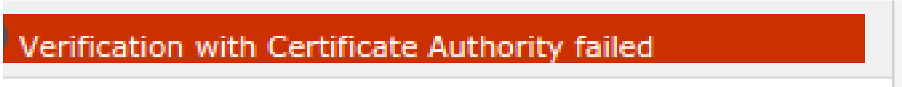
```
Hi,

This email should be signed.

--
--------------------

Bob smime
```

1. **Orange bar, when verification failed due a connection error with the OCSP server or unable to determine the revocation status.**



Martijn Alberts <m.alberts@zarafa.com> Cannot determine revocation status of certificate

1. **Red bar, when a certificate is revoked, CA verification failed or an internal server error.**



Verification with Certificate Authority failed

In any case you can click on the colored bar, this will show a pop-up with extra information about the error or warning message.

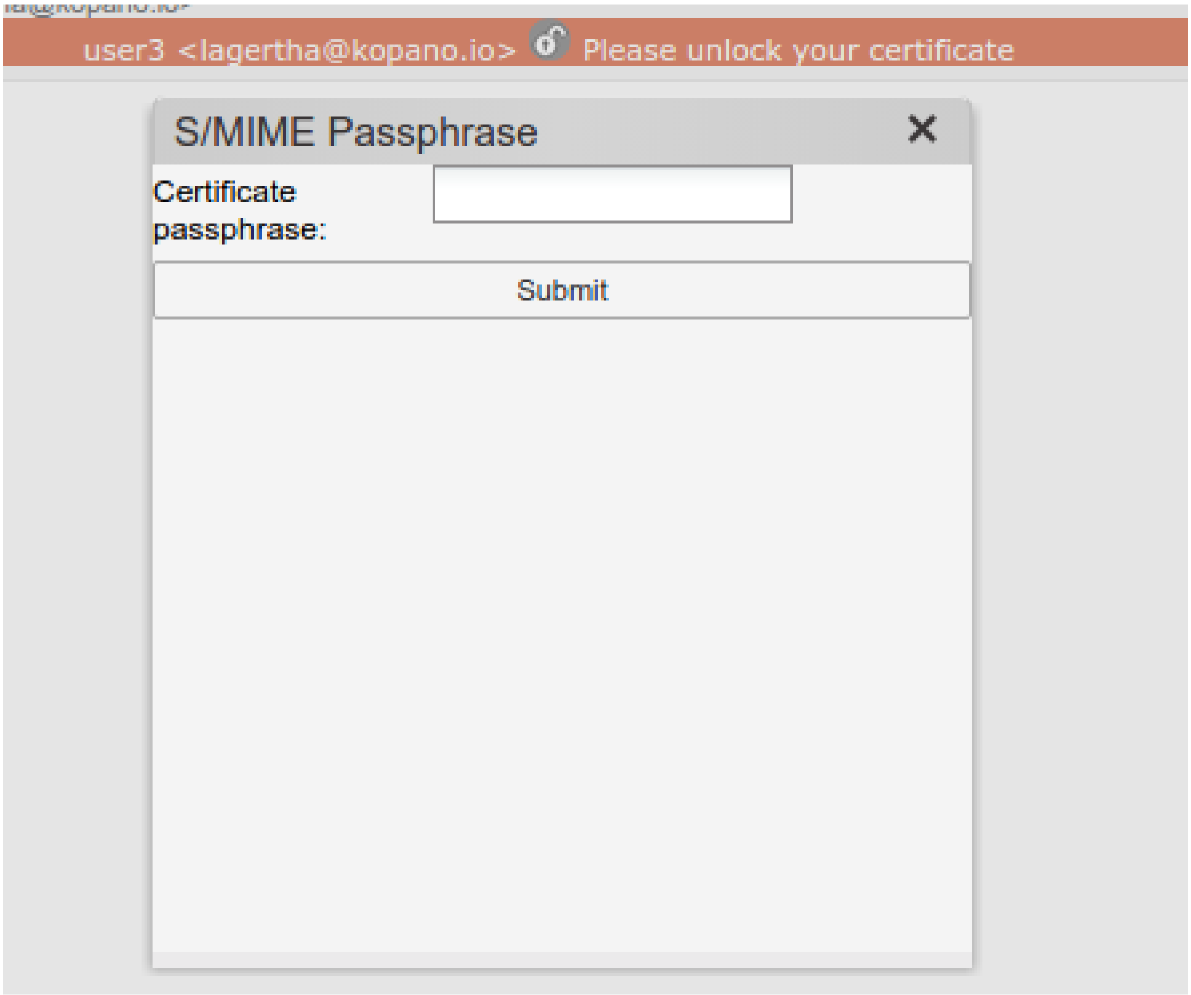
To sign an email, the private certificate has to be already uploaded to the server. If the certificate is already uploaded to the server, we can create a new email and click on the *sign* button, the WebApp will show a popup where you have to enter the passphrase of your certificate. When you click on send, the WebApp will sign your email.

**Note:** Encrypted and signed email will appear in your mail overview's S/MIME column as a lock symbol for encrypted email and a certificate symbol for signed email.

## 6.4 Decryption and encrypting email

To encrypt an email, we need to have the public certificate of the recipient. When you create a new email, click on the encrypt button, write your email and then click on send. If a public certificate is missing, you will receive a notification which recipient doesn't have a public key.

For decryption, you will first need to unlock your private certificate, this can be achieved by clicking on the bar and entering your passphrase in the pop-up. After entering your passphrase, the WebApp will show the decrypted email.



user3 <lagertha@kopano.io> Please unlock your certificate

S/MIME Passphrase

Certificate passphrase:

## 6.5 Encrypting and Signing email

Encrypting and signing email is as simple as clicking on the encrypt and sign button when creating an email. When the email is sent, the email will be first signed and then encrypted. When receiving a signed and encrypted email, the Webapp will show it as an encrypted email only since we only know this after the email has been decrypted.

## 6.6 Allow 'remember passphrase' by browser

**Note:** This functionality only works with Chrome and Firefox. Safari workaround: Add your passphrase to the passphrase vault reachable via: "Preferences -> Password -> Add domain + Password"

This functionality can be enabled by the administrator. See this config setting. After signing an e-mail the browser will ask to remember the passphrase providing your browser is configured to remember passwords. This option can be found in your browser settings.

# CHAPTER 7

## Legal Notice

Adobe, Acrobat, Acrobat Reader and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Apache is a trademark of The Apache Software Foundation.

Apple, Mac, Macintosh, Mac OS, iOS, Safari and TrueType are trademarks of Apple Computer, Inc., registered in the United States and other countries.

Blackberry is the trademark or registered trademark of BlackBerry Limited, the exclusive rights to which are expressly reserved. Kopano is not affiliated with, endorsed, sponsored, or otherwise authorized by BlackBerry Limited.

Collax is a trademark of Collax GmbH.

Debian is a registered trademark of Software in the Public Interest, Inc.

ECMAScript is the registered trademark of Ecma International.

Gentoo is a trademark of Gentoo Foundation, Inc.

Google, Android and Google Chrome are trademarks or registered trademarks of Google Inc.

IBM and PowerPC are trademarks of International Business Machines Corporation in the United States, other countries, or both.

MariaDB is a registered trademark of MariaDB Corporation AB.

Microsoft, Microsoft Internet Explorer, the Microsoft logo, the Microsoft Internet Explorer logo, Windows, Windows Phone, Office Outlook, Office 365, Exchange, Active Directory and the Microsoft Internet Explorer interfaces are trademarks or registered trademarks of Microsoft, Inc.

Mozilla, Firefox, Mozilla Firefox, the Mozilla logo, the Mozilla Firefox logo, and the Mozilla Firefox interfaces are trademarks or registered trademarks of Mozilla Corporation.

MySQL, InnoDB, JavaScript and Oracle are registered trademarks of Oracle Corporation Inc.

NDS and eDirectory are registered trademarks of Novell, Inc.

NGINX is a registered trademark of Nginx Inc. NGINX Plus is a registered trademark of Nginx Inc.

Opera and the Opera “O” are registered trademarks or trademarks of Opera Software AS in Norway, the European Union and other countries.

Postfix is a registered trademark of Wietse Zweitze Venema.

QMAIL is a trademark of Tencent Holdings Limited.

Red Hat, Red Hat Enterprise Linux, Fedora, RHCE and the Fedora Infinity Design logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

SUSE, SLES, SUSE Linux Enterprise Server, openSUSE, YaST and AppArmor are registered trademarks of SUSE LLC.

Sendmail is a trademark of Sendmail, Inc.

UNIX is a registered trademark of The Open Group.

Ubuntu and Canonical are registered trademarks of Canonical Ltd.

Univention is a trademark of Ganten Investitions GmbH.

All trademarks are property of their respective owners. Other product or company names mentioned may be trademarks or trade names of their respective owner.

Disclaimer: Although all documentation is written and compiled with care, Kopano is not responsible for direct actions or consequences derived from using this documentation, including unclear instructions or missing information not contained in these documents.