

# S/MIME et le gestionnaire de listes de diffusion Sympa

## Utilisation des certificats X509 pour l'authentification et la diffusion de messages chiffrés via un serveur de listes

Serge Aumont, *sympa-authors AT cru.fr*, *Url: <http://www.sympa.org>*

Comité Réseau des Universités

Sympa est un gestionnaire de listes de diffusion qui inclut un grand nombre de fonctionnalités telles qu'une interface web complète permettant aux utilisateurs d'accéder à tous les services ainsi que l'administration détaillée de chaque liste. Sympa permet pour chaque opération d'exprimer des conditions d'accès très fines. Encore faut-il garantir une identification sûre des utilisateurs. A cet effet, il est possible de forcer une méthode d'identification basée sur des certificats X509, tant pour les opérations de messagerie (signature S/MIME des messages) que pour les accès sur l'interface http (HTTPS avec certificat du client). En outre, la diffusion de messages chiffrés est rendue possible par la capture des certificats d'abonnés.

Les développements de Sympa ont démarré en 1997. L'objectif était de remplacer les précédentes générations de moteurs de listes de diffusion utilisés par le CRU pour assurer la migration des services « listserv » de Earn depuis 1989. Sympa propose aujourd'hui un large éventail de fonctionnalités tant dans le domaine des services utilisateurs (multiples options d'abonnement, interface WEB avec quelques outils de « groupware » associés aux listes, ...) que dans celui de l'administration (possibilité d'appeler un anti-virus, utilisation d'annuaires LDAP, construction automatique de listes d'abonnés, ...). Il est distribué sous licence GPL et compte (en septembre 2001) plus de 2000 sites utilisateurs dont la plupart des universités françaises.

## 1. Description de Sympa

### 1.1 Encore un gestionnaire de listes de diff...

On trouve une multitude de logiciel de listes de diffusion en particulier dans la communauté des logiciels libres. Pour vous en convaincre, consultez l'inventaire des robots de listes de diffusion (<http://listes.cru.fr/sympa/robots.php3>). Pourquoi en développer un de plus ?

Notre expérience nous a prouvé que le service de listes de diffusion est un service de base au même titre que le WEB et qu'il existe une multitude d'utilisations différentes de ce service, aussi nous pensons qu'un tel service doit être adapté finement aux besoins du site qui l'installe et de chacun des abonnés de ce site. A notre connaissance, Sympa est le seul moteur réunissant les caractéristiques suivantes :

- 1- Complète conformité aux RFCs, en particulier concernant MIME dans tous les services (diffusion de messages, message de commandes, message de bienvenue)
- 2- Multilingue : Sympa est disponible dans 8 langues it, de, fr, us, fi, es, cn, pl
- 3- Notion de robots virtuels (similaire à celle des serveurs virtuels d'Apache)
- 4- Personnalisation des contrôles d'accès, des messages de services des pages web, pour chaque liste, chaque robot virtuel, chaque serveur
- 5- Performances élevées et extensibles : plusieurs centaines de milliers d'abonnés, plusieurs milliers de listes (utilisation d'un SGBD)
- 6- Administration intégrée, par exemple la création ou la suppression de listes
- 7- Code orienté objet permettant des évolutions rapides et une maintenance simplifiée des sources.

### 1.2 Des fonctionnalités originales

#### 1.2.1 WWS : l'interface web universelle d'accès au service

World Wide Sympa est le nom donné à l'interface WEB. Cette interface unique donne une vision complète de toutes les listes. Les utilisateurs non authentifiés ne voient que la partie publique du service, les abonnés ont accès à la liste de leur abonnement et aux ressources privées de leurs listes. Par exemple, ils peuvent supprimer des archives les messages dont ils sont les auteurs. Les propriétaires de listes peuvent administrer la liste des abonnés, gérer les rapports de non remise, modérer les messages en attente, etc. Le listmaster, lui, valide les demandes de création de listes, ferme des listes etc.

#### 1.2.2 Le partage de documents

A chaque liste, il est possible d'associer un espace de dépôt de document. On peut alors définir les droits de dépôt et de consultation de cet espace. La configuration la plus utilisée permet à un groupe de travail de partager des documents en réservant la consultation et le dépôt aux seuls abonnés.



### 1.2.3 Fonctionnalités MIME avancées

Tous les gestionnaires de listes de diffusion permettent la diffusion de message MIME. Sympa utilise MIME :

- Dans l'analyse des commandes en mode messagerie : les commandes dans un message multipart/alternative sont reconnues (nombreux sont les utilisateurs qui utilisent « outlook » sans savoir qu'ils postent des messages au double format `text/html` et `text/plain`).
- De plus en plus de MTA utilisent conformément aux RFCs des messages MIME pour les rapports de non remise. Sympa exploite cette structure pour le traitement automatique des erreurs.
- La diffusion des messages prend en compte la structure MIME dans tous ces aspects : archives web, digest etc. En outre l'option d'abonné « `urlize` » permet de remplacer les attachements par des liens vers l'interface web.

### 1.2.4 Définition dynamique des listes d'abonnés

Comme tous les gestionnaires de listes de diffusion, Sympa permet de constituer des listes par abonnement et désabonnement ; il est aussi possible de constituer des listes dynamiquement par extraction d'adresses email d'annuaires externes. Ainsi, Sympa peut se connecter à un annuaire LDAP ou une base de données pour sélectionner une catégorie d'utilisateurs (comme membre d'une liste donnée). Par exemple, on peut définir dans Sympa une liste de diffusion des étudiants d'une filière directement par les paramètres d'une requête SQL vers le serveur « Apogée » de la scolarité de l'université.

LDAP peut aussi être utilisé pour l'authentification par mot de passe des utilisateurs ou pour définir des privilèges associés à une population définie dans des annuaires d'établissement.

### 1.3 Fonctionnalités S/MIME et listes de diffusion.

Est-il raisonnable de faire confiance au champ `From:` des messages pour identifier les utilisateurs ? Certaines listes de diffusion représentent maintenant des enjeux importants pour la communication de l'établissement : si un président d'université utilise cet outil pour adresser ses vœux à l'ensemble du personnel, il devient du coup intolérable qu'un tiers puisse usurper son identité pour diffuser un message. Par ailleurs, la plupart des opérations d'administration requièrent une authentification sûre, que ces opérations soient menées via le mail ou via une interface web.

Les services offerts par S/MIME dans la communication de personne à personne sont aussi indispensables pour la communication de groupe :

- Signature : Distribuer un message signé avec S/MIME dans une liste est assez facile. Il suffit de ne pas modifier le corps de celui-ci (attention à l'ajout automatique d'un pied de message qui corromprait immanquablement la signature). La plupart des serveurs de listes ignorent purement et simplement la signature des messages. S'ils assurent la diffusion de messages signés ils n'exploitent pas la signature et continuent à baser l'identification des auteurs de messages de commandes et des messages à diffuser sur le champ `From` ou utilise un système de mot de passe peu convivial et dont le niveau de sécurité est discutable<sup>1</sup>.
- Le chiffrement : un bon serveur de listes de diffusion devrait pouvoir diffuser tous les messages reçus pour une liste, y compris les messages chiffrés. Sympa propose ce service.

### 1.4 Objectifs relatifs à S/MIME pour Sympa

Nous avons poursuivi quatre objectifs principaux dans ce domaine :

- 1- Sympa doit reconnaître la signature S/MIME pour toutes les opérations d'abonnement de diffusion de messages, d'accès aux archives, etc. Le niveau d'authentification requis (`From:`, `password` ou signature par chiffrement) doit être configurable pour chaque opération et pour chaque liste : seules certaines opérations ou certaines listes sensibles nécessitent une authentification sophistiquée. Ce système doit pouvoir être appliqué pour les opérations qui ne dépendent pas spécifiquement d'une liste (ex : accès à la liste des listes ou création d'une liste).
- 2- L'authentification par signature S/MIME, requise pour une opération via l'interface messagerie, ne doit pas pouvoir être outrepassée en utilisant l'interface HTTP. L'authentification HTTPS basée sur un certificat personnel du client est requise si une authentification S/MIME l'a été pour cette opération.
- 3- Le processus de diffusion doit permettre la diffusion de messages chiffrés.
- 4- Une gestion minimum des certificats doit être jointe à Sympa avec en particulier un cache des certificats d'abonné, l'installation et la diffusion de certificats X509 pour des listes qui en ont besoin.

## 2. Intégration de la signature S/MIME dans Sympa

Tous les privilèges et en particulier les commandes et la diffusion de messages sont contrôlés par l'évaluation de conditions dépendant du contexte d'appel. Pour assurer une flexibilité maximale, un petit langage appelé « scénario » permet de créer de nouvelles conditions de contrôle adaptées à des usages très variés de l'outil « liste de diffusion ».

<sup>1</sup> Notez que beaucoup de serveurs rejettent les messages de commandes `multipart` et donc interdisent l'usage de messages `multipart/signed`

## 2.1 Les scénarios

Le fichier de configuration de chaque liste spécifie pour chaque opération possible un nom de scénario. Les scénarii correspondants permettent d'évaluer des conditions comme « peut-on diffuser le message de cette personne pour cette liste sachant qu'il contient un attachement ? », « peut-on montrer cette liste à cet usager connecté depuis un poste du réseau local ? », « cette personne est-elle autorisée à créer une liste ? », etc. Un scénario est constitué de règles évaluées séquentiellement.

Structure d'une règle :

```
<condition> <auth method> -> <action>
```

Exemple :

```
is_subscriber ([listname],[sender]) smtp,smime -> reject
```

La condition est relative au contexte (les entêtes du message concerné, le nom de la liste, l'adresse IP du host depuis lequel l'utilisateur est connecté etc). L'action décrit ce que Sympa doit faire si la condition est satisfaite dans le contexte particulier ; par exemple, rejeter le message, diffuser le message ou transmettre le message au modérateur.

La version actuelle de Sympa utilise trois méthodes d'authentification différentes :

- smtp : l'authentification la plus faible, on fait confiance au champ From.
- md5 : l'authentification a été faite par un mot de passe (sur l'interface web) ou par retour d'un ticket à usage unique (calculé par MD5) pour les services en mode messagerie
- smime : si l'authentification est basée sur un certificat X509 (signature S/MIME ou session HTTPS).

### 2.1.1 Exemples de scénario

Le scénario suivant est utilisé pour une liste publique avec des restrictions à la diffusion de messages. Dans ce cas les messages des abonnés à la liste spammer et les messages contenant plusieurs alternatives du même texte sont rejetés, les messages contenant des attachements sont modérés, enfin, les non abonnés doivent confirmer leur identité.

```
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
```

## 3. Diffusion de messages chiffrés

Le besoin de chiffrement dans les listes de diffusion ne fait pas de doute dès lors que les personnes commenceront à utiliser le chiffrement dans la messagerie de personne à personne.

Il est parfaitement inutile de diffuser sous forme cryptée un message que sympa aurait reçu en clair (son auteur n'a pas éprouvé le besoin de chiffrer son message). A l'inverse, un message reçu chiffré ne doit jamais être mis en clair, ni lors de sa diffusion, ni dans les archives, digest, etc. Sympa doit donc diffuser un message en le chiffrant si et seulement si il l'a reçu sous cette forme. Ceci implique en particulier que Sympa puisse gérer un certificat pour chaque liste pour laquelle il y a un besoin de chiffrement.

Vous pouvez essayer une liste permettant le chiffrement : <https://listes.cru.fr/www/info/try-sympa-sec>

### 3.2 Implémentation du chiffrement

Sympa permet d'installer un certificat et une clef privée pour chaque liste. Lors de la réception d'un message chiffré pour la liste :

- 1- Sympa déchiffre le message avec la clef privée de la liste
- 2- Sympa chiffre le message pour chacun des destinataires (en utilisant le certificat de chaque destinataire)

Bien entendu ceci n'est possible que si Sympa dispose d'un accès au certificat de chaque abonné de la liste. Le chiffrement asymétrique impose à Sympa de préparer un message spécifique pour chaque destinataire et à abandonner le groupage SMTP qui optimise la diffusion des messages non chiffrés. Le surcoût de chiffrement de cette opération est donc élevé (bande passante et ressource machine).

Les spécialistes du chiffrement ne manqueront pas de faire remarquer que le chiffrement asymétrique dans S/MIME se limite au chiffrement d'une clef aléatoire utilisée pour faire du chiffrement symétrique. Cette première étape consistant à générer un aléa et à l'utiliser pour chiffrer le message avec un algorithme symétrique pourrait donc être factorisée pour chacun des destinataires. Convenons que les besoins de performances liés à des listes avec plusieurs dizaines de milliers d'abonnés et avec chiffrement sont pour le moment des considérations uniquement théoriques ! En outre il nous semble fondamental d'asseoir la confiance dans les fonctionnalités de chiffrement de Sympa sur une implémentation de référence des algorithmes S/MIME. Plus on utilise des fonctions de haut niveau de la librairie de chiffrement plus on peut avoir confiance dans Sympa lui-même.

### 3.2.1 Le choix d'OpenSSL

C'est une implémentation de référence : OpenSSL est très largement utilisé pour Apache+Mod\_SSL. La librairie OpenSSL est partagée avec le serveur WEB (Apache ou Roxen) et l'installation de OpenSSL permet de sécuriser l'interface WEB aussi bien que l'interface messagerie. OpenSSL est le premier (le seul ?) logiciel libre ayant des fonctionnalités S/MIME.

### 3.2.2 Gestion de certificats

Quand Sympa est configuré pour utiliser OpenSSL, chaque liste peut disposer d'un certificat et d'une clef privée. Les nouveaux abonnés reçoivent un message de bienvenue signé avec la clef privée de la liste (vérifiable avec le certificat de la liste joint dans le message). Comme pour la correspondance de personne à personne, le client de messagerie du nouvel abonné stocke le certificat de la liste lorsqu'il vérifie la signature du message ; celui-ci peut donc envoyer des messages chiffrés à l'adresse de la liste.

De la même façon, Sympa stocke les certificats contenus dans tout message signé adressé au robot ou à une liste. Si l'abonnement à une liste a été configuré pour imposer l'abonnement par message signé, Sympa dispose des certificats de tous les abonnés de la liste ; la diffusion de messages chiffrés est donc possible.

### 3.2.3 Quelques repères pour le développement

Pas de spécialiste du chiffrement parmi les auteurs de Sympa. Nos objectifs ne peuvent donc être atteints qu'en utilisant une API de haut niveau avec le produit de chiffrement. OpenSSL autorise une utilisation aisée et stable d'un très petit sous ensemble de commandes de OpenSSL composé de 4 appels : chiffrement, déchiffrement, signature et vérification de la signature d'un message.

Nous avons porté une attention particulière à la cohérence des différentes chaînes de traitement. Par exemple, un message chiffré peut être soumis à modération mais ne doit jamais être passé en clair lors de ce processus (transmission au modérateur, validation par le modérateur, diffusion puis archivage).

Les messages signés sont des messages multi-parties (`multipart/signed`). L'authentification S/MIME de commandes placées dans un message signé suppose préalablement d'être capable de traiter les messages multi-parties. Par ailleurs, l'empreinte d'un message utilisée pour la signature de celui-ci n'inclut pas les entêtes du message (on devine facilement les raisons de cette restriction : les entêtes sont souvent modifiées lors de l'acheminement du message, ce qui casserait la signature). De ce fait, la méthode d'authentification `smime` ne doit pas être appliquée à des commandes placées dans le sujet d'un message même si celui-ci est correctement signé<sup>2</sup>.

Le `distinguished name` d'un certificat permettant de signer des messages contient toujours l'adresse email du titulaire du certificat. Sympa ignore les autres composants du `distinguished name`. L'adresse du "Signer" et le "Sender" d'un message peuvent différer, Sympa ne valide la signature du message que s'il y a identité (à la casse près) entre les deux adresses.

#### Quelques pièges :

- Sympa utilise les commandes d'OpenSSL, si la « phrase de passe » (passphrase) utilisée pour protéger la clef privée d'une liste est passée dans les arguments d'une commande, la commande Unix `ps` révèle ce secret. Nous utilisons donc un « pipe » nommé.
- Ne jamais stocker en clair un message reçu.
- Du point de vue S/MIME, l'empreinte est calculée sur le corps brut du message, c'est à dire en considérant tout le contenu depuis la première ligne vide jusqu'en fin de message et sans considération sur le codage ventuel des parties de corps. Sympa manipule un objet message structuré. Or, la méthode `base64` de cet objet ne préserve pas le nombre de caractères par ligne de la forme codée en `base64` d'une partie de message. Ceci n'altère pas le décodage mais affecte bien entendu la signature de celui-ci. Sympa manipule dorénavant un objet message mais il en conserve une image fichier.
- L'option d'abonné "digest" permettant de recevoir une compilation périodique des messages d'une liste, est incompatible avec le chiffrement en particulier parce qu'un « digest » peut contenir des messages en clair et des messages chiffrés. La version actuelle de Sympa remplace dans le « digest » les messages chiffrés par une indication qu'un message chiffré a été diffusé dans la liste (il est aussi possible d'interdire l'option `digest liste` par liste).

### 3.2.4 Obstacles relatifs aux archives et à la signature

Il est difficile de maintenir la compatibilité entre les archives WEB et les messages S/MIME : la signature doit être vérifiée lors de la lecture du message. En effet, si l'on vérifie les signatures seulement lors de la réception du message, rien ne garantit que le message n'a pas été modifié entre la vérification de la signature et la lecture de celui-ci. Dans le cas d'archives, le délai entre l'ajout du message dans les archives (dès sa réception) et sa lecture est le plus souvent très long et les risques de modifications du message importants. Par ailleurs, même si malgré un coût élevé, on modifiait le système d'archivage pour convertir en `html` les messages au moment de leur consultation et non au moment de leur réception, pourrait-on pour autant présenter la signature du message alors que l'affichage d'une forme `HTML` d'un message initialement en `text/plain` constitue objectivement une altération de celui-ci ?

<sup>2</sup> Vous aussi soyez méfiant avec les messages signés, il est possible d'en changer la date, le sujet, la liste des destinataires sans en altérer la signature !

### 3.2.5 Obstacles relatifs aux archives et au chiffrement

Le message est archivé sous sa forme chiffrée. Même si l'utilisateur consultant les archives le fait via une session HTTPS avec certificat du client permettant de garantir qu'il est un destinataire autorisé pour ce message, est-il raisonnable d'afficher en clair (dans une session chiffrée) un message confidentiel ? Dans un tel cas par exemple, le cache local du client contiendrait une image en clair du message. Nous préférons introduire une nouvelle fonctionnalité des archives permettant de se faire renvoyer un ancien message et y inclure les algorithmes de chiffrement de la diffusion ordinaire aux abonnés. Cette solution permet aussi au destinataire de vérifier une éventuelle signature du message.

### 3.2.6 Liste de révocation, expiration de certificat

A ce jour (octobre 2001), la version de développement de Sympa intègre enfin un gestionnaire de tâches périodiques. Celui-ci est conçu pour la programmation d'événements internes tel que l'envoi d'un rappel automatique des abonnements de chacun ou la mise à jour des signatures du moteur antiviral. Il est aussi possible de l'utiliser des pour gérer la mise à jour des listes de révocations de chaque autorité de confiance. Celles-ci sont alors partagées avec le serveur Apache tout comme la liste des autorités de confiance. Une tâche périodique permet en outre de prévenir l'expiration du certificat d'une liste en alarmant le listmaster et les propriétaires de la liste concernée. En effet, l'expiration du certificat d'une liste empêche Sympa de signer correctement le message de bienvenue et surtout, empêche les abonnés de poster des messages chiffrés à destination de la liste.

Enfin, même si cette fonction revient plutôt à l'autorité de certification émettrice, il est prudent de programmer l'envoi d'un avertissement à chaque utilisateur dont le certificat arrive bientôt à expiration permettant à celui-ci de renouveler son certificat (un message quelconque signé avec le nouveau certificat et adressé à Sympa permet à celui-ci de remplacer l'ancien certificat).

## 4. Et PGP, GPG ?

Nous sommes souvent critiqués sur le fait d'avoir choisi S/MIME plutôt que PGP pour un logiciel libre. Quelles sont les raisons de notre choix ?

Convenons que S/MIME est une norme ouverte et que OpenSSL n'est pas moins libre que GnuPG !

Il est bien évident que Sympa est avant tout le produit de liste que nous utilisons dans nos universités, avant d'être un produit que nous diffusons. Nos développements dans Sympa se doivent d'être cohérents avec nos activités dans le domaine des PKI. Au delà de cet opportunisme, il importe que les solutions d'authentification dans Sympa s'appliquent aussi bien sur son interface WEB que sur son interface messagerie. C'est possible avec des technologies basées sur des certificats X509, cela ne l'est pas avec PGP. Est-il vraiment intéressant de sécuriser la moitié d'un produit ?

Cependant, certains utilisateurs ont marqué leur intérêt pour le support de GPG dans Sympa. Le système des scénarii et la simplicité de l'API de chiffrement que nous utilisons avec S/MIME rendrait très simple l'introduction GPG dans Sympa et nous sommes très ouverts à une telle contribution à Sympa.