La signature des messages : une réponse contre le spam ?

François Morris
IMPMC, CNRS, Université Pierre et Marie Curie
Case 115
4 place Jussieu
75252 PARIS cedex 05
Francois.Morris@impmc.jussieu.fr

Résumé

Généralement les auteurs de spam agissent masqués en utilisant des adresses usurpées. DKIM est un protocole nouveau basé sur la signature électronique qui permet de protéger à la fois le contenu du message et son en-tête. Il a été conçu pour être un outil dans la lutte contre le spam. Les protocoles comme PGP ou S/MIME sont éprouvés mais ont montré leurs limites. DKIM, en déplaçant la signature au niveau du MTA, se veut résolument plus simple à mettre en oeuvre tant du point de vue technique que de ses implications en matière d'organisation. A la question fondamentale que constitue la confiance et la réputation que l'on peut accorder à l'émetteur d'un message répond un engagement du signataire qui applique et, ce qui est relativement nouveau dans Internet, publie sa politique de contrôles sur les messages émis. Si DKIM n'est sûrement pas la panacée, il implique une notion de responsabilité des différents acteurs ce qui a un côté très fortement structurant sur l'architecture du système d'information.

Mots clefs

Courrier électronique, signature électronique, spam, pourriel, DNS

1 Introduction

La confiance dans le courrier électronique est ternie par les nombreux abus que constituent le spam[1], l'hameçonnage[2], les vers et autres codes malveillants. On constate que la plupart des messages non désirés ou franchement hostiles sont émis avec des adresses d'expéditeur usurpées ou ne correspondant à personne.

L'architecture actuelle et les protocoles utilisés pour la distribution du courrier électronique ne permettent pas de s'assurer de l'identité de l'expéditeur d'un message reçu ou que le message envoyé est bien distribué au destinataire spécifié et uniquement à celui-ci.

Différentes approches ont été proposées pour lutter contre ce fléau. Elles se regroupent en trois catégories :

- Analyse du contenu du message (en-tête + corps) : logiciels anti-spam comme spamassassin.
- Vérification si le message est bien émis par une machine autorisée : SPF [1], Sender ID [2][3].
- Signature électronique.

Diverses considérations font que les autres approches comme le défi-réponse ou l'introduction d'un coût pour l'émetteur d'un message ont peu de chances d'être déployées sur une large échelle.

Seules les techniques cryptographiques de signature électronique qui permettent de sécuriser les échange et d'établir la confiance seront traitées ici. Le double défi consiste à les mettre en œuvre sans remettre en cause l'existant et sans avoir nécessairement à installer une lourde infrastructure pour gérer la confiance comme avec les IGC (Infrastructure de Gestion de Clés).

Dans les premières implémentations PGP ou S/MIME, le message est signé par son expéditeur et vérifié par son destinataire. Cela a montré ses limites, aussi il a été proposé de déléguer les tâches de signature et vérification aux MTA proches de l'émetteur et du destinataire. C'est l'objet d'une nouvelle proposition de standard DKIM [3].

2 Limites des protocoles actuels

Afin de faciliter la compréhension de ce qui va suivre, d'analyser les faiblesses conduisant aux problèmes actuels et de préciser les termes employés, il paraît utile de procéder à quelques rappels sur le fonctionnement du courrier électronique dans Internet et sur les standards employés ainsi que sur les mécanismes mis en œuvre lors de la signature électronique.

2.1 Architecture de distribution du courrier électronique

Comme défini dans le RFC 2822 [4], un message est constitué de deux parties séparées par une ligne vide : l'en-tête et le corps du message. L'en-tête comprend différents champs. Chaque champ est composé de son nom suivi du caractère « : » et de sa valeur. Le corps du message est un simple texte en caractères 8 bits. Afin de pouvoir transmettre des documents d'autres types ou structurés en différentes parties, il a été définie une méthode permettant de les convertir en une suite de lignes contenant des

[1] Le terme de pourriel formé à partie de poubelle et courriel a été proposé pour traduire « spam ».

[2] Phishing en anglais

[3] Les implications de ces protocoles en matière d'organisation, de déploiement, de politique de sécurité sont proches de celles pour DKIM.

caractères ASCII. Il s'agit de MIME (« Multipurpose Internet Mail Extensions ») [5].

Les messages sont relayés de MTA[4] (« Messages Transfer Agent » ou « agent de transfert de messages ») en MTA en utilisant le protocole SMTP [6]. De fait il s'agit d'un transfert de fichiers, au message (en-tête et corps) on ajoute une enveloppe contenant l'expéditeur, le destinataire et une identification du relais qui expédie le message. Ces informations sont utilisées pour permettre le routage du message vers le destinataire final.

Le MDA (« Message Delivery Agent) » distribue le courrier dans les boites aux lettres des utilisateurs, souvent sa fonction est confondue avec celle du dernier MTA dans la chaîne conduisant à l'utilisateur. Le MUA (« Message User Agent ») est l'outil permettant à l'utilisateur de gérer ses messages : consulter, composer, envoyer. Généralement le MUA utilise le protocole POP3 [7] ou IMAP [8] pour récupérer les messages reçus et SMTP pour en envoyer de nouveaux..

2.2 Expéditeur et destinataire

Différents champs : From, Sender, Resent-From, Resent-Sender, Reply-To, List-ID se rapportent à l'expéditeur d'un message. Ils sont plus ou moins bien définis dans le RFC 2822 et l'usage qui en est fait par les différentes implémentations de MUA et MTA est pour le moins fluctuant. Le RFC 2821 définit aussi un expéditeur « MAIL FROM » et l'identité du MTA relais « HELO ».

De même les champs To, Cc, Bcc, Resent-to, Resent-cc, Resent-bcc du RFC 2822 auquel il faut ajouter le « RCPT TO » de l'enveloppe déterminent le destinataire.

La signature électronique introduit un nouvel intervenant, le signataire du message qui n'est pas nécessairement l'expéditeur.

Aucune règle ne définit clairement les relations qu'il doit y avoir entre les informations dans l'enveloppe et celles dans l'en-tête du message alors que le bon sens aurait voulu qu'il y ait accord entre ces deux couches.

En l'absence de règles strictes, le développement du courrier électronique avec notamment des applications comme le renvoi de messages, les listes de diffusion, a conduit à une utilisation assez anarchique de ces différents paramètres. Le résultat est qu'aujourd'hui essayer de vérifier la cohérence des différents paramètres pour se prémunir contre les usurpations d'identité est totalement illusoire car on est quasi certain de bloquer un usage légitime.

Il s'agit pour l'essentiel de protocoles conçus à une époque où l'on se préoccupait fort peu de sécurité. Il faut dire à la décharge des concepteurs que c'était une époque où les différents acteurs s'efforçaient tous d'œuvrer au bon fonctionnement du réseau et que le matériel n'avait pas une puissance suffisante pour mettre en œuvre des mécanicismes élaborés de sécurité. La situation a hélas aujourd'hui bien changé et l'on se retrouve face à des individus malveillants dont il faut se protéger. Des techniques pour améliorer grandement la sécurité existent, notamment celles employant la cryptographie, mais leur mise en œuvre sans remettre en cause l'organisation existante est délicate. Clairement toute solution qui ne serait pas compatible avec l'architecture existante ou imposerait par exemple le remplacement de tous les MUA est exclue

2.3 Principes de la signature électronique

L'aspect le plus connu de la cryptographie concerne la confidentialité. L'expéditeur va chiffrer le corps du message et l'envoyer au destinataire en y ajoutant des informations sur la façon dont a été effectué le chiffrement. Le destinataire va recevoir et déchiffrer le message.

La cryptographie est aussi utilisée pour l'authentification et le contrôle d'intégrité. Pour s'assurer qu'un message n'a pas été modifié en cours de transmission et que son expéditeur est bien celui qui le prétend, on va utiliser les techniques de signature électronique de documents. Le principe est le suivant. L'expéditeur calcule une empreinte du message à l'aide d'une fonction de hachage. Cette fonction possède les propriétés suivantes :

- Elle est simple à calculer.
- Son résultat, l'empreinte a une longueur fixe (par exemple 128 bits pour MD5, 160 pour SHA-1).
- Connaissant un texte, il est difficile[5] de trouver un autre texte ayant la même empreinte.
- Connaissant une empreinte, il est difficile de trouver un texte correspondant à cette empreinte[6].

L'empreinte est chiffrée avec la clé privée de l'expéditeur pour constituer la signature du message. Le texte du message est alors envoyé en y adjoignant sa signature. Le destinataire va d'une part calculer l'empreinte du texte du message reçu et d'autre part déchiffrer la signature en utilisant la clé publique de l'expéditeur. Si ces deux valeurs concordent, le destinataire est sûr que le message n'a pas été modifié en cours de transport et que l'expéditeur possédait bien la clé privée correspondant à la clé publique. La façon dont on s'assure qu'une clé

[4] Par la suite nous utiliserons les acronymes anglais

[5] Difficile est toujours relatif à un moment donné. Cela signifie qu'actuellement avec les méthodes connues et le matériel disponible il est pratiquement impossible de résoudre le problème.

[6] Des travaux récents ont montré que les fonctions cryptographique de hachage utilisées aujourd'hui comme MD5 et SHA-1 n'étaient peut-être pas aussi robustes que l'on pensait. Outre que les implications pratiques restent à évaluer cela ne remet pas en cause fondamentalement le principe de la signature électronique mais conduit à s'intéresser plus aux propriétés des fonctions de hachages pour les améliorer.

publique est bien celle de l'individu considéré est fondamentale et constitue l'essentiel de la problématique liée à la signature électronique.

3 Signature par le MUA

La première idée concernant la sécurisation des messages a été de considérer le courrier électronique comme un simple transfert de fichiers n'offrant aucune sécurité. C'est alors à l'expéditeur et au destinataire ou plus exactement aux MUA d'effectuer les opérations cryptographiques permettant de sécuriser l'échange de messages dans ce canal non fiable.

Pour la signature électronique des messages, il existe deux familles de protocoles PGP et S/MIME. La différence essentielle résulte dans l'utilisation de certificats et le recours à une IGC pour S/MIME. Il existe une variante de PGP ou les informations de signature sont codées dans une partie MIME, il s'agit de PGP/MIME défini dans le RFC 3156.

3.1 PGP

PGP [9] utilise des lignes séparatrices en clair[7] comme

'----- BEGIN PGP SIGNED MESSAGE -----'

ou

'----- BEGIN PGP SIGNATURE -----'

dans le corps du message au sens du RFC 2822 pour distinguer le texte du message proprement dit de sa signature. Cela permet à un humain ne disposant pas d'un MUA gérant les signatures PGP de lire sans problème le texte du message et d'ignorer sa signature.

Afin de pouvoir vérifier la signature le destinataire doit récupérer, généralement à partir d'un serveur, la clé publique de l'expéditeur à moins qu'il ne la possède déjà. Une fois en possession de la clé publique, il faut décider si on fait confiance on non à cette clé ou plutôt à son émetteur. Si cette clé a été signée par une ou plusieurs personnes auxquelles je fais confiance, alors je pourrai faire confiance à cette clé. On a pu dire que le modèle de confiance de PGP est un modèle tribal, les amis de mes amis sont mes amis.

3.2 S/MIME

S/MIME [10] utilise les facilités de structuration de document de MIME pour spécifier la signature. Un humain ne disposant pas d'un MUA gérant S/MIME verra cette signature sous la forme d'un document attaché de type « application/pkcs7-signature » ce qui à l'usage se révèle assez perturbant.

S/MIME utilise des certificats x509 pour stocker les clés publiques ce qui impose l'utilisation d'une IGC. Le certificat de l'expéditeur est inclus avec la signature ce qui évite d'avoir à le récupérer à partir d'un serveur. Le destinataire pourra alors contrôler l'intégrité du message et vérifier l'identité de l'expéditeur s'il a auparavant récupéré par un canal sûr et installé le certificat de l'autorité de certification de l'expéditeur.

3.3 Limites

PGP et S/MIME existent, sont implémentés dans différents MUA depuis plusieurs années et fonctionnent plutôt bien. L'expérience a montré que la signature résistait assez bien au transfert entre les différents MTA. Cependant il faut bien constater que leur déploiement reste assez limité. La coexistence de deux standards avec des philosophies différentes a constitué incontestablement un frein au développement. Les querelles que l'on pourrait qualifier de théologiques entre les partisans de l'un ou l'autre standard n'ont pas arrangé les choses.

Dans les deux cas, seul le corps du message est signé, l'en-tête est exclu de la signature. Il est donc possible d'intercepter un message signé, de modifier son en-tête en changeant l'objet par exemple ce qui peut tromper le destinataire. En principe les MUA vérifient que l'adresse de l'expéditeur du message est bien celle qui figure dans son certificat, cependant en pratique ce contrôle n'est pas strict pour tenir compte des situations particulières que l'on rencontre dans la réalité. Tout ceci est loin d'être idéal lorsque l'on s'attache à éviter les usurpations d'identité pour lutter contre le spam. Il serait certes possible d'inclure le message avec son en-tête dans une partie MIME de type « message/rfc822 » afin de contrôler aussi l'en-tête mais cela exigerait une modification des MUA.

Les concepteurs de PGP ou ceux de S/MIME avaient comme objectif premier d'assurer la confidentialité, ce qui signifie faire du chiffrement. La signature électronique s'intègre tout naturellement dans les schémas et protocoles utilisés pour le chiffrement. Par contre il est probablement possible de concevoir des protocoles plus simples si on ne traite que de la signature en ignorant le chiffrement.

La signature est de la responsabilité personnelle de l'expéditeur et la vérification celle du destinataire. En particulier chaque utilisateur doit spécifier les clés PGP des expéditeurs ou les certificats des autorités de certification auxquels il fait confiance. Un organisme qui voudrait imposer à ses utilisateurs, une politique dans ce domaine aura beaucoup de difficultés à la déployer sur l'ensemble des postes de travail. En effet cela exige de distribuer et paramétrer les MUA ainsi que gérer les clés.

4 Signature par le MTA : DKIM

La signature électronique reste une idée séduisante pour lutter contre les usurpations d'identité et par conséquent

[7] Si on utilise MIME (RFC 3156) ces lignes n'existent pas et la structuration du message est analogue à celle de S/MIME cf. infra.

contre le spam. Aussi des propositions ont été faites pour mettre en œuvre une solution plus simple que PGP ou S/MIME et n'impliquant pas de modifier les MUA. Cela a fait l'objet d'un groupe de travail à l'IETF dénommé MASS[8] [11] qui a débouché sur une proposition de standard[9] DKIM [3]. Par la suite nous nous intéressons uniquement à celui-ci, le seul qui semble pouvoir s'imposer. DKIM[10] résulte de la fusion de Domain Keys de Yahoo et de Identified Internet Mail (IIM) de Cisco.

4.1 Objectifs

Pour qu'une nouvelle technique puisse être réellement déployée, il faut minimiser les modifications du système actuel. Les MUA ne seront pas concernés, seuls les MTA et encore seulement certains d'entre eux auront à être mis à niveau. En effet si modifier tous les MUA est totalement irréaliste, la mise à jour du logiciel sur certains relais de messagerie est plus acceptable.

Pour pouvoir s'assurer de façon totalement incontestable de l'expéditeur d'un message ou a fortiori fournir une preuve légale, la signature électronique doit être implémentée en respectant un certain nombre de critères stricts. PGP ou S/MIME doivent pouvoir répondre à cet objectif. S'il s'agit simplement de pouvoir rejeter les messages dont l'identité de l'expéditeur a été usurpée[11], il est possible de s'affranchir de certaines contraintes et simplifier grandement l'implémentation.

Il n'y a aucune obligation de signer les messages ce qui assure la compatibilité avec l'existant.

4.2 DKIM-Signature

Par rapport au RFC 2822 DKIM définit un nouveau champ dans l'en-tête d'un message : « DKIM-Signature: »[12]. La valeur de ce champ est une liste de paramètres séparés par des « ; ». Chaque paramètre est écrit sous la forme suivante : nom du paramètre suivi du caractère « = » puis de sa valeur. On trouvera en annexe un exemple.

[8] Le groupe de travail n'est pas encore officialisé au sein de l'IETF. Les personnes intéressées se sont rencontrées au cours d 'une « BOF » au dernier congrès de l'IETF et participent à une liste de diffusion.

[9] Comme pour toute proposition de standard à l'IETF il pourra y avoir des modifications. Rien ne garantit non plus que DKIM sera finalement accepté et encore moins qu'il sera réellement implémenté par les différents acteurs.

[11] Dans les listes de discussion, on voit se manifester certaines opinions avec des propositions pour augmenter la valeur probante de la signature. A mon avis cela impose un complexité accrue au niveau technique et encore plus de l'organisation qui ne peut conduire qu'à un échec lors du déploiement.

[12] Pour DomainKeys le seul qui soit réellement déployé (Google et Yahoo l'utilisent), le champ correspondant est « DomainKey-Signature: ».

4.3 Signature

Le message est signé par un MTA proche du MUA qui a envoyé le message. Du fait de sa proximité le MTA a accès à suffisamment d'informations pour déterminer si l'adresse de l'expéditeur du message correspond bien à l'individu connecté. Le MTA ne modifie aucunement le corps du message, il se contente d'ajouter dans l'en-tête un champ spécifique décrivant les informations permettant de vérifier la signature.

La signature est vérifiée par un MTA proche du MUA du destinataire. Celui-ci va extraire de l'en-tête les informations permettant de récupérer (généralement à l'aide d'une requête DNS) la clé publique du MTA signataire, les algorithmes cryptographiques utilisés ainsi que la signature. Il procédera ensuite à la vérification de la signature. Suivant la présence ou non d'une signature, la validité de celle-ci et en fonction de la politique définie le message va être refusé, accepté et transmis au destinataire ou bien marqué comme suspect.

L'algorithme de signature est défini par le paramètre « a= ». Actuellement le seul défini est « rsa-sha1 » signifiant que l'on utilise SHA-1 comme fonction de hachage et RSA pour le chiffrement asymétrique. Il faut bien comprendre qu'avec les progrès de la cryptanalyse et l'augmentation des performances des machines aucun algorithme ne peut être considéré comme fiable indéfiniment. Il faut donc prévoir de pouvoir utiliser différents algorithmes, il s'agit là d'une complication inévitable.

La signature est codée en base64 pour ne contenir que des caractères ASCII. Elle est spécifiée par le paramètre « b= ».

4.4 Ce qui est signé

Ce qui est signé est constitué du corps du message ainsi que certains champs de l'en-tête. La liste ordonnée des champs signés est spécifiée par le paramètre « h= » dans le champ définissant la signature. Le champ « DKIM-Signature » est toujours inclus dans la signature à l'exception de la valeur du paramètre « b= ». Il s'agit de protéger les informations servant à vérifier la signature. La signature elle-même qui ne peut bien évidemment pas être incluse dans ce qui va être signé est protégée en soi par ses propriétés cryptographiques.

Souvent les MTA ne se contentent pas d'ajouter comme c'est normal certains champs mais procèdent à une réorganisation plus ou moins complète de l'en-tête. Le fait de spécifier les champs à signer ainsi que leur ordre permet d'avoir une signature résistant bien au relais entre les différents MTA. Par rapport à PGP ou S/MIME où seul le corps du message était signé, DKIM apporte la sécurité de savoir que des informations importantes comme la date, l'expéditeur, le destinataire ou l'objet du message n'auront pas été modifiées en cours de transit.

Le paramètre optionnel « z= » contient la liste des champs signés dans l'ordre spécifié par « h= » avec leur valeur et telle qu'elle a été présentée à la signature. Il n'a pas à être utilisé pour la vérification, il s'agit uniquement d'une information utilisée pour la mise au point.

Il est possible de préciser qu'au delà d'une limite spécifiée par le paramètre « l= », le corps du message sera exclu de la signature. Ce qui peut être utile pour les listes de diffusion ou les MTA qui ajoute une publicité en fin de message. Cependant cette fonctionnalité reste très controversée. En effet il serait possible à un individu malveillant d'ajouter, sans que cela soit détectable, son texte à la fin d'un message légitime et d'en changer alors la signification, cela d'autant plus qu'on ne peut exclure la possibilité avec certains MUA de masquer le message initial en jouant des facilités de mise en page offertes par certains formats de textes enrichis.

Avant d'être signé le texte est d'abord réduit à une forme canonique. La méthode utilisée est définie par le paramètre « c= ». Le but de cette opération[13] est de se prémunir des réécritures effectuées par certains MTA qui notamment suppriment ou ajoutent des espaces, changent le repliement des lignes dans les en-têtes. Deux algorithmes sont initialement proposés. Le premier « simple » se contente de supprimer les lignes vides en fin de message. Le second « nowsp » supprime tous les espaces et convertit en minuscule les noms des champs dans l'en-tête. Il a été avancé qu'en changeant le découpage des lettres en mots il serait possible de changer le sens d'un message. Si pour un texte créé spécialement à cet effet, c'est envisageable, cela semble irréaliste pour un message quelconque.

4.5 Récupération de la clé publique

Le MTA qui vérifie une signature doit récupérer la clé publique du signataire.

Le paramètre « d= » spécifie le domaine qu'il faut interroger pour récupérer la clé.

Le paramètre « s= » permet de préciser un sous domaine dans le domaine défini par « d= ». Il s'agit essentiellement d'un moyen permettant de publier plusieurs clés pour un domaine. DKIM ne possédant pas de mécanisme pour la durée de validité ou la révocation des clés, le fait de générer, d'utiliser pour la signature et de publier une nouvelle clé résout le problème.

Le paramètre « q= » définit la méthode pour récupérer la clé publique, par défaut il s'agit de « dns » ce qui signifie que la clé est rangée dans un enregistrement DNS. Le paramètre « i= » permet de préciser sous forme d'une adresse électronique l'identité de celui pour le compte de qui le message est signé. La partie nom à gauche du caractère « @ » peut être vide. Le partie domaine de l'adresse doit être identique à ce qui est spécifié dans le paramètre « d =». Il n'est pas obligatoire que cette identité corresponde à un champ de l'en-tête du message. Le fait que dans la complexité du monde réel ce paramètre puisse réellement apporter une sécurité reste controversé.

Les paramètres « t= » et « x= » permettent de définir respectivement la date de la création de la signature et sa limite de validité. Une signature expirée ne doit pas être considérée comme valide. Il faut noter le caractère éphémère des signatures, la vérification est effectuée par un MTA lors du transit du message ce qui va prendre au pire quelques jours. Nous sommes bien loin de la signature électronique d'un document officiel qui doit pouvoir être vérifiée pendant plusieurs années.

Pour le DNS, la seule méthode définie aujourd'hui, on fera une requête au sous-domaine « _domainkey » du domaine défini précédemment. Par exemple avec

DKIM-Signature : q=dns ; d=exemple.fr; s=sep2005.abc ; i=@abc.exemple.fr

la requête DNS se fera sur

sep2005.abc._domainkey.exemple.fr

4.6 Clé publique

Pour stocker la clé, il a été prévu de définir ultérieurement un enregistrement RR de type « DKK » dans le DNS. En attendant, on utilise un enregistrement de type « TXT ».

Le paramètre « k= » définit le type de clé, la valeur par défaut qui doit être supportée par tout le monde est « rsa ».

Le paramètre « h » spécifie les algorithmes de hachage autorisés pour calculer l'empreinte. Par défaut ils le sont tous. Il faut obligatoirement supporter « sha1 ».

Le paramètre « p= » est la valeur codée en base64 de la clé publique. S'il est vide cela signifie que la clé a été révoquée.

Le paramètre optionnel « s= » permet de restreindre l'usage de la clé à certains services comme le courrier électronique. Il joue un rôle analogue à celui de l'attribut « X509v3 Key Usage » dans les certificats x509.

Le paramètre optionnel « g= » définit la granularité de la clé. Le but est de déterminer quelle adresse de signataire a légitimement le droit d'utiliser cette clé. Si cette valeur ne correspond pas à celle du paramètre « i= » dans le champ « DKIM-Signature: » le MTA qui vérifie doit considérer la signature comme invalide. La valeur « * » qui est celle par défaut signifie que toutes les adresses sont valides.

Le paramètre optionnel « t= » a pour valeur « y » si le domaine se content de tester DKIM. Le MTA qui vérifie ne doit pas alors faire de distinction entre du courrier signé ou non signé. Le déploiement de DKIM est une opération suffisamment complexe où l'on a vite fait d'oublier des cas

[13] En français canoniser et canonisation sont réservés au domaine théologique alors que canonique a aussi l'acception de forme simplifiée.

particuliers pour qu'il soit vivement conseillé de commencer par ce mode de fonctionnement.

5 Politique de signature par l'émetteur

Un point fort de DKIM est qu'un domaine émetteur de messages peut publier sa politique de signature : signer tous les messages, en signer certains, n'en signer aucun, ne pas envoyer de message.

Cela va permettre à un MTA vérificateur de décider du traitement des messages n'ayant pas de signature valide. Il devrait ainsi les refuser si l'émetteur signe toujours.

Un domaine qui implémente la politique de signer tous ses messages va en tirer plusieurs profits. D'abord il va rejeter automatiquement tous les messages portant une adresse usurpée de son propre domaine. Certes il serait possible de dire que les adresses portant son propre nom de domaine et provenant de l'extérieur sont usurpées, cependant les choses ne sont pas aussi simples, il suffit de considérer le cas des nomades, des renvois de messages et des listes de diffusions. Plus d'autres domaines vont implémenter DKIM moins il va recevoir de « bounce » avec ses propres adresses. Enfin il va accroître sa réputation vis à vis des autres acteurs d'Internet.

Pouvoir signer tous les messages émis avec sa propre adresse de domaine exige de connaître précisément d'une part quels sont les expéditeurs de messages sur le réseau et d'autre part quels sont les expéditeurs qui ont à émettre des messages depuis l'extérieur. A priori cela devrait être simple : tous les expéditeurs sur mon réseau portent une adresse correspondant à mon domaine et tous ceux qui ont une adresse correspondant à mon domaine sont sur mon réseau. La réalité est beaucoup plus complexe. Il y a les visiteurs que l'ont accueille sur le réseau, les nomades qui se connectent à l'extérieur. Il y a les listes de diffusion, les renvois de messages. L'analyse à ce sujet exigée par le déploiement de DKIM est une excellente chose en matière de sécurité car elle oblige à clarifier la situation et à définir une politique pour ces différentes situations et à mettre en œuvre des solutions ad hoc (VPN, webmail, SMTP+TLS).

Si on détecte un comportement répréhensible de la part d'un utilisateur, il n'est pratiquement pas possible de révoquer la clé ayant servi à la signature. En effet la même clé sert pour plusieurs expéditeurs, certes DKIM permettrait d'utiliser une clé par utilisateur mais ce serait inutilement coûteux et pas du tout dans l'esprit du protocole. En outre les mécanismes de cache du DNS retarderait la diffusion de la révocation ce qui rend plus douteux le fait qu'elle arrive avant la vérification message.

5.1 Publication de la politique

La politique est publiée en utilisant un enregistrement à définir « SSP » de type RR du DNS. En attendant, on utilise un enregistrement de type « TXT »[14]. Pour récupérer la politique le vérificateur fait une requête DNS à « _policy._domainkey.<domain>[15] » où « <domain> » est le domaine de l'expéditeur. Outre cette politique globale par nom de domine, il est possible de définir une politique par adresse individuel dans ce cas la requête se fera à « <local>._policy._domainkey.<domain> » où <local> est la partie locale (à gauche de @) de l'adresse.

S'il n'existe pas d'enregistrement pour <domain> on doit réitérer la recherche pour le niveau au dessus dans l'arborescence en s'arrêtant à la racine.

La syntaxe utilisée par cet enregistrement est la même que précédemment.

Le paramètre « o= » définit la politique. Les différentes valeurs possibles sont :

- « ~ » certains messages mais pas tous sont signés.
- « - » tous les messages sont signés, les messages non signé provenant de cette entité ne doivent pas être acceptés et les messages signés par une tierce partie devraient être acceptés.
- « ! » tous les messages sont signés, les messages non signé provenant de cette entité ne doivent pas être acceptés par contre les messages signés par une tierce partie ne devraient pas être acceptés
- « . » cette entité n'envoie pas de messages.
- « ^ » répéter la requête au niveau utilisateur.

Le paramètre « t= » avec la valeur « y » signifie qu'il s'agit d'une politique en phase de test et un MTA vérificateur ne devrait pas en tenir compte.

Il a été défini quelques paramètres optionnels supplémentaires. « n= » permet de spécifier un texte lisible par un humain pour préciser différentes informations. « r= » permet de donner l'adresse électronique à laquelle il fait s'adresser en cas de problèmes. « u » a été prévu pour pointer sur une URI qui donnera plus d'information sur la politique de sécurité.

5.2 L'engagement du signataire

Le Fournisseur d'accès Internet (FAI) qui signe un message envoyé par un de ses clients certifie que l'adresse électronique spécifiée dans l'en-tête du message émis correspond bien à une des adresses qu'il gère pour son

[14] Il s'agit d'une définition volontairement simpliste et intérimaire d'un protocole pour spécifier une politique. En effet c'est d'une notion nouvelle qui suscite beaucoup de discussions Par ailleurs le groupe de travail MARID (MTA Authentication Records In DNS) qui aurait pu fournir une réponse a été fermé.

[15] Il faut noter ici une divergence avec DomainKeys qui stocke cette information dans _domainkey.<domain>. DKIM ajoutant la possibilité d'une politique par utilisateur, il fallut distinguer les enregistrements contenant une politique de ceux contenant une clé.

client. A partir des données de connexion, il est possible de remonter au compte du client et donc à ses adresses, le MTA n'a plus qu'à vérifier si le champ « From: » de l'en-tête correspond bien.

Pour le responsable du système d'information d'un organisme qui signe les messages envoyés par ses employés l'engagement est plus fort. S'il garantit toujours l'identité de l'expéditeur, il certifie, au moins implicitement et en tout cas cela sera vu de cette façon par le destinataire, que l'expéditeur parle au nom de son organisme.

La gestion du courrier électronique et la signature en particulier peuvent être déléguées à un prestataire extérieur. Dans ce cas les responsabilités de chacune des parties doivent être bien précisées par un contrat.

5.3 Vérification de l'expéditeur

Comme vu précédemment, il est impératif de vérifier qui est le vrai expéditeur d'un message avant de le signer. Cela peut s'effectuer de différentes façons en fonction du contexte.

Si l'utilisateur envoie son message à partir d'une machine qui a déjà été authentifiée, il suffit de se rapporter aux données de connexion et aucune authentification supplémentaire n'est nécessaire. Cette situation se retrouve dans plusieurs configurations : connexion utilisant PPP (fréquemment utilisé par les FAI), authentification 802.1x sur un réseau local, réseau privé virtuel (VPN).

Lorsque la machine n'est pas authentifiée comme c'est le cas sur un réseau local, il faut bien authentifier l'expéditeur du message. La solution est d'utiliser l'ajout au protocole SMTP de l'option d'authentification [12]. Il faut alors prendre garde au fait que pratiquement il est difficile d'utiliser une méthode ne faisant pas circuler en clair le mot de passe. Cela peut être acceptable sur un réseau local, cela ne l'est assurément pas si l'on doit utiliser un réseau public. Dans ce cas, il faut utiliser la version sécurisée avec TLS de SMTP [13]. Il est alors envisageable et dans certains cas judicieux de remplacer le couple identifiant, mot de passe par un certificat client. De toute façon on n'échappera pas à l'obligation de mettre en œuvre une gestion des identités qui offre par ailleurs bien des avantages.

Un FAI qui expédie le courrier électronique de ses abonnés ou une entreprise ceux de ses employés devrait idéalement s'assurer avant de transmettre un message que celui-ci provient bien d'une adresse valide dans son domaine et a été envoyé par une personne dûment authentifiée. Si les différents acteurs d'Internet procédaient à cette vérification les spammeurs auraient une tâche bien plus difficile. Entre autres choses, les protocoles comme DKIM ou SPF/Sender-ID permettent à un MTA d'annoncer qu'il a effectué cette vérification et que par conséquent les messages en provenance de son domaine sont a priori plus fiables.

Sur la façon dont s'effectue cette vérification et sur les implications en matière d'architecture de réseau on se rapportera à [15].

6 Politique de vérification par le destinataire

La vérification par le destinataire du message reçu peut s'effectuer par le MUA de l'utilisateur ou/et le MTA de son organisation. D'une part parce que la mise en oeuvre est moins contraignante en terme de déploiement, d'autre part parce que la politique se définit et s'implémente au niveau de l'organisation, la vérification devrait s'effectuer essentiellement au niveau du MTA tout en laissant à l'utilisateur la décision finale dans les cas douteux.

Le bon endroit pour placer la vérification de la signature est sur le MTA le plus en amont possible et en tout cas avant les traitements anti-virus et anti-spam. En effet une vérification de signature est nettement moins coûteuse qu'une analyse de contenu à la recherche de virus ou de spam.

Il est judicieux que le MTA vérifiant la signature ajoute dans l'en-tête du message le champ « Authentication-Results » [16] le résultat de la vérification. L'anti-spam ou le MUA pourra utiliser utilement cette information pour classifier le message.

6.1 Résultat de la vérification

Différents cas se présentent :

- Message avec une signature invalide. En principe il s'agit d'un message falsifié et il faut le rejeter sans autre forme de procès. Cependant on ne peut exclure que l'un des MTA au cours du transit ait effectué une réécriture intempestive. Il semble sage dans un premier temps de ne pas éliminer totalement ces messages mais de les marquer comme très douteux ou bien de les mettre dans une zone de quarantaine et de procéder alors à une analyse humaine pour détecter un éventuel problème.
- Message non signé. Il faut consulter la politique publiée par l'entité expéditrice. Si celle-ci n'envoie aucun message ou signe tous ses messages, il faut évidemment rejeter le message reçu.
- Message ayant une signature valide. Le message n'a pas été falsifié et son expéditeur est connu. La foi que l'on peut avoir en lui est liée au degré de confiance dans le signataire et à la réputation de l'expéditeur.

6.2 Réputation

Ce n'est pas parce qu'un message est signé que l'on peut lui faire totalement confiance. En effet il suffit de quelques euros à un spammeur pour enregistrer un domaine, créer les bons enregistrements dans le DNS et envoyer des messages signés. On peut faire confiance aux spammeurs pour s'adapter, ce ne seront probablement pas les derniers à adopter DKIM. De même certains signataires seront plus ou moins regardants sur le contrôle de l'identité des expéditeurs. Un message provenant d'un inconnu et signé par son FAI aura a priori moins de valeur que celui d'un employé d'une organisation appliquant un strict code de bonne conduite.

DKIM garantit l'identité de l'expéditeur mais ne dit rien sur la confiance que l'on peut avoir en lui. Si cette question de réputation est sous-jacente dans beaucoup de discussions autour de DKIM, elle ne peut être incluse dans le protocole. Une entité pourra publier sa politique en la matière mais même si une norme définit la façon de présenter les choses, il faudra toujours un humain pour interpréter le document[16]. La réponse ne peut qu'être pragmatique. En fonction de son expérience on fait totalement, moyennement, un peu, pas du toute confiance aux messages provenant de telle entité. Cela passe par l'établissement de listes blanches, de listes noires, voire de notations pour les différents domaines. De fait les logiciels anti-spam actuels et les règles introduites par les utilisateurs pour filtrer leur courrier utilisent déjà de telles listes. La signature ne fera qu'en renforcer l'efficacité

6.3 Politique.

En fonction de son environnement, de ses objectifs, ses contraintes, le vérificateur va se définir une politique plus ou moins stricte pour le traitement des messages reçus.

Certains FAI ont un poids suffisant pour imposer qu'ils n'accepteront que les messages authentifiés par DKIM ou SPF/Sender-ID. Ainsi Microsoft vient d'annoncer qu'à partir de novembre 2005 tous les messages reçus par Hotmail ou MSN et non marqués avec Sender-ID seront considérés comme spam [14].

7 Questions ouvertes

7.1 Implication de l'utilisation du DNS

Le DNS est nécessaire pour gérer les clés utilisées par DKIM. Le DNS n'est pas sécurisé et de nombreuses attaques contre lui ont été démontrées [17]. La version sécurisée [18] devrait régler ces problèmes mais elle est loin d'être couramment déployée.

Il reste que si elle est parfaitement envisageable, une attaque de DKIM par l'intermédiaire du DNS est probablement trop coûteuse pour un simple expéditeur de spam. En effet il lui faudrait faire en sorte que sur assez longue période un nombre suffisamment grand de MTA récupère des enregistrements DNS falsifiés. Une prise de contrôle direct du serveur DNS serait alors bien plus efficace que l'attaque sur le protocole. Une attaque ciblée contre un MTA vérificateur pour lui faire récupérer de faux enregistrements DNS et ensuite lui faire accepter de faux messages et refuser les légitimes, serait bien plus réaliste. Il faut cependant préciser que pour ce dernier type d'attaque DKIM n'est pas la seule cible possible, falsifier un enregistrement MX est tout aussi efficace pour effectuer un déni de service, détournement de correspondance ou de l'hameçonnage (phishing) (faux serveurs webs).

Il faut rappeler que l'objectif de DKIM est de pouvoir raisonnablement être sûr de l'authenticité d'un message reçu et non d'avoir une preuve légale sur le contenu et l'auteur d'un document. PGP ou S/MIME sont faits pour cela. Cependant une migration vers DNSsec est certainement souhaitable.

Outre ces questions liées à la sécurité, il faut considérer la charge supplémentaire sur le DNS. Pour chaque message il y a une requête pour récupérer la clé, une autre (deux dans le cas d'une politique par utilisateur) pour récupérer la politique. Les mécanismes de cache du DNS devraient grandement arranger les choses.

Pour rester compatible avec les différentes implémentations de DNS, ne pas avoir de problèmes ave les éléments de filtrages sur le réseau., la taille pratiquement utilisable pour les enregistrements DKIM est sérieusement limitée (paquet de 512 octets) ce qui notamment impose une contrainte de 2048 bits à la longueur des clés utilisées.

7.2 Répétition

DKIM est plutôt mal protégé contre la répétition[17]. Considérons un individu qui souhaite diffuser du spam. Il va envoyer à lui-même un message à partir d'un FAI réputé qui signe tout ce qu'il émet. Il va récupérer son message avec l'en-tête et le diffuser sans rien modifier. Le message reçu par la victime aura alors une signature parfaitement valide.

Il est aussi possible à un individu malveillant d'intercepter un message signé et de l'envoyer tel quel à de nombreux destinataires ce qui ternira la réputation du malheureux expéditeur initial en le faisant passer pour un spammeur alors qu'il n'y est absolument pour rien.

[16] C'est la même situation qu'avec les IGC où un champs dans le certificat permet de référencer un document définissant la politique de certification.

[17] Replay en anglais

Ce qui est une faiblesse peur être considéré comme un atout dans un autre contexte. Ainsi les listes de diffusions qui se contentent de rediffuser les messages reçus n'altèrent pas les signatures DKIM ni S/MIME d'ailleurs.

Tout ceci ne peut fonctionner que parce que les standards ne définissent pas qu'il doit y avoir un lien entre le destinataire dans l'enveloppe (RCPT TO du RFC 2821) et celui dans l'en-tête du message (To, Cc, Bcc du RFC 2822). Une vérification de la cohérence entre ces informations permettrait théoriquement d'éliminer le risque puisque le destinataire de l'en-tête du message est signé et ne peut donc être falsifié. Cependant une application stricte de cette règle éliminerait un grand nombre de messages légitimes comme ceux envoyés par certaines listes de diffusion.

Tout n'est cependant pas si désespéré. Les mécanismes employés dans logiciels anti-spam fonctionneront d'autant mieux qu'on aura la certitude que l'adresse du destinataire n'a pas été falsifiée. Ensuite il sera difficile à l'expéditeur du message initial de dénier l'avoir envoyé ce qui peut avoir un caractère dissuasif.

Les messages signés avec PGP ou S/MIME sont aussi vulnérables à ce type de scénario. SenderID/SPF qui base son authentification sur le MTA expéditeur est probablement plus efficace dans cette situation.

7.3 Liste de diffusion

Rien ne ressemble plus à du spam qu'une liste de diffusion : envoi massif de messages, difficultés à déterminer quels sont l'expéditeur et le destinataire, la liste ou un individu. Généralement les méthodes proposées pour lutter contre le spam ont d'énormes problèmes avec les listes, DKIM ne fait pas exception à la règle.

Un gestionnaire de liste réécrit les messages en modifiant les en-têtes, en ajoutant souvent un texte au message initial. Tout ceci est difficilement compatible avec la signature.

Plus fondamentalement se pose la question pour le compte de qui faut-il signer : celui qui soumet le message ou bien la liste ? La réponse reste très discutable. Par exemple on pourrait dire que pour une liste modérée, il faut signer au nom de la liste tandis que pour une autre non modérée, il faut le faire au nom de celui qui a soumis le message.

A la différence de Sender ID, le renvoi de courrier (forwarding) ne pose a priori guère de difficultés. En effet seule l'enveloppe est concernée, le corps du message ou son en-tête à l'exception des champs de trace (Received, Return-Path) qui sont exclus de la signature, ne sont pas modifiés.

8 Conclusion

Le rétablissement de la confiance dans le courrier électronique implique la responsabilisation des différents acteurs de l'Internet. Cela passe par des techniques de signature comme DKIM qui permettent de s'assurer de l'identité des expéditeurs et de la non-falsification des messages. Du modèle libertaire des origines Internet évolue vers une situation où les contraintes réglementaires et légales ainsi que les relations contractuelles joueront un plus grand rôle.

Le fonctionnement du courrier électronique dans le monde réel résulte d'une implémentation plus ou moins stricte de standards qui ne sont eux même pas toujours très précis. Il faut donc être extrêmement prudent dans le déploiement de nouveaux protocoles comme DKIM car la moindre modification risque d'entraîner des effets de bord non prévus et indésirables.

La signature des messages est très fortement structurante pour le système d'information. Il faut impérativement authentifier l'expéditeur d'un message ce qui a des répercutions sur l'architecture [15].

La signature des messages ne rendra pas caduque l'utilisation de filtres anti-spam au niveau du réseau ou sur le poste de travail de l'utilisateur. Au contraire elle en renforcera l'efficacité car il sera possible de se baser sur information dont on est sûr qu'elle n'est pas falsifiée.

La signature par les MTA avec DKIM est plus simple à déployer que PGP ou S/MIME mais n'offre pas le même niveau de confiance que permet la signature par le MUA et l'utilisation d'une IGC. Il s'agit de deux solutions complémentaires plutôt que concurrentes.

Annexe

DKIM n'étant pas encore disponible, les exemples ci-dessous ont été produits avec DomainKeys. Pour des raisons de lisibilité et éviter de fournir des adresses aux spammeurs les messages ont été modifiés : suppression de champs non pertinents, troncature des clés et signatures, modification des adresses. Le message a été traité avec dkfilter [19], aucune action particulière n'est effectuée, le message est seulement transmis à l'anti-virus, anti-spam élément suivant sans la chaîne. Le résultat de la vérification est ajouté dans le champ « Authentication-Results » qui pourrait être utilisé par l'anti-spam.

```
Authentication-Results: mail.exemple.fr
 from=alice@yahoo.fr; domainkey=pass
Return-Path: <alice@yahoo.fr>
Received: from
web26404.mail.ukl.yahoo.com
 by mail.exemple.fr with SMTP
 for <bob@exemple.fr>; Wed, 7 Sep 2005
18:05:55 +0200 (CEST)
DomainKey-Signature: a=rsa-sha1;
 q=dns; c=nofws;
 s=s1024; d=yahoo.fr;
 h=Message-ID:Received:Date:From:
```

```
   Subject:To:MIME-Version:
   Content-Type:
   Content-Transfer-Encoding;
 b=a175OkTaidUHGwhKwgEP3n424qc=  ;
Message-ID: <1234@web26404.yahoo.com>
Received: from [134.157.17.225] by
 web26404.mail.ukl.yahoo.com via HTTP;
 Wed, 07 Sep 2005 18:06:04 CEST
Date: Wed, 7 Sep 2005 18:06:04
From: Alice <alice@yahoo.fr>
Subject: Essai
To: bob@exemple.fr
MIME-Version: 1.0
Content-Type: text/plain;
 charset=iso-8859-1
Content-Transfer-Encoding: 8bit

Ceci est un essai
```

L'enregistrement contenant la clé peut être récupéré par :

```
host -t TXT s1024._domainkey.yahoo.fr
```

qui retourne :

```
s1024._domainkey.yahoo.fr text "k=rsa\;
t=y\; p=MIGfMA0\; n=A 1024 bit key\;"
```

Le résultat avec un message volontairement altéré est :

```
Authentication-Results: mail.exemple.fr
 from=alice@yahoo.fr;
 domainkey=neutral
 (signature invalid;
  no policy for yahoo.fr)
```

Bibliographie[18]

[1] Sender Policy Framework http://www.ietf.org/internet-drafts/draft-schlitt-spf-classic-02.txt

[2] Sender ID, http://www.ietf.org/internet-drafts/draft-lyon-senderid-core-01.txt ; http://www.ietf.org/internet-drafts/draft-lyon-senderid-pra-01.txt

[3] DKIM, DomainKeys Identified Mail http://mipassoc.org/dkim/index.html

[4] RFC 2822, Internet Message Format.

[5] RFC 2045 à 2049, Multipurpose Internet Message Extensions

[6] RFC 2821, Simple Mail Transfer Protocol.

[7] RFC 1939, Post Office Protocol - Version 3

[8] RFC 3501, Internet Message Access Protocol

[9] RFC 2440, OpenPGP Message Format

[10] RFC 2633, S/MIME Version 3 Message Specification

[11] MASS, Message Authentication Signature Standards http://www3.ietf.org/proceedings/05aug/mass.html

[12] RFC 2554, SMTP Service Extension for Authentication

[13] SMTP TLS

[14] http://www.zdnet.fr/actualites/internet/0,39020774,39235930,00.htm

[15] Serge Aumont, Claude Gross. L'impact de la lutte contre le SPAM et les virus sur les architectures de messagerie. Dans *Actes du congrès JRES2005*, Marseille, décembre 2005

[16] Internet draft, Message Sender Authentication Header http://www.ietf.org/internet-drafts/draft-kucherawy-sender-auth-header-02.txt

[17] RFC 3833, Threat Analysis of the Domain Name System (DNS)

[18] RFC 4033, DNS Security Introduction and Requirements

[19] http://jason.long.name/dkfilter/

[18] Les textes des RFC sont disponibles sur le site de l'IETF http://www.ietf.org/rfc.html