S/MIME and Sympa mailing lists manager Using signature and encryption with a mailing list manager

Serge Aumont and Olivier Salaün

Comité Réseaux des Universités

Preface

The development was initiated in 1997. The goal was to replace a previous mailing lists server "TULP", initiated in 1992 to organize the Bitnet services migration and was used essentially by French universities. Since April 97, Sympa development provided by the French academic network team (CRU) and distributed under Gnu Public Licence.

Now Sympa includes a lot of sophisticated features and is widely used around the world. On Febrary 2001, we known about 1.500 domains using Sympa : Universities, ISP, administrations, government departments, associations, private companies, ...

1 Sympa description

1.1 Yet another mailing lists server…

The Open source community provide many mailing lists servers (http://listes.cru.fr/sympa/robots.php3). Why developing one more mailing lists server ?

We believe that mailing lists are a major service that needs a perfect adaptation to each application domain. Sympa is the only software that provides the following characteristics:

1. Full compliance to RFCs including MIME in any aspect of the service
2. Internationalization: Sympa is available with the it, de, fr, us, fi, es, cn langages (pl and se are under waiting).
3. Enhanced customisation possibilities
4. Good performances
5. Easy to manage for list master. Example: dynamic lists directory.
6. Integrated user and admin web interface

Easy extensions of features (object oriented code).

1.2 A few original features

Using internal RDBMS

Firt goal is to support large services with high performances and scalability relying on RDBMS (MySQL, PostgreSQL, Oracle, Sybase supported). Sympa has been tested with 300.000 subscribers in a single list on a medium server (bi PII 550mhz with 512 Mega RAM) and some users already have real usage with more than 100.000 subscribers using MySQL. The database allows high performances. It also resolves data acces conflict between Sympa‘s mail and web interface that share the same datas.

The web interface is a portal for all lists:

WWSympa is the name given to Sympa‘s web interface. It is a directory of lists provided by Sympa. Non-authenticated users can access public information but they can also login to access to their private list environment. If identified they can access public list but also some private lists depending on their privileges. After authentication advanced functions are accessible on the web interface.

All administration task can be performed from the web interface: list creation, configuration and closing, messages moderation, subscribers and bounces management .

Shared document repository

The web interface includes groupware features such as html archives with chronological and thread sorting, search form and message deletion (we think that any message sender or list owner as the write to remove his message from archives). In addition, each list owner can create a document repository and define who can upload documents in it, who can browse this repository. This control can be based on the list of subscribers.

Advance MIME features

Any mailing list manager can distribute MIME messages. Sympa also use MIME:

- in message command parsing ; commands in multipart message are recognized
- bounces process ; almost all error report are structured in multipart message
- web archives, digest respect original MIME structure.
- Ouput of commands can use an y mime structure. Example: you can define a welcome message including some images, html etc

Dynamic subscribers definition

All mailing list manager provide subscription and unsubscription to define the subscribers population. But it is now extremely important to be able to define list subscriber using their properties . Usage example: definition of a list of students for some educational unit using the scholarship database. Sympa allows subscriber definition using dynamic external access to LDAP directories or some SQL databases. No further management of subscription and unsubscription is required .

2 S/MIME features and needs for mailing lists

Is user authentication based on From: header field still reasonable ? Mailing lists are nowadays an important media. It is not acceptable that somebody can spoof mailing list editor email to distribute false information. Now almost all administration operations require secure authentication, both on the mail and web interface.

S/MIME features in use for person-to-person mail are also needed for group mail:

- Signature: distributing S/MIME signed messages in a mailing list is fairly easy. Be awarer that adding a message footer or decode and re-encode a message will probably break the signature. Most mailing lists servers can distribute signed messages, though they ignore the signature. The authentication is always based on the From: or based on some password technique. Password authentication is not user friendly and provides a low security level. Note that many lists servers reject all multipart command message, therefore multipart/signed messages are rejected.
- Encryption: a nice mailing lists server should distribute message as received. This include encryption of message if received encrypted for the list. Encryption is completely impossible with any mailing lists server.

2.1 Sympa's objectives with S/MIME

We have set 4 main objectives for Sympa secure services:

1. Sympa must be able to recognize S/MIME signature for any operation (subscribe, distribute message, archive access, ...). Sympa provides a way to define the expected authentication level for each operation for each list. Sophisticated authentication technologie should be required for some sensible operations on a list though basic method is in use for most operations.
2. S/MIME authentication must not be bypassed via the web interface. Authentication based on HTTPS user certificate in the web interface must be required automatically if S/MIME authentication is required on the mail robot interface.
3. The distribution process should accept encrypted message for a list.
4. Certificates and private keys minimum management via Sympa to allow S/MIME and HTTPS usage right now.

3 Sympa S/MIME signature integration

Commands and message distribution are controlled using scenario. It describes who may perform what and what authentication method is required. Scenarios have been introduced in Sympa for a maximum flexibility in setting up access control. This is intended for allowing the definition of an infinite variation on list types depending on list usage such as news-letter, hot-line, private working group, public forum, intranet mailing lists, ...

3.1 The scenario concept

For each list, the configuration file defines a scenario for each operation that can be performed. Scenarios are applied for a particular operation to answer questions such as "Is this user allowed to subscribe to this list ?", "May this list be listed by anyone ?", "is this user trusted to create a new list ?". A scenario consists in rules, evaluated from the first to the last.

Rule structure:

```
  <condition> <auth method> -> <action>
example:
is_subscriber ([listname],[sender]) smtp,smime -> reject
```

The condition is related to the context (the incoming message, the user, the list etc). The authentication method depends on access method to the requested service. The action describe what sympa must do if both condition and authentication method match the current context. Current version of Sympa uses 3 different authentication method:

- smtp: if authentication is based on the From: header field
- md5: applied when a ticket confirmation key generated using MD5 finger print or a password sent by email has been given by the requestor
- mime: if authentication is based on X509 user certificate (S/MIME signature or HTTPS user authentication)

scenario/example

The following scenario is used for a public list with message submission restrictions (applied to control if an incoming message can be distributed, rejected or forwarded to the list editor). This scenario rejects incoming message from subscribers of list spammer and message with multipart/alternative content-type. It submits messages with attachement to the list editor and requires at least md5 authentication from non subscribers.

```
is_subscriber (spammer,[sender])                 smtp,smime -> reject,quiet
match([header->Content-type],/multipart\/alternative/) smtp -> reject
match([header->Content-type],/multipart/)              smtp,smime -> editorkey
!is_subscriber ([listname],[sender])                   smtp -> request_auth
!is_subscriber ([listname],[sender])                   md5,smime  -> do_it
true()                                           smtp,smime -> do_it
```

Authentication method in scenario.

Scenarios are used both for messages and for web access. md5 authentication method is applied when user replies by email to a confirmation request. The same when user is logged using a password received by email.

S/MIME signature and HTTPS user authentication are both based on user X509 certificate, the smime authentication method is applied for both cases. Using HTTPS with user authentication, the "login" and "logout" actions are hidden. No cookie is sent to carry user authentication.. HTTPS can be used without user certificate ; only one benefit: encrypting critical data such as passwords.

4 Mailing lists and encryption

The need for encryption in mailing lists is the same as the need for encryption while exchanging person-to-person email. Sending an encrypted message to a list is as easy as sending a message to a person.

There is no need to distribute an encrypted message when Sympa receives it in a clear form (system as robust as the weaker element), but Sympa will never distribute in clear form a message it received in an encrypted form. Sympa must distribute an encrypted message if and only if it was received encrypted. If a list has its own certificate, a sender can choose either to send an encrypted or a clear message to the list.

You may try it now: https://listes.cru.fr/wws/info/try-sympa-sec

4.1 Encryption how to

Sympa is designed to manage a X509 email certificate for each list (needed for encryption only). When receiving an encrypted message for a list, Sympa:

1. decrypts this message (using the list's private key)
2. encrypts and sends this message to each subscriber who provided a certificate.

Of course, this is possible only if Sympa can access subscribers' certificates. S/MIME is based on asymetric encryption, so when sending a encrypted message to multiple receipients, the sender must prepare a different single message for each receipient using it‘s public key (extracted from it‘s X509 certificat). This requierement force Sympa to break the SMTP grouping that optimize non crypted distribution. Therefore encrypted messages requieres more server ressources and more bandwith.

Fortunately we will not need very large lists using encryption for a long time. The performance of encryption process is not critical. The real need is a high level API for encryption in order to trust Sympa implementation.

Why does Sympa use OpenSSL library

- it's a trusted implementation: OpenSSL is widely used for Apache and a lot of more usages. It‘s an open source software and a lot of developpers contributes to OpenSSL debugging.
- OpenSSL library is shared by Sympa and Apache (or Roxen). This allow a un ique installation for both Apache and Sympa usage. More and more Linux distribution provide OpenSSL in standard configuration.
- it's the first S/MIME open source software

Sympa management of certificates

If Sympa was configured with OpenSSL usage, any list can have its own certificat and private key. Then new subscribers will receive a signed "welcome message". This way each of them has a copy of the list certificate.Then they can send encrypted messages to the list.

It may be a nice idea to configure subscription in such a list to require "smime" authentication method. This allow Sympa to store subscribers certificate when receiving the signed command message. Sympa will then have access to any subscriber certificate and is abale to distribute a encrypted message to any of them.

Development guidelines

- Thaugh we are not crypto gurus, users should trust Sympa encryption features. Sympa sources should remain easy to maintain. OpenSSL allows reasonable and maintainable development on Sympa itself. It provides many APIs. We decided to use a very simple subset of OpenSSL commands with only 4 program calls: decrypt message, encrypt message, verify signature, sign a message.

We took care of inconsistencies. Example: moderating an encrypted message means using encryption in all steps ; from the sender to Sympa, from Sympa to the moderator and in validation process.

Precautions in design

Signed message Content-Type is multipart/signed. First need is to accept commands in multipart messages. Usually, SMTP headers are not part of message digest that is signed. So commands in the subject can be altered without corrupting the signature ! Sympa commands in message subject are applied using smtp authentication method even if the message is signed.

Signer and Sender of a message can be different. This can be dangerous but common in real life (the boss is the signer, his secretary is the sender). Sympa accepts signed messages only if signer and sender have the same email address.

Overcoming programming traps:

- Sympa uses OpenSSL commands as API. If the pass phrase used to protect private key is set in OpenSSL arguments, the Unix command ps will show the pass phrase ! We chose to send it via a named pipe.
- Never store on disk an encrypted message or a private key in a clear form.
- The user option "digest" is incompatible with encryption message because a single digest can contain both clear and encrypted messages. Current version of Sympa just replaces the digest encrypted messages by an indication that a encrypted message was distributed into the list.

Obstacles concerning web archives/signature

It is difficult to provide compatibility between web archives and S/MIME messages: the signature is checked when message is received. Then it is easy to show a button for checking the signature. But are we allowed to keep a signature on a document that we converted into HTML ? The HTML conversion is made once (at reception time) while signature checking is performed at consultation time using the original message. How to ensure that nobody has changed the HTML form on disk?

Obstacles concerning web archives/encryption

The message is archived using the original encrypted form. It is unsecure to display an encrypted message in a clear form across an encrypted and authenticated HTTPS session. HTTPS is not a native encrypted application, it is just HTTP over SSL. Resulting are some traps, for example the message can be stored in the client cache in a clear form.

TODO: we decide to introduce a new feature in web archive to receive by mail an old message. The same algorithm as for encrypted process distribution will be used.

4.2 What about PGP, GPG ?

Why did we choose S/MIME rather than PGP: S/MIME developments in Sympa are part of one of our project for security based on PKI. S/MIME is based on X509 user certification. Certificates management provides a solution for secure mail but also for secure web. Secure mail in Sympa is of poor interest if the web interface is not securised as well. PGP can't do both.

However some users still prefer nice pine or exmh that can't use S/MIME, most users have an email user agent that support S/MIME (Outlook and Netscape messenger), some of them needs a pluggin (Eudora).

Because users asked for it, we plan to introduce PGP signature checking and perhaps PGP encryption.

5 Sympa project direction

Sympa is a very active project ; we still have a lot of project for it: full virtual robot, plugging a virus scanner, increase performances (some users want 50 000 lists), new groupware tools such as shared bookmarks calendar, e-vote feature etc. Two important direction are in progress:

- About certificates: current version of Sympa do not use certificate revocation list this is mandatory for real security. It would be nice to warn users when his/her certificate is going to expire and alert list master and list owners before list certificate expires. In addition it would be practicale to allow user certificate download from web subscribers review.
- LDAP is also a major development direction. Sympa allready uses LDAP extract subscribers email, it will also uses it for authentication method with a per domain LDAP directory definition, for evaluation of condition in scenario and for cooperation with different Sympa servers providing a lists directory.