REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

Département d'Electronique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme d'Ingénieur d'Etat en Electronique

Etude et développement d'une application pour l'envoi anonyme des messages électroniques

Proposé et dirigé par :

M^me^ L.HAMAMI

Présenté par :

M^r^ DRIFEL YACINE
M^r^ HABCHI Abdelaziz

Promotion : juin 2002

DEDICACES

Je dédie ce mémoire à toutes les personnes qui sont chères à ma vie.
A savoir

Mon père et ma mère qui m'ont permis d'atteindre ce niveau d'étude et qui ont contribué à ma réussite.

Ma petite sœur ainsi que mon petit frère que je leurs souhaite un très grand succès dans leurs études.

Mes chers enseignants : Abdelmadjid, Sadek, Ahmed, Laarbi.

L'équipe de la bibliothèque d'Ousama Ibn Zaid : Said, Redouane et Yacine.

Mes amis : parmi lesquels : Abdelaziz, Ferhat, Lies, Mahrez, Sidali, Toufik, Amine.c, Miad, Souhil, Kader, Amine.B, Rafik, Elhadi…….

YACINE DRIFEL

A ma Mère qui m'a entouré de chaleur,
A mon cher Père pour son soutient,
A tous mes frères et sœurs,
Ainsi qu'à tous mes amis,
Je dédie ce mémoire.

Abdelaziz HABCHI

REMERCIEMENTS

Nous remercions en premier lieu, DIEU le tout puissant de nous avoir permis de mener à bien ce modeste travail. Ensuite, nous tenons à exprimer nos vifs remerciements à toutes celles et à tous ceux qui nous ont soutenu tout au long de la préparation de ce travail, encouragé à le poursuivre et l'achever.

Nos remerciements vont à nos professeurs et plus particulièrement à notre promotrice M^{me} L.HAMAMI pour sa disponibilité, son engagement et ses précieux conseils.

Nos remerciements les plus distingués vont à M^{r} KHELLADI pour son aide précieuse, sa serviabilité et sa très grande gentillesse.

Nous remercions vivement M^{elle} GUERTI pour avoir présidé notre jury, ainsi que M^{r} BOUSBIA pour avoir examiné notre travail.

Enfin, nous remercions tous les membres de nos familles ainsi que tous nos amis pour leur soutien moral et matériel et leurs encouragements permanents.

Abstract:

Most security concerns focus on eavesdropping to prevent outsiders from listening in electronic conversations. Encrypted messages can still be tracked, revealing who is talking to whom. This tracking is called traffic analysis and may reveal sensitive information.

As the Web becomes an important part of modern day communication and electronic commerce, protecting the privacy of electronic messages becomes increasingly important. Just like mail, electronic messages travel in electronic envelopes. Protecting the privacy of electronic messages requires both safeguarding the contents of their envelopes and hiding address on their envelopes. Although communicating parties usually identify themselves to one another, there is no reason that the use of a public network like the Internet ought to reveal to others who is talking to whom and what they are talking about. In certain cases anonymity may be desirable: e-mail and Web browsing should not require revealing one's identity

The goal of this work is to create a system where people wanting to preserve their anonymity on Internet (electronic mail in particular) can use an infrastructure for encrypting and routing the messages and requests. The objective is to prevent the indiscreet ones from seeing the contents of the packages as to go up until their source by traffic analysis.

Key words: Security, cryptography, anonymity, confidentiality, traffic analysis, remailer.

ملخص:

تنتقل في الإنترنت حزم متبعة عدة مسالك من آلة إلى أخرى، تقوم هذه الآلات باتباع هذه المسالك باستعمال برامج معلوماتية متخصصة. يتغير الطريق حسب توفر الآلات و الارتباطات، وهذا ما يكسب الإنترنت قوة. لكن، لا يمكن ضمان عدم التقاط أو إتلاف الحزم المتبادلة من طرف متصلين.

الهدف الرئيسي من سرية المعلومة هو جعل هذه الحزم غير مفهومة من طرف أي جهة خارجية، و بالتالي سرية المعلومة تتوقف على حماية محتوى كل اتصال من الاستماع على الشبكة. ولكن لا نستطيع تجنب مراقبة الاتصالات التي تهدف إلى تحديد هوية المتصلين من خلال أي اتصال. فمثلا حماية الحياة الخاصة في البريد الإلكتروني، يتطلب حماية موضوع الرسالة وإخفاء العناوين، مع إبقاء إمكانية تعرف كل من المتصلين على الآخر إن أرادا ذلك.

لا توجد حجة لكشف المتصلين وعما يتحدثان، وهذا م يؤدي إلى ضرورة إخفاء هوية المتصلين و بالتالي الحفاظ على كتمها.

تتمحور الفكرة الأساسية لموضوع مشروعنا حول إمكانية إنشاء نظام خاص بالأشخاص الراغبين في كتم هويتهم عند استعمال البريد الإلكتروني في شبكة الإنترنت و هذا بواسطة وسائل الترميز وتغيير الاتجاه، من أجل منع الجواسيس من رؤية الحزم و معرفة مصدرها عن طريق مراقبة الاتصالات.

المفاتيح: الأمن، الترميز، كتم الهوية، سرية المعلومة، مراقبة الاتصالات.

Résumé :

Sur Internet, circulent des paquets qui suivent des circuits de routage de machines en machines. Chacune de ces machines effectue ce routage avec des logiciels standards aux spécifications bien connues. Le chemin varie en fonction de la disponibilité des machines et connexions. C'est ce qui fait la souplesse d'Internet. Mais, il est impossible de garantir que les paquets échangés entre deux correspondants ne sont pas interceptés ni altérés.

L'objectif principal de la confidentialité est d'empêcher que ces paquets soient compréhensibles par une entité tierce non autorisée. En effet, la confidentialité consiste à protéger le contenu de la communication de toute écoute sur le réseau. Cependant, elle n'empêche pas l'analyse du trafic qui a pour but de déduire à travers une communication les identités des communicants.

Donc, protéger la vie privée dans messagerie électronique par exemple, nécessite la protection du contenu des messages et la dissimulation des adresses. Les parties communicantes s'identifient l'une à l'autre : il n'y a pas de raison de divulguer à tous les autres qui parle à qui et de quoi parlent-ils. Dans certaines applications, il est nécessaire de cacher les identités des communicants et donc de préserver l'anonymat.

Les questions d'anonymat surgissent à de nombreuses occasions sur Internet, notamment pour la messagerie électronique et la navigation.

Le but de ce travail est de créer un système où les personnes voulant préserver leur anonymat sur Internet (la messagerie électronique en particulier) peuvent utiliser une infrastructure pour crypter et router les messages et requêtes. L'objectif est d'empêcher des indiscrets de voir le contenu des paquets ainsi que de remonter jusqu'à leurs sources par analyse du trafic réseau.

Mots clés : Sécurité, cryptographie, anonymat, confidentialité, analyse de trafic, remailer.

Sommaire

CHAPITRE 4: Onion Routing

CHAPITRE 5: Les remailers

CHAPITRE 6: Réalisation

Introduction générale

L'homme de part sa nature sociable, cherche des moyens de plus en plus rapides et efficaces pour communiquer à distance et c'est durant le 20ème siècle qu'il a réalisé ses objectifs les plus obstinés. En effet, l'information et la télécommunication ont connu une évolution spectaculaire et ont concerné surtout la collecte, le traitement et la distribution de l'information. Aujourd'hui, pour rendre les ressources partageables à la portée de chaque utilisateur et assurer une grande fiabilité aux ressources employées, de nombreux réseaux ont vu le jour utilisant les services offerts par la microinformatique et profitant de l'avènement des systèmes distribués.

Plus ambitieux, l'homme cherchera à relier des réseaux hétérogènes entre eux. C'est dans ce contexte qu'Internet a évolué et constitue de nos jours un tissu planétaire de services et produits, et un moyen très efficace pour interagir et diffuser les informations. Cependant, l'utilisation de ce réseau mondial a rendu les systèmes informatiques au sein des entreprises, laboratoires de recherche et gouvernements de plus en plus ouverts à d'éventuelles menaces par un intrus trouvant son intérêt : dans la destruction de ces systèmes, dans l'interception des informations trafiquant entre eux ou dans l'identification de l'identité des communicants touchant ainsi à leur droit légitime et légal à l'anonymat. Et si les différentes techniques de cryptographie et d'authentification sont actuellement les moyens les plus efficaces utilisés pour protéger la confidentialité et l'intégrité des données transitant entre les réseaux interconnectés, il s'est avéré qu'elles ne peuvent pas empêcher l'analyse du trafic qui a pour but de déduire à travers une communication les identités des communicants menaçant ainsi leur anonymat.

En effet, qui n'a jamais reçu de mail concernant la vente par correspondance provenant de sites totalement inconnus ? Qui a bien pu récupérer votre adresse e-mail à vos dépend ? Ce genre d'infraction est utilisé à des fins commerciales dont le but est de prendre contact avec le plus de clients potentiels possible et de connaître les données personnelles des consommateurs afin de toucher la clientèle cible. On voit alors tout de suite que le profiling (connaissance de l'intérêt et des goûts des consommateurs) est tout à fait contraire au respect de la vie privée sur Internet.

Le service mail est actuellement le moyen le plus populaire de communications, mais en même temps le moins sécurisé compte tenu des nombreuses informations transportées avec

les messages, qui peuvent être facilement récupérées par des techniques efficaces permettant l'analyse du trafic de données.

Le but de notre travail consiste donc, à étudier d'abord la notion d'anonymat, et surtout voir la possibilité d'assurer ce droit légitime et légal dans le monde électronique, puis proposer des solutions adéquates à ce sujet.

Notre document est structuré en six chapitres :

Le premier chapitre nous permet d'avoir une vue d'ensemble sur le concept de la sécurité informatique, afin d'introduire d'une manière générale les différentes notions nécessaires pour l'établissement de tout système informatique sûr et solide.

Le deuxième chapitre est consacré à l'étude des différents mécanismes de sécurité informatique tels que la cryptographie et l'authentification.

Le chapitre trois comporte la définition du concept de l'anonymat, sa relation avec la confidentialité ainsi que les différentes techniques mises au point pour assurer l'anonymat sur Internet. Cela afin d'arriver dans le chapitre 4, à décrire une des techniques les plus efficaces pour garantir un niveau d'anonymat très élevé; il s'agit d'Onion Routing, méthode faite pour être utilisée dans plusieurs services Internet.

Le chapitre cinq comporte la description des remailers : serveurs implémentés spécialement pour protéger l'anonymat dans l'envoi des courriers électroniques.

Enfin, le dernier chapitre présente la réalisation de notre logiciel permettant l'envoi des messages anonymes à travers une chaîne de remailers initialement choisis à partir d'une liste. Son fonctionnement général est expliqué à travers deux exemples avec leurs résultats.

Pour clore ce document, une conclusion générale est présentée, et comportera des éventuelles perspectives que l'on pourra prévoir pour la complémentation ou l'extension de ce travail.

Chapitre 1

La sécurité informatique

1. Introduction :

Vu la croissance des systèmes informatiques et l'augmentation considérable de réseaux interconnectés, la sécurité de données (sécurité informatique) est devenue aujourd'hui un problème crucial pour les utilisateurs et les propriétaires à la fois. Par conséquent, de nombreuses études ont été faites pour essayer de construire des systèmes informatiques de plus en plus sûrs en prenant en compte deux éléments principaux :

- Les entités qui sont des utilisateurs humains ou des processus agissant sur le système, elles sont parfois appelées sujets [1].
- Les objets qui sont des composants passifs qui consistent en des informations et des ressources systèmes [1].

La sécurité informatique a donc pour objectif de mettre en œuvre des mécanismes qui protègent ces deux éléments tout en assurant la confidentialité, l'intégrité et la disponibilité de service [1].

2. Les types de menaces sur les systèmes informatiques :

Avant de voir ce qu'est la sécurité informatique, ses différents aspects, services et mécanismes, il faudrait tout d'abord mettre le point sur les types de menaces et attaques contre lesquelles la sécurité informatique devra lutter.

Les menaces envers un système de communication de données de manière générale comprennent les éléments suivants :

- Destruction d'informations et/ou d'autres ressources.
- Corruption ou modification d'informations.
- Vol, suppression ou perte d'informations et/ou d'autres ressources.
- Divulgation d'informations.
- Interruption de services.

Les menaces peuvent être classées en menaces accidentelles ou menaces intentionnelles, elles peuvent être actives ou passives.

2.1. Les menaces accidentelles [2]:

Les menaces accidentelles sont celles qui existent sans qu'il y ait préméditation, par exemple: défaillance de système, bévues opérationnelles et bogues dans le logiciel.

2.2. Les menaces intentionnelles [2]:

Les menaces intentionnelles peuvent aller de l'examen fortuit, utilisant des outils de contrôles facilement disponibles, aux attaques sophistiquées, utilisant une connaissance spéciale du système. Une menace intentionnelle qui se concrétise peut être considérée comme une attaque.

2.3. Les menaces passives [2]:

Ce type de menaces ne produit aucune modification d'information contenue dans le système et avec lequel ni le fonctionnement, ni l'état du système ne changent. L'utilisation de branchements clandestins passifs pour observer des informations via une ligne de communication est une concrétisation d'une menace passive.

2.4. Les menaces actives [2]:

Les menaces actives comprennent l'altération d'informations contenues dans un système, ou des modifications de l'état ou du fonctionnement du système. Une modification malveillante des tables de routage d'un système par un utilisateur non autorisé est un exemple de menace active.

2.5. Notion d'attaques [2]:

Comme nous l'avons dit précédemment, une attaque est une menace intentionnelle d'une entité quelconque dans un but bien déterminé que ce soit pour altérer un système informatique ou pour aboutir à une information intéressante ou autre.

Nous pouvons avoir :

- Des attaques de l'intérieur : Qui se produisent lorsque des utilisateurs légitimes d'un système se comportent de façon non attendue ou non autorisée.

- Des attaques de l'extérieur : Dans lesquelles un intrus utilise des techniques telles que :
  - Le branchement clandestin (actif ou passif).
  - L'interception d'émission.
  - Le déguisement en utilisateur du système ou en composant de celui-ci.
  - Le court circuit des mécanismes d'authentification ou de contrôle d'accès.

2.6. Quelques types d'attaques spécifiques :

2.6.1. Le déguisement [2]:

Le déguisement est le procédé par lequel une entité se fait passer pour une autre afin de bénéficier de son privilège dans l'accès à des informations confidentielles ou autres. Le déguisement est généralement utilisé avec d'autres types d'attaques surtout lorsqu'il s'agit de l'authentification. Par exemple une entité, qui a une autorisation d'accès à un système informatique quelconque mais ayant peu de privilèges, peut utiliser un déguisement pour obtenir des privilèges supplémentaires en usurpant l'identité d'une entité qui a ces privilèges. Il suffit de capturer une séquence d'authentification correcte qui a eu lieu puis la rejouer.

2.6.2. Le rejeu [2]:

Le rejeu survient lorsqu'un message ou une partie d'un message sont interceptés puis rejoués (retransmis convenablement) pour produire un effet non autorisé. Ce type d'attaque est utilisé surtout pour réaliser un déguisement.

2.6.3. La modification des messages [2]:

La modification d'un message a lieu lorsque le contenu d'une transmission de données est modifié sans que cela soit détecté et produit un effet non autorisé. Par exemple lorsqu'un message « Autoriser 'Mohamed' à lire le fichier de 'Karim' » est modifié en « Autoriser 'Ahmed' à lire le fichier de 'Karim' ».

2.6.4. Les trappes [2]:

Lorsqu'une entité d'un système est modifiée pour permettre à un attaquant de produire un effet non autorisé sur une demande ou lors d'un événement prédéterminé, le résultat est

appelé trappe. Par exemple une validation de mot de passe peut être modifiée de façon à valider également le mot de passe d'un attaquant, en plus de son effet normal.

2.6.5. Le cheval de Troie [2]:

Un « Cheval de Troie » est un programme introduit dans le système avec une fonction non autorisée en plus de sa fonction autorisée. Un relais qui copie des messages à distance à partir d'une voie non autorisée est un Cheval de Troie.

2.6.6. Les virus [4]:

Ce sont des programmes qui, en s'exécutant, se reproduisent de nouveaux programmes qui deviennent alors des Chevaux de Troie.

2.6.7. Les vers [4]:

Les vers sont des programmes autonomes qui se propagent sur les réseaux dans le but de les saturer et provoquant ainsi un déni de services pour les utilisateurs autorisés, c'est à dire les empêcher de remplir leurs propres fonctions.

Notons qu'il existe beaucoup d'autres types qui n'ont pas été cités.

Il a fallu donc définir toute une politique de sécurité informatique se reposant sur un ensemble de règles et critères internationaux et normalisés pour protéger les systèmes de traitement de l'information ainsi que leur interconnexion contre ces différents types d'attaques et menaces, en mettant en œuvre des services et mécanismes de sécurité qui devront nous donner les solutions convenables pour faire face aux éventuelles menaces que ce soit de l'intérieur ou de l'extérieur.

3. La politique de sécurité informatique [2]:

La politique de sécurité informatique est constituée d'un ensemble de lois, règles et pratiques qui régissent le traitement des informations sensibles et l'utilisation des ressources par le matériel et le logiciel d'un système.

Cette forme de sécurité traite une grande variété de problèmes puisqu'elle est liée au contenu informatif du système :

- Comment les programmes doivent réagir au sein de l'ordinateur pour renforcer la politique de sécurité choisie ?
- Quelles sont les mécanismes matériels que peut nécessiter un système d'exploitation (mémoire virtuelle par exemple) ?
- Quel est le mécanisme de cryptage qui doit être choisi ?...etc.

5.3. La sécurité administrative :

Elle permet de mettre en œuvre les procédures administratives qui gèrent les ressources informatiques, en plus, elle assure la sécurité du personnel, cette dernière protège les objets contre toutes les attaques effectuées par les utilisateurs autorisés.

6. Les services de sécurité informatique [4]:

Comme nous l'avons dit en introduction, l'objectif de la sécurité informatique est de mettre en œuvre des services utilisant des mécanismes appropriés pour la protection des systèmes informatiques ainsi que leur interconnexion contre les attaques qui les menacent. Les services les plus importants sont :

6.1. L'authentification :

Avant même de vérifier qu'un sujet est autorisé à accéder à un objet, le système doit être sûr de l'identité du sujet. C'est à dire qu'il doit vérifier que le sujet est bien celui qu'il dit être.

Deux types d'authentification sont généralement présents dans un système informatique : l'authentification de l'utilisateur lors de sa connexion au système et l'authentification de l'entité homologue lors de l'ouverture d'une communication entre deux entités du système.

6.1.1. L'authentification de l'utilisateur :

Lors de la connexion de l'utilisateur au système informatique, il s'agit, pour le système, de vérifier l'identité de celui-ci ainsi que son droit d'accès à l'ensemble du système et d'associer

un identificateur à cet utilisateur afin de pouvoir, par la suite, identifier l'ensemble de ses actions effectuées.

6.1.2. L'authentification de l'entité homologue :

Lorsque deux sujets désirent communiquer : ils veulent, au préalable, vérifier l'identité de l'entité homologue afin d'être sûrs de dialoguer avec le sujet désiré (les sujets veulent vérifier que l'entité homologue est bien celle qu'elle dit être).

6.2. La confidentialité :

Les services de confidentialité assurent la protection des données contre toute divulgation non autorisée. On distingue plusieurs formes :

6.2.1. La confidentialité des données en mode connexion :

La confidentialité des données échangées lors d'une connexion entre deux utilisateurs est assurée par ce service.

6.2.2 La confidentialité des données en mode sans connexion :

Ce service garantit la confidentialité de données d'un sujet utilisant un service en mode sans connexion tel que le courrier électronique.

6.2.3. La confidentialité sélective par champ :

Seulement la confidentialité des champs sélectionnés dans les données de l'utilisateur est garantie, que ce soit en mode connexion ou sans connexion.

6.2.4. La confidentialité du flux de données :

Il s'agit ici de la protection des informations qui pourraient être dérivées de l'observation du flux de données.

6.3. L'intégrité des données :

Ce service assure la protection contre les menaces actives qui peuvent modifier les informations sensibles. Il peut prendre les formes suivantes :

6.3.1. L'intégrité en mode connexion avec reprise :

Cette forme de service sert à détecter toutes les données modifiées, insérées, supprimées ou rejouées lors d'une connexion entre utilisateurs (Prend en charge les tentatives de reprise).

6.3.2. L'intégrité en mode connexion sans reprise :

C'est le même service que le précédent, sauf que celui-ci ne considère pas les tentatives de reprise.

6.3.3. L'intégrité en mode connexion sélective par champ :

Ce service assure l'intégrité des champs sélectionnés dans les données de l'utilisateur au cours d'une connexion et prend la forme d'une indication permettant de savoir si les champs sélectionnés ont été modifiés, supprimés ou rejoués.

6.3.4. L'intégrité en mode sans connexion :

Ce service assure l'intégrité de données en mode sans connexion et peut prendre la forme d'une indication permettant de savoir si une unité de donnée reçue est modifiée.

6.3.5. L'intégrité en mode sans connexion sélective par champ :

Ce service assure l'intégrité d'une unité de données en mode sans connexion et peut prendre la forme d'une indication en cas d'une éventuelle modification de cette unité de donnée.

6.4. La non-répudiation :

Ce service peut prendre l'une des deux formes suivantes :

6.4.1. Non-répudiation avec preuve de l'origine :

Le destinataire des données reçoit la preuve de leur origine, par conséquent, il sera protégé de toute tentative de l'expéditeur de nier le fait qu'il a envoyé ces données ou leur contenu.

6.4.2. Non-répudiation avec preuve de la remise :

Ici, c'est l'expéditeur qui est protégé contre toute tentative ultérieure du destinataire de nier le fait d'avoir reçu les données ou leur contenu.

6.5. Le contrôle d'accès :

Le service de contrôle d'accès assure la protection d'un objet donné du système informatique de toute utilisation non autorisée. Il doit vérifier donc qu'un sujet a le droit d'accéder à des donnés de ce système (contenues dans un fichier par exemple), comme il doit vérifier qu'un sujet a le droit de communiquer avec un autre sujet (alors considéré dans ce cas comme un objet à protéger). Et conformément aux objectifs de la sécurité informatique, le contrôle d'accès doit vérifier que les accès aux ressources d'un système informatique conservent la confidentialité et l'intégrité des données.

7. Conclusion :

Ce premier chapitre, nous a permis d'avoir un bon aperçu sur les concepts de bases de la sécurité informatique. En passant par les aspects, la politique, les formes, les services de sécurité informatique, nous avons pu constater l'importance primordiale de ce domaine pour que les différentes entreprises industrielles et commerciales, les laboratoires de recherches scientifiques et militaires, et les gouvernements puissent concevoir leurs propres systèmes informatiques interconnectés un peu partout dans le monde entier afin d'effectuer les échanges d'informations sensibles d'une manière très rapide et surtout plus sûre.

La confidentialité et l'intégrité des données transitant sur les réseaux étant les deux aspects les plus importants, différents mécanismes ont été mis en œuvre pour préserver ces deux aspects en l'occurrence la cryptographie avec ces différents algorithmes et techniques.

L'étude détaillée de ces mécanismes fera l'objet du prochain chapitre.

Chapitre 2

Les mécanismes de sécurité informatique

1. Introduction :

Un mécanisme de sécurité est un ensemble d'algorithmes implémentés par l'un des services de sécurité précédemment décrits pour garantir l'exécution de toutes les règles et les lois définies dans la politique de sécurité. Ces mécanismes peuvent être implémentés séparément, ou en combinant deux ou plusieurs, afin d'assurer une meilleure robustesse contre les attaques auxquelles ils sont destinés. Voici donc les mécanismes de base actuellement utilisés.

2. La cryptographie [5]:

La cryptographie permet de dissimuler les informations qui circulent sur le réseau en les rendant incompréhensibles à toutes les personnes à qui elles ne sont pas destinées, surtout à celles qui ne doivent impérativement pas en prendre connaissance. Le cryptage se fait à l'aide d'une clé, comparable à un mot de passe. A l'autre bout de la chaîne, la connaissance de cette clé par le destinataire est indispensable pour le déchiffrement du message crypté.

Les algorithmes de base qui ont été utilisés en cryptographie au fil de l'histoire, se répartissent en trois grandes familles :

- Les algorithmes de substitution, qui consistent à remplacer les lettres du texte clair par d'autres caractères.
- Les codes, qui consistent à convertir les lettres du texte clair en nombres.
- Les algorithmes de transposition qui consistent à permuter l'ordre des lettres du texte clair de façon à le rendre incompréhensible.

Si ces algorithmes n'offrent aucune sécurité lorsqu'ils sont utilisés séparément, leur combinaison a cependant pu donner naissance à des algorithmes particulièrement fiables, actuellement utilisés. Ces algorithmes se répartissent entre ceux à clé privée (Algorithmes symétriques) et ceux à clé publique (Algorithmes asymétriques). Les premiers (DES, IDEA...) reposent sur le concept d'une clé secrète servant tant au chiffrement qu'au déchiffrement des messages. La clé utilisée pour crypter les messages doit ainsi être impérativement transmise aux destinataires dans le plus grand secret, ce qui n'est pas une chose aisée puisque, que ce soit par courrier ou par téléphone, elle peut être interceptée. Pour surmonter cette difficulté, deux chercheurs américains, Diffie et Hellman, ont mis au point un

nouveau système de chiffrement : les algorithmes à clé publique. Ces algorithmes, appelés aussi asymétriques, utilisent en fait une paire de clés : la première, dite clé publique, sert à crypter les messages, alors que la seconde, dite clé privée, connue du seul destinataire de ces messages, sert à les décrypter. Le plus célèbre algorithme à clé publique est le RSA.

Ce procédé de chiffrement a en outre permis de mettre au point un procédé de signature numérique pour pouvoir identifier avec certitude l'auteur d'un message ou d'une transaction électronique, particulièrement utile pour sécuriser les opérations de commerce électronique. Sur le réseau mondial, ces algorithmes sont employés par divers protocoles cryptographiques de protection des transactions en ligne ainsi que par un logiciel cryptographique particulièrement performant, le PGP *(Pretty Good Privacy)*, offert gratuitement sur Internet et qui sert à la protection du courrier électronique et à l'authentification.

Comme la cryptographie, la stéganographie permet de dissimuler la teneur des informations, mais en recourant cependant à une autre approche. Alors que la cryptographie dissimule les messages en les rendant incompréhensibles, la stéganographie vise à rendre l'existence même de ces messages secrète. Le watermarking constitue une de ses principales applications. Il consiste à camoufler dans un médium une marque imperceptible servant de signature numérique et contenant différentes informations telles que les permissions attachées au document, l'identité du propriétaire ou celle de ceux qui ont le droit d'y avoir accès ou de l'utiliser. Il a été mis au point pour lutter contre la piraterie informatique qui a connu un boom sans précédent suite au remplacement des supports analogiques par des supports numériques, rendant par là même la copie conforme des informations numérisées circulant sur la toile, qu'elles soient son, image ou vidéo, extrêmement aisée et rapide.

2.1. Les algorithmes à clé privée :

Les algorithmes à clé secrète sont aussi appelés algorithmes **symétriques**.
Ceci est dû au fait que la même clé est utilisée pour le chiffrement et le déchiffrement d'un document. Il est donc nécessaire que les deux interlocuteurs se soient mis d'accord sur une clé privée auparavant, par courrier, par téléphone ou lors d'un entretien privé.

Cette clé doit être gardée secrète. La sécurité d'un algorithme à clé secrète repose sur la clé ; si celle-ci est dévoilée, alors n'importe qui peut chiffrer ou déchiffrer des messages dans ce cryptosystème.

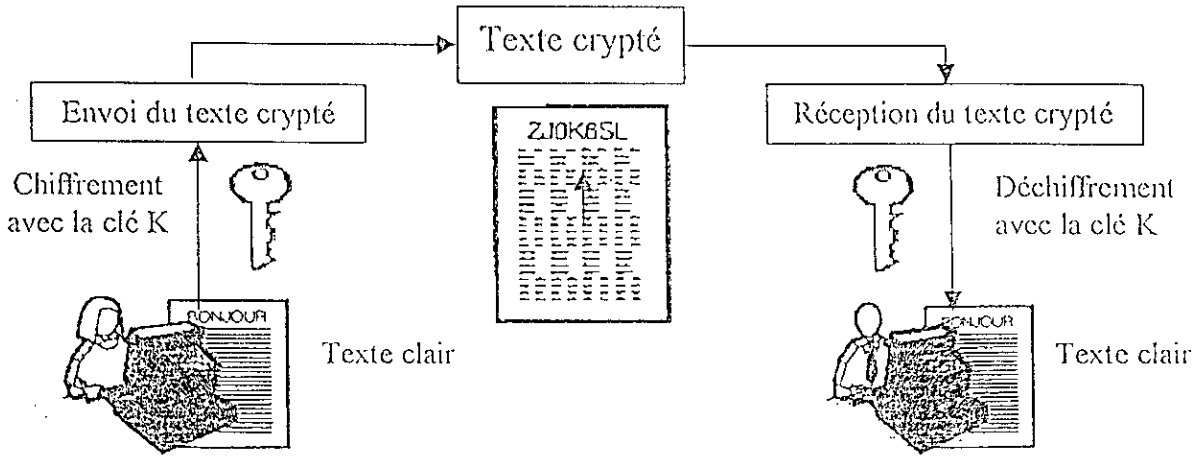


Figure 2.1 : Cryptosystème à clé privée

Les algorithmes à clé secrète sont classés en deux catégories.

Certains opèrent sur le message en clair un bit à la fois. Ceux-ci sont appelés « **algorithmes de chiffrement en continu** ».

D'autres opèrent sur le message en clair par groupes de bits. Ces groupes de bits sont appelés blocs, et les algorithmes correspondants sont appelés « **algorithmes de chiffrement par blocs** ».

Pour des algorithmes réalisés sur ordinateur, la taille typique des blocs est de 64 bits.

Dans notre étude, nous allons traiter le **DES** et l'**IDEA**.

❖ DES « Data Encryption Standard » :

Le DES est un système de chiffrement par blocs; il chiffre les données du texte en clair par blocs de 64 bits en utilisant une clé d'une certaine longueur.

L'algorithme est une combinaison de deux algorithmes élémentaires (substitution et permutation) appliquée au texte, basée sur la clé.

Le DES a 16 rondes (cycles), c'est à dire qu'il applique 16 fois la même combinaison de techniques au bloc de texte en clair.

Après une permutation initiale, le bloc est coupé en une partie droite et une partie gauche, chacune d'une longueur de 32 bits. Après cela, il y a 16 rondes d'opérations identiques, appelées « fonctions *f* », lors desquelles les données sont combinées avec la clé. Après la seizième ronde, les parties gauche et droite sont rassemblées et une permutation finale (l'inverse de la permutation initiale) termine l'algorithme.

A chaque ronde, les bits de la clé sont décalés et 48 bits sont alors sélectionnés parmi les 56 bits de la clé. La partie droite des données est étendue à 48 bits par une permutation expansive, combinée avec 48 bits de la clé décalée et permutée par *ou exclusif,* remplacée par 32 nouveaux bits par un algorithme de substitution et permutée une fois de plus.

La « fonction *f* » est constituée de ces quatre opérations. La sortie de la « fonction *f* » est alors combinée avec la moitié gauche par *ou exclusif*.

Le résultat de ces opérations devient la nouvelle moitié droite. Ces opérations sont répétées 16 fois, donnant le DES à 16 rondes.

Les deux organigrammes des figures 2.2 et 2.3 donnent une idée générale sur le fonctionnement du DES, et du principe de génération des clés.

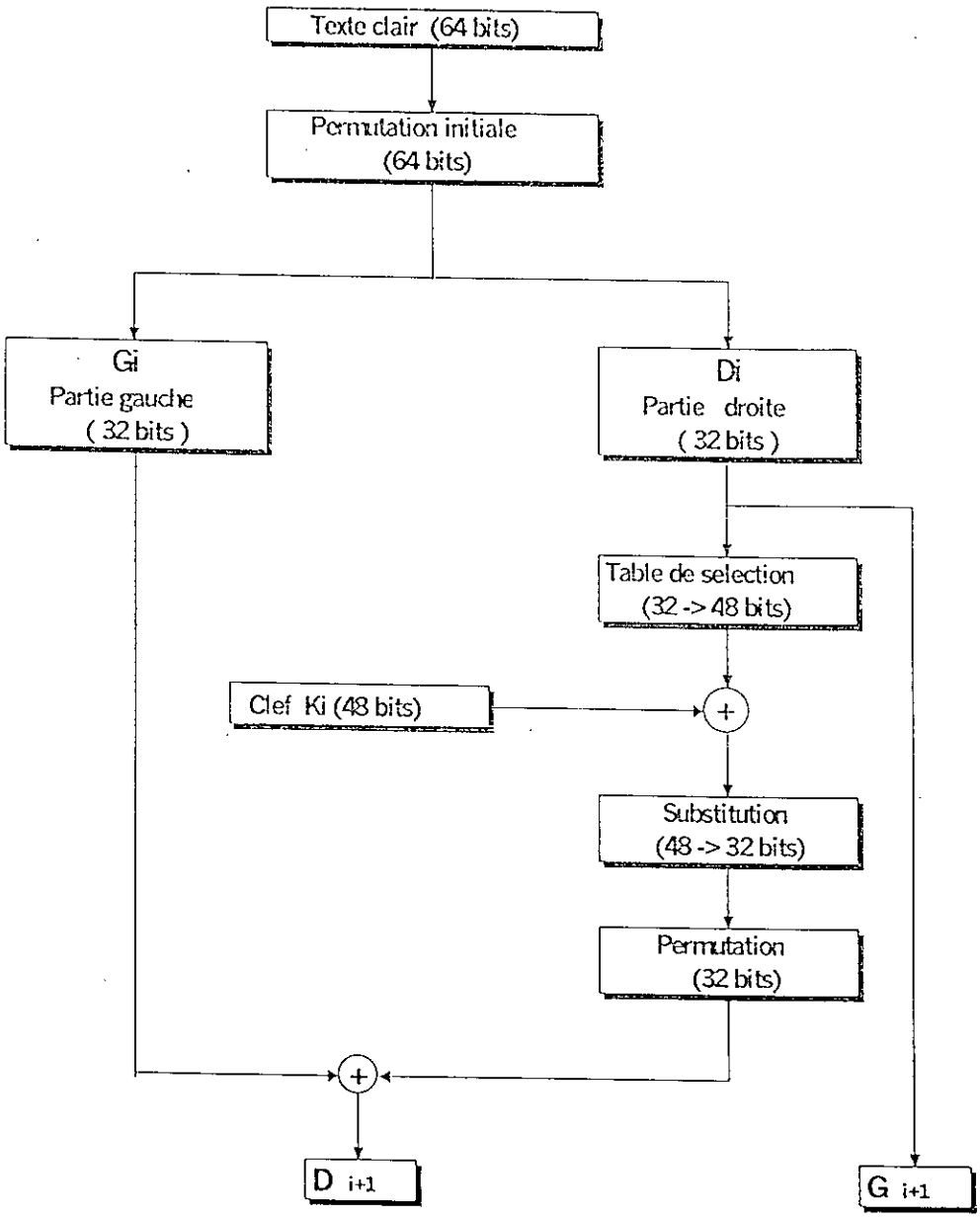


Figure 2.2 : Organigramme du DES

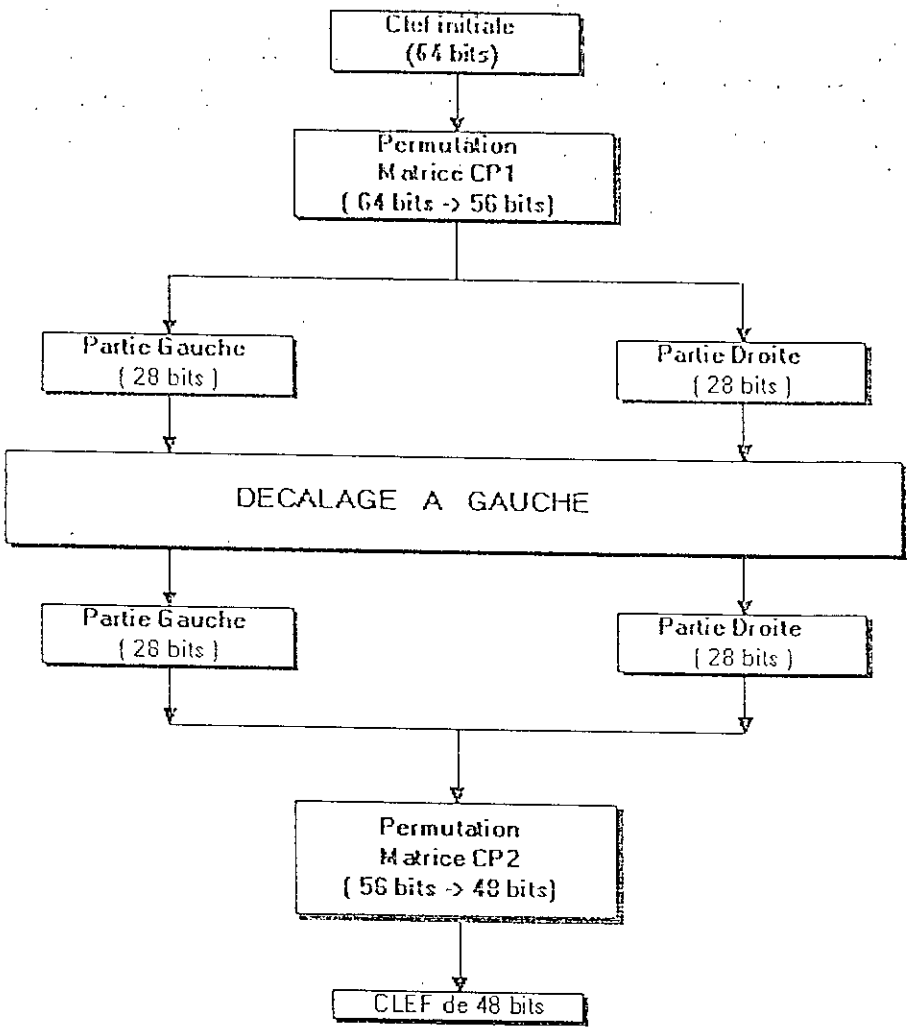


Figure 2.3 : Organigramme du plan de génération des clés

Pour le déchiffrement du DES, les opérations effectuées sur le texte à crypter (substitution, permutation, ou exclusif et décalage) ont été choisies pour offrir une propriété très utile : le même algorithme est utilisé pour le chiffrement et le déchiffrement.

Avec le DES il est possible d'utiliser la même fonction pour chiffrer un bloc et le déchiffrer. La seule différence est que les clés doivent être utilisées dans l'ordre inverse. Si les clés de chiffrement de chaque ronde sont K_1, K_2 ... K_{16}, alors les clés de déchiffrement sont respectivement K_{16}, K_{15} ... K_1.

L'algorithme qui engendre les clés pour chaque ronde est également circulaire, et le nombre de décalages à effectuer se lit à partir de la fin de la table au lieu du début.

Certaines réalisations utilisent le DES *triple* (voir figure 2.4), la cryptanalyse sera encore plus difficile que dans le cas du DES *simple*.

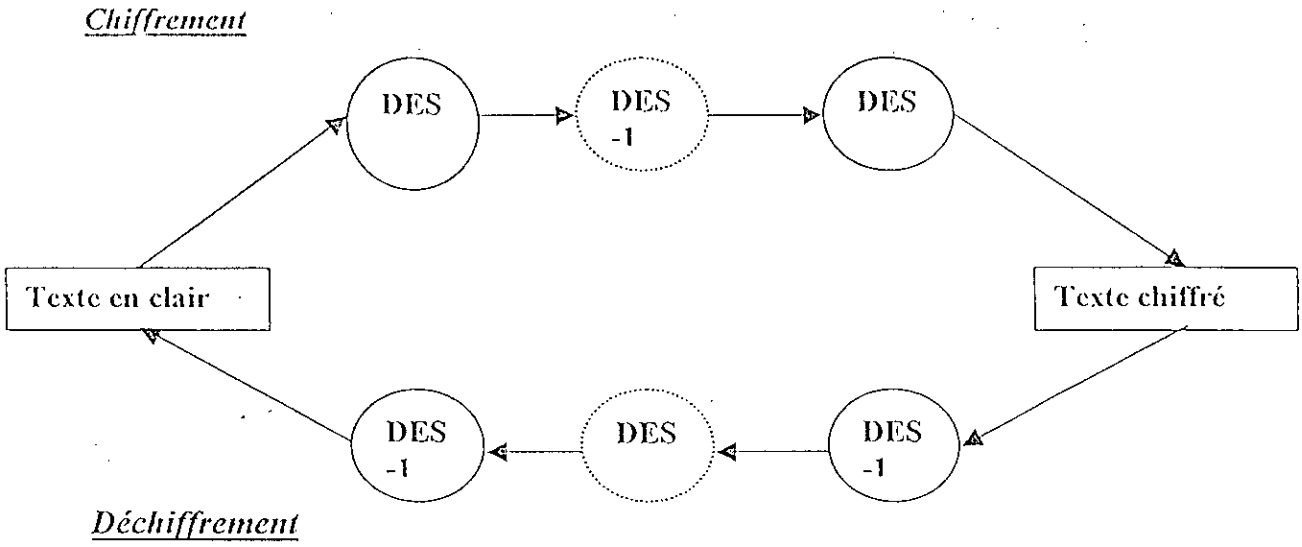


Figure 2.4 : DES triple

N.B: Dans le DES triple, on utilise pour le cryptage et le décryptage le même format de données et la même clé.

Pour l'attaque d'un document crypté avec DES, la seule méthode connue à ce jour pour décrypter un message crypté avec DES, est la méthode dite "brute" qui consiste à tester la totalité des différentes clés de 56 bits possibles. Le problème majeur est qu'il y en a 2^{56} , soit exactement :

72 057 595 037 927 936 différentes ! Cela peut prendre un temps considérable. Cependant, les services secrets peuvent avoir les moyens matériels de briser de tels codes, il leur suffit d'avoir une ou des machines extrêmement puissantes, ce qui pourrait tout à fait être possible pour des nations importantes...

❖ IDEA « International Data Encryption Algorithm »:

IDEA est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 rondes.

La philosophie de la conception de cet algorithme, utilisé par PGP, est basée sur le mélange d'opérations de différents groupes algébriques (voir figure 2.5).

Il n'utilise que trois opérations algébriques simples qui sont :

- Ou exclusif ;
- Addition modulo 2^{16} ;
- Multiplication modulo $2^{16}+1$.

Toutes ces opérations (qui sont les seules opérations de l'algorithme car il n'y a pas de permutations) manipulent des blocs de 16 bits.

Le texte est découpé en blocs de 64 bits, redivisés en quatre blocs de 16 bits : X_1, X_2, X_3, X_4. Ces blocs deviennent les entrées de la première, des huit rondes de l'algorithme.

A chaque ronde, ces blocs sont multipliés, additionnés, et combinés par des ou exclusif, avec 6 sous-blocs K_1, K_2,...K_6, dérivés de la clé.

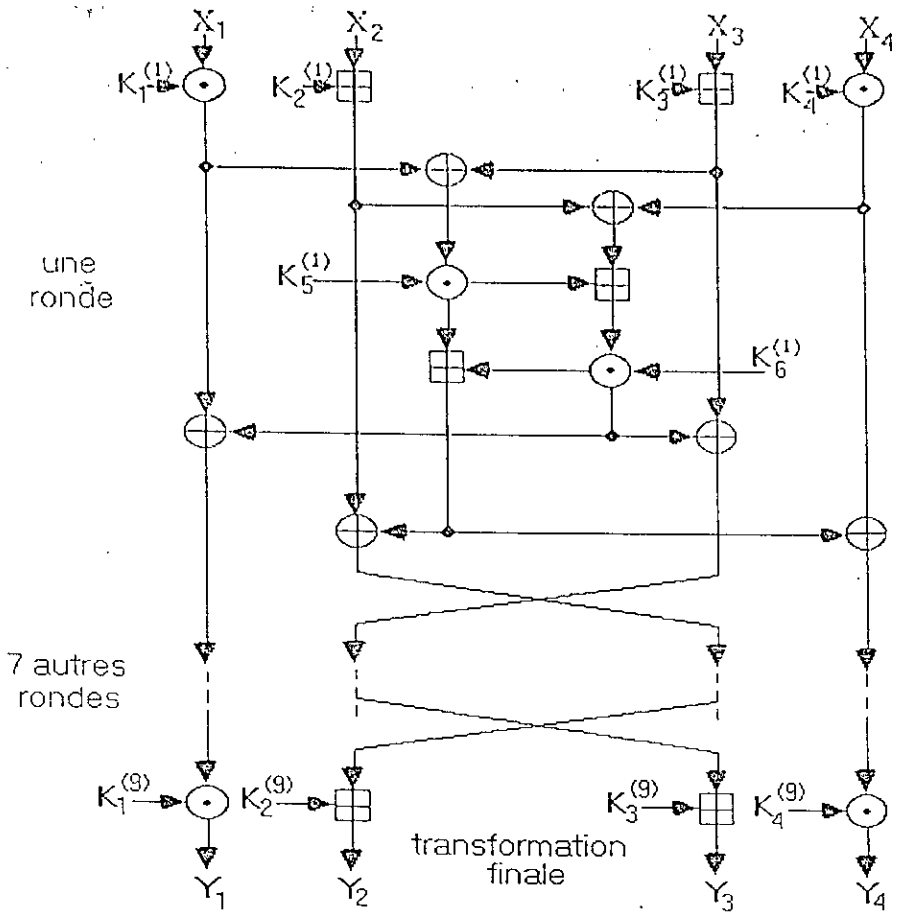
A la fin de chaque ronde le deuxième et le troisième sous-bloc sont échangés.

En ce qui concerne les sous-clés, l'algorithme en utilise 52 (6 pour chacune des 8 rondes, et 4 pour la transformation finale).

La clé initiale de 128 bits est divisée en 8 sous-clés de 16 bits. Ce sont les 8 premières clés de l'algorithme (6 de la première ronde, et 2 de la deuxième).

Ensuite, la clé est décalée circulairement de 25 bits vers la gauche, puis est redivisée en 8 sous clés (4 de la deuxième ronde, et 4 de la troisième).

La clé est de nouveau décalée circulairement de 25 bits vers la gauche, puis redivisée et ainsi de suite jusqu'à la fin de l'algorithme.



X_i : Sous-blocs de 16 bits de texte clair
Y_i : Sous-blocs de 16 bits de texte chiffré
$K_i^{(r)}$: Sous-blocs de clé de 16 bits

⊕ : Ou exclusif bit à bit de sous blocs de 16 bits

⊞ : Addition modulo 2^{16} d'entiers de 16 bits

⊙ : Multiplication modulo $2^{16}+1$ d'entiers de 16 bits avec le sous-bloc nul correspondant à 2^{16}

Figure 2.5 : Squelette d'IDEA

En ce qui concerne la vitesse d'IDEA, les versions logicielles actuelles d'IDEA sont presque aussi rapides que le DES.

Sur un 80386 avec une fréquence d'horloge de 33 MHz, IDEA chiffre à une vitesse de 880 Kbits/s, et 2400 Kbits/s sur un 80486 cadencé à 66 MHz.

Du côté matériel, une puce VLSI contenant 251000 transistors sur une puce de 107,8 mm^2, chiffre les données à l'aide d'IDEA à la vitesse de 177 Mbits/s avec une fréquence d'horloge de 25 MHz.

La longueur de clé d'IDEA est de 128 bits – plus du double de celle du DES.
En faisant l'hypothèse que l'attaque exhaustive est la plus efficace, il faudrait 2^{128} chiffrements pour retrouver la clé.
Si on conçoit une puce capable de tester 1 milliard de clés par seconde, puis en mettra 1 milliard à la tache, ceci prendra 10^{13} années (plus que l'âge de l'univers !).
Une matrice de 10^{24} puces pourrait trouver la clé en une journée, mais tous les atomes de silicium de l'univers ne seraient pas suffisants pour concevoir une telle machine.

2.2. Les algorithmes à clé publique :

Deux chercheurs américains, Whitfield **DIFFIE** et Martin **HELLMAN** proposent une autre approche : au lieu d'utiliser une clé connue de l'expéditeur et du destinataire, il serait plus intéressant d'utiliser pour le chiffrement une clé nommée **clé publique**, disponible sur Internet par exemple, et une autre clé appelée **clé privée**, connue du seul destinataire, qui lui permet de déchiffrer ses messages. De ce fait, les algorithmes à clé publique sont aussi appelés algorithmes **asymétriques**. C'est le principe de la boîte aux lettres : tout le monde peut y glisser quelque chose, mais seul celui qui possède la clé peut récupérer le courrier.

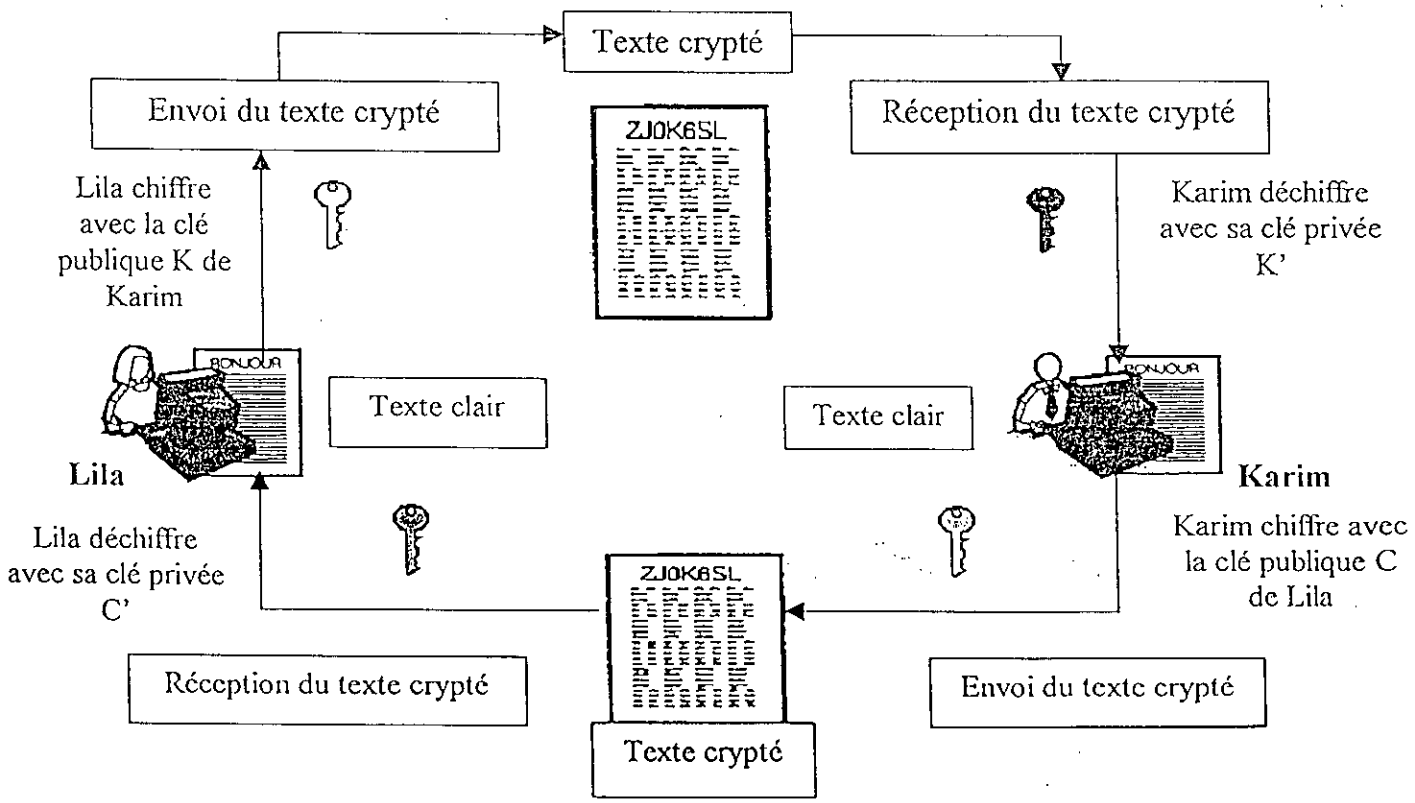


Figure 2.6 : Cryptosystème à clé publique

❖ **RSA « Rivest Shamir Adelman »:**

Le RSA est né au Weizmann Institute en Israël. Du nom de ses inventeurs (Ron RIVEST, Adi SHAMIR et Leonard ADELMAN), il permet de distribuer librement la clé publique, mais sans que personne ne puisse décrypter les messages ainsi cryptés.

Le brevet de cet algorithme appartient à la société américaine RSA Data Security, qui fait maintenant partie de Security Dynamics et au Public Key Parteners, (PKP à Sunnyvale, Californie, Etats-Unis) qui possèdent les droits en général sur les algorithmes à clé publique.

Tout le principe de RSA repose sur le fait qu'il est très difficile et très long de factoriser un très grand nombre en deux facteurs premiers.

Pour crypter avec RSA, on commence par calculer les différentes clés nécessaires au cryptage et au décryptage.

Ces derniers se feront à l'aide de simples formules mathématiques.

La base de l'algorithme RSA, c'est le nombre **n**. Ce nombre doit être le produit de deux nombres premiers **p** et **q**, très grands et ayant sensiblement le même nombre de chiffres. Ceci pour la simple raison que la puissance du cryptage RSA est basée sur la difficulté de factoriser un grand entier.

En bref, si un jour quelqu'un arrive à factoriser le **n** qui a servi à constituer nos clés, c'est à dire à découvrir les nombres **p** et **q**, alors il n'aura aucun mal à reconstituer notre clé privée. C'est pour cela que l'on choisit des nombres premiers **p** et **q** d'environ 100 à 200 chiffres et même plus, pour rendre la factorisation hors de portée, même des meilleurs ordinateurs.

Ensuite il nous faut trouver un entier e compris entre 2 et $\varphi(n)$ tel que e soit premier avec $\varphi(n)$.

$\varphi(n)$ est la fonction indicatrice d'Euler, c'est en fait le nombre d'entiers inférieurs à n qui sont premiers avec lui, on a donc :

$$\varphi(n) = (p-1).(q-1).$$

$\varphi(n)$ se calcule très facilement ici, puisque l'on a p et q.

Maintenant que l'on a n et e, nous sommes prêts à crypter.

Les nombres n et e forment ici notre clé publique que l'on notera [n,e]. Il nous faut calculer le nombre d qui sera nécessaire au décryptage. Selon la théorie de RSA, nous devons avoir d tel que (e.d-1) soit divisible par $\varphi(n)$.

Pour trouver d nous devons alors résoudre l'équation diophantienne $e.d+k.\varphi(n)=1$ à l'aide de l'arithmétique.

Comme e et $\varphi(n)$ sont premiers entre eux, le théorème de Bezout prouve qu'il existe d et k dans Z tel que : $e.d + k.\varphi(n) = 1$.

On pourra résoudre l'équation grâce à l'algorithme d'Euclide.

Après résolution, on arrivera à une classe de solutions de la forme $d = r.\varphi(n) + d_0$ (où r appartient à Z) puisque e a été choisi premier avec $\varphi(n)$.

L'ensemble des solutions d à l'équation $e.d+k.\varphi(n)=1$ est une classe de congruence modulo $\varphi(n)$, il y a donc une unique solution d comprise entre 2 et $\varphi(n)$, donc $d=d_0$.

En résumé, on doit chercher le nombre d tel que : $\mathbf{e.d = 1 \bmod [\varphi(n)]}$, d'où :

$$\mathbf{d = e^{-1} \bmod [\varphi(n)]}$$

Nous voilà prêts à décrypter. Le nombre d est notre clé privée.

Nous pouvons à présent rendre publique notre clé publique [n,e] et garder secrète notre clé privée. Quant aux nombres p, q, et $\varphi(n)$, on doit, soit les conserver secrets, soit les détruire car ils ne serviront plus.

Pour crypter un document que l'on aura auparavant transformé en nombres m_i inférieurs à n (codes ASCII par exemple), il nous faut effectuer l'opération :

$$\mathbf{c_i = m_i^e \bmod n}$$

c est ici notre nombre m une fois crypté. La première opération peut être très longue à effectuer à la main, l'utilisation d'un ordinateur et d'un programme spécial est fortement conseillée.

Pour décrypter un document c, il nous faut effectuer l'opération suivante :

$$\mathbf{m_i = c_i^d \bmod n}$$

m sera bel et bien notre nombre décrypté, qu'il ne restera plus qu'à retransformer en texte ou en autre chose. La preuve de cette algorithme de chiffrement est faite avec le théorème de Fermat et le théorème chinois des restes connus depuis quelques siècles !

Puisque toutes les opérations ont été effectuées modulo n :

$$c_i^d = (m_i^e)^d = m_i^{e.d} = m_i^{k.(p-1).(q-1)+1} = m_i \,.\, m_i^{k.(p-1).(q-1)} = m_i \times 1 = m_i$$

N.B : D'après le théorème de Fermat généralisé par Euler : $a^{\varphi(n)} \bmod(n) = 1$.

Pour ses performances, à sa vitesse maximum, le RSA est environ 100 fois moins rapide que le DES.

La réalisation matérielle la plus rapide de RSA avec un module de 512 bits a un débit de 64 Kbits/s. Il existe aussi des puces qui effectuent des chiffrements RSA avec des modules à 1024 bits.

Mais des puces à un débit qui approcherait le million de bits par secondes sont prévues.

En logiciel, le DES est environ 100 fois plus rapide que RSA.

Ce nombre pourrait changer avec les progrès de la technologie, mais n'atteindra jamais la vitesse des algorithmes à clé secrète.

En ce qui concerne la cryptanalyse du RSA, comme on l'a vu précédemment, la résistance d'un document crypté avec l'algorithme RSA s'appuie sur le fait qu'il est extrêmement difficile de factoriser en deux facteurs premiers un très grand nombre. L'attaque va donc consister à utiliser des algorithmes de factorisation les plus rapides, et les plus puissants possibles, pour factoriser le nombre n extrêmement grand de la clé publique visée. L'attaque peut aussi consister à essayer de deviner les nombres p-1 et q-1, ou d'essayer toutes les valeurs de d jusqu'à tomber sur la bonne mais ces techniques sont encore moins efficaces que l'attaque par factorisation de n. L'attaque d'un tel document est encore beaucoup plus longue (pour une taille du nombre n raisonnable) que l'attaque d'un document crypté avec DES.

C'est pourquoi, de grandes recherches en mathématiques sur des algorithmes de factorisation de plus en plus rapides sont effectuées partout dans le monde.

La méthode RSA, est réputée pour sa quasi-invulnérabilité, quand elle est utilisée avec une très grande clé (plus de 200 chiffres).

A titre d'exemple, une équipe de 600 utilisateurs a mis six mois à casser un nombre de 129 chiffres!

Mais cette méthode pourrait s'écrouler si quelqu'un parvenait un jour à écrire un tel algorithme. Car RSA repose sur un principe qui a l'air évident mais qui n'a jamais été prouvé ! Actuellement, il n'y a aucun algorithme/méthode connu, capable de factoriser dans un temps convenable une très grande clé.

2.3. PGP « Pretty Good Privacy »:

C'est un outil de protection téléchargeable gratuitement de l'Internet, conçu à l'origine par Philip ZIMMERMANN.

C'est de loin le logiciel de cryptographie le plus utilisé dans l'Internet par les particuliers. Et ce, parce qu'il est à la fois rapide, pratique et surtout gratuit.

PGP utilise très souvent ***IDEA*** (comme algorithme à clé secrète au lieu du DES) pour le chiffrement, ***RSA*** (avec des clés allant jusqu'à 2047 bits) pour la gestion des clés et les signatures numériques, et ***MD5*** comme fonction de hachage à sens unique.

C'est la **combinaison algorithme à clé publique / algorithme à clé privée** qui donne à PGP sa **vitesse** et sa **grande sécurité.**

D'après Willian Crowell, Directeur délégué de National Security Agency, 20 Mars 1997 :

> ***« Si tous les ordinateurs du monde -260 millions- étaient mis à travailler sur un seul message crypté avec PGP, cela prendrait encore un temps estimé à 12 millions de fois l'âge de l'univers, en moyenne, pour un simple message. »***

L'opération de cryptage se fait donc en deux étapes principales :

- PGP crée une clé secrète IDEA de manière aléatoire, et crypte les données avec cette clé.
- PGP crypte la clé secrète IDEA précédemment créée au moyen de la clé publique RSA du destinataire.

De même, l'opération de décryptage se fait elle aussi en deux étapes :

- PGP décrypte la clé secrète IDEA au moyen de la clé privée RSA.
- PGP décrypte les données avec la clé secrète IDEA précédemment obtenue.

2.4. La stéganographie :

Parallèlement à la cryptographie, on trouve la stéganographie. C'est une technique qui a le même but, cacher des informations, mais a une autre approche de la question.

La stéganographie permet de faire passer un message de manière sûre, non pas en le chiffrant, mais en le dissimulant dans un autre message, de sorte que l'existence même du secret soit dissimulée.

Alors qu'avec la cryptographie habituelle, la sécurité repose sur l'assurance que le message ne sera pas compris, avec la stéganographie, la sécurité vient du fait que l'on pense que le message ne sera pas détecté.

Certaines sortes de stéganographie ne tromperont pas une personne, mais peuvent facilement tromper de gros ordinateurs passant au crible l'Internet à la recherche de messages intéressants.

De nos jours, la stéganographie, comme la cryptographie habituelle, a évolué en rencontrant l'informatique. Les messages, transformés en longues suites de bits, sont camouflés parmi les bits d'un autre fichier, image, son, vidéo...
La méthode la plus utilisée consiste à camoufler chaque bit du message dans le dernier bit de chaque octet du fichier qui sert de camouflage.

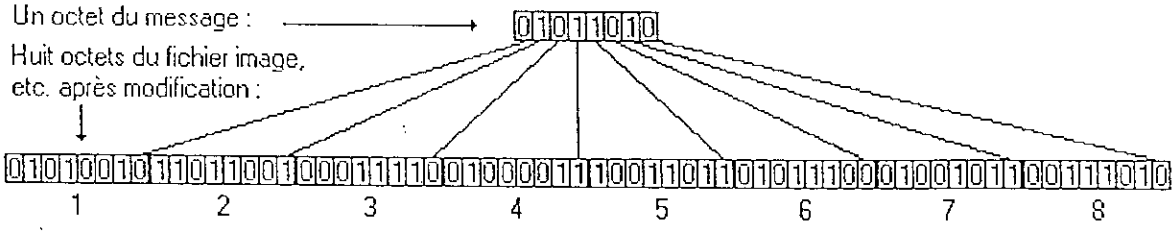


Figure 2.7 : Utilisation de la stéganographie avec les images bitmap

Prenons le cas d'une image au format bitmap. Soit une image 24 bits (voir la figure 2.7), c'est-à-dire environ 16 millions de couleurs. 24 bits sont utilisés pour chaque pixel : soit 3 octets, un donne la valeur du rouge (0 à 255), un pour le vert et un pour le bleu (c'est la décomposition RGB qui utilise ces trois couleurs primaires pour reconstituer n'importe quelle autre couleur).
En ne modifiant que le dernier bit de chaque octet, la couleur finale ne variera pas beaucoup (quasiment invisible à l'œil nu).
L'image, qui semblera normale pour quiconque l'interceptera, contiendra en outre un message caché pour une personne avertie.

L'une des applications intéressantes de la stéganographie est : Le **Watermarking** :
Il consiste à camoufler dans un **médium** (qui peut être une image, une chanson, un film vidéo), une marque (en anglais, watermark signifie filigrane) qui doit être suffisamment imperceptible pour ne pas détériorer le médium et suffisamment robuste pour pouvoir être

décelée même après traitement du médium que ce soit un traitement usuel ou celui résultant d'une attaque du système de marquage.

On peut considérer que ces marques servent de signature numérique du document.

En modifiant quelques bits de l'image, on peut ainsi signer et protéger cette image.

Le contenu d'une marque, typiquement quelques bits d'information, peut être de différentes natures.

1. Elle contient des informations sur les permissions attachées au document.
2. La marque peut indiquer qui est propriétaire du document.
3. On peut également marquer l'ayant droit du document, c'est-à-dire la personne à qui le propriétaire a donné une copie.

3. Mécanismes d'authentification :

Dés l'apparition des systèmes Multi-Utilisateurs, le problème de l'authentification des utilisateurs s'est posé : l'accès au système doit être autorisé uniquement aux sujets habilités.

La phase d'authentification est décomposée en deux parties : l'identification lors de laquelle l'utilisateur présente son identifiant qui doit être unique et connu par tout le monde, et l'authentification où l'utilisateur prouve son identité. Pour cela, diverses techniques sont employées, telles que l'envoi d'un mot de passe quand l'utilisateur veut accéder au système ou l'utilisation des cryptosystèmes asymétriques (signature numérique) où seule l'entité voulant s'authentifier connaît la méthode de cryptage du message alors que toutes les autres entités connaissent la méthode de décryptage.

Dans ce cas, cette entité ne révèle pas la clé de cryptage, c'est un secret qui n'est pas partagé avec d'autres entités contrairement au mot de passe qui doit être connu aussi par le vérifieur.

3.1. Le mot de passe (password) [4]:

Les mots de passe ou PIN (personal identification number) est un mécanisme sur lequel beaucoup de systèmes d'authentification sont basés. Cependant, ces systèmes sont vulnérables. Tout d'abord, ces mots de passe peuvent être révélés ou devinés. Plusieurs mesures et recommandations sont établies pour le choix des mots de passe. En plus de ces vulnérabilités, l'écoute de la ligne peut révéler les mots de passe. Pour éviter ceci, des mécanismes de protection de mots de passe sont mis en œuvre, sans avoir recours à la

cryptographie, la protection peut être assurée en utilisant des fonctions de hachage à sens unique (définies ultérieurement).

Supposons que le mot de passe correct de l'identité *id* est p. Au niveau du terminal où un processus d'authentification prend place, l'identité *id* et un mot de passe p' sont entrés. La valeur q'=f(p') est calculée et envoyée au vérificur avec l'identité *id*.

Le vérificur tient pour chaque *id* la valeur q générée pour le mot de passe correct p. Si q=q' alors le bon mot de passe est entré et l'identité est authentifiée.

3.2. La signature numérique [5]:

C'est un mécanisme d'authentification qui se base sur l'application d'une fonction mathématique qu'on appelle fonction de hachage sur le message (ou sur une portion seulement).

Pour ne pas être falsifiable, on va crypter cette signature par exemple avec l'algorithme RSA.

➢ **Définitions préliminaires :**

1) Fonction de hachage :

C'est une fonction mathématique ou autre, qui convertit une chaîne de caractères de longueur quelconque en une chaîne de caractères de taille fixe.

2) Fonction à sens unique :

C'est une fonction relativement aisée à calculer mais considérablement plus difficile à inverser. En d'autres termes, étant donné un x, il est facile de calculer $f(x)$, mais étant donné $f(x)$ il est extrêmement difficile de calculer x.

Les fonctions à sens unique sont très souvent utilisées pour l'authentification.

Roger NEEDHAM et Mike GUY ont réalisé qu'un ordinateur n'a pas besoin de connaître les mots de passe, mais doit juste distinguer entre un mot de passe correct, et un mot de passe incorrect.

Au lieu de stocker le mot de passe, l'ordinateur va stocker le résultat de la fonction à sens unique appliquée à chacun des mots de passe.

Lorsqu'une personne entre son mot de passe, l'ordinateur calcule le résultat de la fonction à sens unique appliquée sur ce mot de passe, et compare le résultat du calcul avec le contenu de sa base de données, et permettra l'accès.

L'attaque de quelqu'un qui pénètre dans un système informatique, et qui vole la liste des mots de passe n'est plus à craindre, car l'ordinateur ne stocke plus la table des mots de passe corrects mais la liste des résultats de l'application de la fonction à sens unique aux mots de passe.

Cette liste est inutilisable, car la fonction à sens unique ne peut être inversée.

3) *Fonction de hachage à sens unique H(M) :*

Cette fonction opère sur un message M de longueur arbitraire. Elle fournit une valeur de hachage ***h*** de longueur fixe ***m*** .

h=H(M), où ***h*** est de longueur ***m.***

Cette fonction a les caractéristiques suivantes :

Etant donné M, il est facile de calculer ***h***.

Etant donné ***h***, il est difficile de calculer ***M***.

Etant donné ***m***, il est difficile de trouver un autre message ***M'*** tel que ***H(M)=H(M')***.

➢ Différents protocoles de signatures :

Il existe différentes méthodes ou protocoles permettant de signer un document :

1) *Signature de documents à l'aide d'un cryptosystème à clé secrète et d'une tierce partie :*

Lila veut signer un message numérique et l'envoyer à Karim. Avec l'aide d'Ahmed (tierce partie digne de confiance) et d'un cryptosystème à clé secrète, elle peut le faire.

Ahmed peut communiquer avec Lila et Karim, il partage une clé secrète Ka avec Lila et une clé différente K_b avec Karim.

Le protocole est le suivant :

1° Lila chiffre son message pour Karim avec la clé Ka et envoie le résultat à Ahmed.

2° Ahmed déchiffre le message avec Ka.

3° Ahmed assemble le message déchiffré et un avis comme quoi il a reçu ce message de Lila. Ahmed chiffre le résultat avec K_b.

3° Karim déchiffre le tout avec K_b. Il peut maintenant lire le message et la certification d'Ahmed comme quoi Lila a bien envoyé ce message.

2) *Signature de documents à l'aide d'un cryptosystème à clé publique :*

Le protocole est le suivant :

1° Lila chiffre le document avec sa clé privée, signant ainsi le document.

2° Lila envoie le résultat à Karim.

3° Karim déchiffre le message avec la clé publique de Lila, vérifiant ainsi la signature.

Comme c'est le cas dans PGP, on peut réaliser ce type de signature avec RSA.

Lila chiffre son document en utilisant RSA avec sa clé privée (qu'elle seule connaît théoriquement), puis envoie le résultat à Karim.

Quant à Karim, il essaiera de décrypter le message avec la clé publique de Lila.

Si le déchiffrement fonctionne, cela veut dire que la signature a été "forgée" avec la clé privée de Lila, donc le message qu'il a reçu vient bien d'elle.

3) *Signature de documents à l'aide d'un cryptosystème à clé publique et d'une fonction de hachage à sens unique :*

Dans les applications pratiques, les algorithmes à clé publique sont souvent trop inefficaces pour signer de longs documents. Pour gagner du temps, les protocoles de signature numérique sont souvent réalisés avec des fonctions de hachage à sens unique. Au lieu de signer le document, Lila signe l'empreinte du document.

Le protocole est le suivant :

1° Lila, calcule à l'aide de la fonction de hachage à sens unique, l'empreinte de son document.

2° Lila chiffre, à l'aide de l'algorithme de signature numérique ,cette empreinte avec sa clé privée, signant ainsi par la même occasion le document.

3° Lila envoie le document et l'empreinte signée à Karim.

4° Karini calcule, à l'aide de la fonction de hachage à sens unique, l'empreinte du document que Lila a envoyé. Ensuite, à l'aide de l'algorithme de signature numérique, il déchiffre l'empreinte signée avec la clé publique de Lila. La signature est valide si l'empreinte de la signature est la même que celle qu'il a produite.

4. Mécanismes de contrôle d'accès :

Ces mécanismes peuvent utiliser l'identité authentifiée d'une entité ou des informations relatives à l'entité (telle que l'appartenance à un ensemble connu d'entités) ou des capacités de l'entité pour déterminer et appliquer ses droits d'accès.
Si l'entité essaie d'utiliser une ressource non-autorisée ou une ressource utilisée avec un type d'accès incorrect, la fonction de contrôle d'accès rejettera cette tentative et pourra en outre consigner l'incident afin de générer une alarme.
Les mécanismes de contrôle d'accès peuvent, par exemple, être basés sur l'utilisation d'un ou plusieurs éléments tels que : les bases d'informations de contrôle d'accès où sont gardés les droits d'accès des entités homologues.

5. Mécanismes de communication [1]:

Quand les objets sont transportés, il est spécialement gênant de préserver l'intégrité et la confidentialité et d'éliminer les canaux cachés. Par conséquent le transfert de l'information sensible oblige l'utilisation des techniques de chiffrement comme nous l'avons déjà vu ; malgré cela, différents problèmes subsistent tels que le problème des écoutes passives, la coupure et le déguisement.

Pour se confronter à ces problèmes, la sécurité de la communication se ramène à la construction d'un canal sécurisé entre les entités, cette construction commence par l'authentification mutuelle de ces entités ; la solution diffère selon la technique d'authentification retenue :

- Si le protocole cryptographique implanté n'utilise pas de serveur d'authentification, les deux entités désirant communiquer partagent une clé secrète. Cette clé secrète peut alors être utilisée pour le chiffrement et/ou la signature des messages.

- Si le protocole cryptographique utilise les serveurs d'authentification, les deux entités ne partagent pas de secrets. Pour permettre à ces deux entités de chiffrer et/ou de signer les messages, il est nécessaire de générer une clé de session ou clé temporaire. Cette opération est effectuée par le serveur d'authentification qui devient ainsi un serveur de clé de session. Cette clé est distribuée à chaque entité sous forme chiffrée à l'aide de la clé que le serveur a en commun avec chaque entité.

Lorsque le protocole n'utilise pas de serveur d'authentification, la solution paraît satisfaisante. Cependant, l'utilisation d'une telle méthode rend chaque communication statique : un espion enregistrant un message d'une communication entre deux entités données, peut rejouer ce message lors d'une communication future entre les mêmes entités. L'utilisation d'estampilles (signature) permet de se protéger contre une telle attaque. Malgré cela, l'utilisation d'un serveur d'authentification chargé de distribuer une clé de session pour les échanges entre les entités est généralement préférée pour les deux raisons suivantes :

- La clé étant modifiée régulièrement, le chiffrement des messages peut être considéré comme dynamique, c'est à dire qu'un même message est chiffré différemment pour deux sessions différentes.

- La clé temporaire ne servant que pour une session, les techniques de chiffrement et/ou de signature des messages après authentification ne nécessitent pas des outils cryptographiques aussi solides (c'est à dire, résistant longtemps à la cryptanalyse) que la phase d'authentification.

6. Conclusion :

A travers cette partie, nous avons pu avoir une vue d'ensemble sur les mécanismes les plus importants dans la sécurité informatique, afin de pouvoir par la suite rentrer dans un autre concept qui demande de la protection en faisant appel à l'un de ces différents mécanismes de sécurité. Il s'agit de l'anonymat dans le monde électronique qui sera étudié en détail dans le prochain chapitre.

Chapitre 3

L'anonymat et la confidentialité

1. Introduction:

L'Internet est devenu rapidement un outil important pour les communications modernes d'aujourd'hui et le commerce électronique.

Les importants soucis sont focalisés sur l'interception des communications par les étrangers pour les empêcher d'être à l'écoute des conversations électroniques. Même les messages cryptés peuvent être tracés en révélant qui est entrain de parler et à qui. Ce traçage est appelé analyse de trafic et peut révéler des informations sensibles. Par exemple, l'existence d'une collaboration inter-sociétés qui peut être confidentielle, les e-mails ...etc.

2. Problématique [6]:

Les lettres envoyées à travers les bureaux de postes sont habituellement dans une enveloppe marquée avec l'adresse de l'expéditeur et du destinataire.

On a confiance que les bureaux de postes ne regardent pas ce qui est à l'intérieur de l'enveloppe parce que le contenu de l'enveloppe est privé. On a aussi confiance que les bureaux de postes ne contrôle pas qui envoie les courriers et à qui, parce que cette information est aussi considérée privée.

Ces deux informations sensibles : le contenu de l'enveloppe et son adresse, s'appliquent également bien aux communications électroniques sur l'Internet et le web. Comme le web est devenu une partie importante dans les communications modernes d'aujourd'hui et le commerce électronique, la protection de la confidentialité des messages électroniques est devenue de plus en plus importante. Exactement comme les courriers, les messages électroniques parcourent dans des enveloppes électroniques. La protection de la confidentialité des messages électroniques exige à la fois la protection du contenu des enveloppes et de cacher les adresses dans leurs enveloppes. Il n'y a pas de raison que l'utilisation du réseau public comme Internet faille révéler aux autres qui sont entrain de parler, à qui et de quoi parlent-ils. Les plus importants soucis sont : l'analyse de trafic et l'interception des communications.

Dans ce chapitre, nous allons mettre le point sur les deux éléments les plus importants dans la protection des communications et les échanges de données dans le monde électronique d'aujourd'hui, qui sont la confidentialité des informations échangées et surtout l'anonymat des parties communicantes, puis nous étudierons la relation qui existe entre eux. Après notre

étude portera sur les différents aspects de l'anonymat, et en fin les différentes techniques utilisées pour assurer l'anonymat dans le monde électronique.

3. L'Anonymat et la Confidentialité [7]:

L'anonymat est fortement lié au concept d'identité. Pour définir l'anonymat nous allons en premier lieu comprendre la notion d'identité.

3.1. L'identité :

C'est la condition d'être une personne spécifiée ou bien la condition d'être soi même et non quelqu'un d'autre.

L'identité dans les systèmes informatiques est un ensemble d'informations sur une entité qui le diffère des autres entités similaires. L'identité est souvent requise pour permettre la confiance et la confidentialité dans le monde électronique.

L'identification veut dire qu'une personne fournit quelques informations sur son identité. On assume puisque la personne est capable de fournir ces informations c'est elle la personne qui stipule.

Ceci ne fournit pas un haut niveau de confiance, car il est facile d'avoir ces informations et les moyens nécessaires pour stipuler qu'on est quelqu'un d'autre. Par exemple, en découvrant le numéro de la carte de crédit de quelqu'un. Pour garantir un haut niveau de confiance, l'authentification est utilisée. L'authentification exige à une personne de fournir quelques informations supplémentaires qui lui sont propres et connues par elle-même, quand cette personne fournit correctement les informations demandées elle est autorisée à faire certaines actions dans le système informatique.

Malgré cela, il y a des personnes qui préfèrent ne pas être reconnues dans beaucoup de situations.

3.2. Définition de l'anonymat :

L'anonymat signifie que la personne ne révèle pas sa vraie identité.

3.3. Problèmes avec l'anonymat :

Puisqu'on ne peut se rendre compte de ce que fait une personne anonyme, l'anonymat attire à commettre des activités illégales et soupçonneuses, ceci est vu comme un problème majeur, et souvent utilisé comme argument contre l'anonymat. Un autre argument souvent utilisé contre l'anonymat est que beaucoup d'interactions, par exemple le retrait de l'argent d'un compte, exigent la confiance et la crédibilité, et ceci requière que la personne doive être identifiable.

Malgré cela, la confiance et la crédibilité peuvent ne pas exiger nécessairement l'identification de l'utilisateur. Comme une personne peut avoir différentes identités, l'utilisateur peut utiliser différents pseudo-identités dans des situations différentes. La responsabilité devrait être garantie par une tierce partie. A travers les informations que la tierce partie possède, la vraie identité de l'utilisateur pourrait être associée au pseudonyme utilisé. Donc l'identité réelle pourrait être déterminée s'il y a raison valable pour cela. Les pseudonymes offrent un bon moyen pour rester anonyme, mais ils limitent effectivement la possibilité de tricher. L'idée des pseudonymes a été présentée par DAVID CHAUM en 1980.

Dans la société, il y a différentes opinions sur l'importance et le rôle de l'anonymat, certains le voient comme un risque, d'autres le voient comme droit légitime. Mais d'une manière générale l'anonymat apporte des avantages et des inconvénients.

3.4. La Confidentialité :

La confidentialité est un moyen pour assurer la protection des informations privées contre toute divulgation non autorisée. La confidentialité est vue comme un besoin de base pour l'humanité car la plupart des gens détestent vivre dans un monde où n'importe qui peut savoir toute chose sur quelqu'un d'autre.

3.5. La relation entre l'anonymat et la confidentialité :

L'anonymat est fortement lié au concept de l'identité, quand une personne est anonyme, elle ne révèle pas sa véritable identité.

La confidentialité est encore le contrôle de l'information personnelle et l'espace personnel.

L'identité est quelque chose de personnel. Dans le monde électronique l'identité est déterminée en fournissant une partie d'information personnelle aux autres. Ainsi la protection de l'identité est une partie de la confidentialité d'où :

- L'anonymat peut être vu comme une partie de la confidentialité :

Si une partie d'information par laquelle un utilisateur peut être identifié est fournie, la donnée personnelle peut être enregistrée et faire référence à cette personne plus tard. La donnée peut être classée sous forme de cross-référence (table de correspondance) si plusieurs sources utilisent la même information d'identité. Donc l'identité simplifie l'extraction des informations personnelles et affaiblit la confidentialité. Ainsi si on reste anonyme on peut renforcer le niveau de la confidentialité.

- L'anonymat n'est pas nécessairement suffisant pour garantir la confidentialité :

Si on est anonyme, mais la fourniture d'informations sur soi même ou sur son comportement est traquée, il est possible de déterminer son identité basée sur informations ou faire la correspondance entre l'information avec d'autres informations contenant son identité à l'aide des tables de correspondance (cross-référence).

3.6. Caractéristiques de l'Anonymat :

Un service d'anonymat est un service qui assure la propriété d'anonymat aux utilisateurs. Il réduit la capacité d'un adversaire à déduire les vrais points finaux d'une communication sur un réseau quelconque. Un réseau anonyme est un réseau de nœuds qui communiquent en utilisant la propriété d'anonymat. Si par exemple A décide d'envoyer un message à B en utilisant un service d'anonymat, on dit que :

- Le service de l'anonymat fournit l'anonymat de l'émetteur si l'adversaire est incapable de conclure que la source du message est A.
- Le service de l'anonymat garantit l'anonymat du destinataire si l'adversaire est incapable de conclure la destination du message.
- Le service de l'anonymat garantit une indépendance émetteur/récepteur si étant donné de multiples messages anonymes envoyés de A à B, l'adversaire est incapable de déterminer si les messages se réfèrent à la même source et à la même destination.

D'une manière générale, l'anonymat est qualifié de :

- Anonymat faible : s'il assure l'intraçabilité c'est à dire qu'il est impossible de faire un lien entre l'origine de l'objet et son destinataire.

- Anonymat fort : si à l'intraçabilité, est ajoutée l'indépendance où il est impossible de lier deux actions d'un même individu.

4. Techniques d'Anonymat :

Comme le développement de la technologie de l'information a imposé de nouvelles menaces sur la confidentialité et l'anonymat, il a aussi apporté de nouveaux mécanismes pour les protéger. La cryptographie garantit un certain niveau de confidentialité, mais l'utilisateur n'est pas anonyme.

Des techniques sont proposées pour assurer l'anonymat et la confidentialité des communications sur Internet.

4.1. Les remailers [7]:

Les remailers permettent aux utilisateurs d'utiliser le courrier électronique et les services d'information sans révéler leurs identités.

Un remailer prend le courrier original, enlève son entête et renvoie le corps du message avec une nouvelle identité.

Principalement, il existe trois type de remailers :

❖ Type 0 :

Ces remailers supportaient l'anonymat en enlevant les entêtes d'identification des messages en partance et en fournissant un pseudonyme aléatoire pour le destinataire. Le serveur maintient une table secrète d'identité pour relier les pseudonymes avec leurs adresses réelles. La sécurité de ces remailers est plutôt faible. En effet, les utilisateurs doivent avoir confiance que leurs identités ne sont pas révélées lors de l'envoi de leurs courriers à travers ces remailers. Mais si la table secrète d'identité est rendue publique l'anonymat de l'utilisateur est rompu. Ce qui constitue le point faible de ce type de remailers.

Anon.penet.fi est un exemple d'un tel type. Le serveur de Anon.penet.fi est ordonné de révéler l'identité des pseudonymes en cas d'une enquête judiciaire. Il est possible de révéler aussi la vraie identité des pseudonymes par l'observation des messages entrants et les messages sortants.

Ce type de remailers est rarement utilisé de nos jours.

❖ **Type 1 (ou cypherpunk remailers) :**

Ils n'utilisent pas les pseudonymes et les listes d'identité, et évitent de garder les fichiers log (fichiers de stockage d'informations), ces remailers acceptent des courriers cryptés, ils les décryptent puis renvoient le message résultant. Ce type de remailers utilise une chaîne de serveurs pour atteindre une sécurité robuste. Le premier serveur reçoit le message crypté (en PGP), il l'intégrera dans plusieurs couches contenant dans chaque couche l'adresse du prochain serveur de la chaîne suivie, cryptée avec la clé publique du serveur correspondant. Chaque serveur recevant le message va le décrypter avec sa clé privée. La prochaine étape du processus se déroulera de la même manière jusqu'à ce que le point final du chemin reçoit le message en clair...

Ces remailers peuvent aussi réordonner les messages sortants de manière aléatoire pour les protéger contre les attaques des espions.

❖ **Type 2 (Mixmasters remailers) :**

Ou tout simplement appelés les Mixes. Ils sont les plus récents et les plus sophistiqués. Ils sont plus sécurisés que ceux du type1.

Un Mix est une entité qui relie des messages. Il fonctionne selon un principe où le relayage est constitué de façon à ce qu'il n'y ait aucune corrélation possible (temporelle et volumique) entre les messages d'arrivée et ceux de sortie.

Lorsqu'une entité pratique l'analyse de trafic circulant sur un réseau donné dans le but d'espionnage, elle peut appliquer 4 attaques sur ce trafic :

- 2 attaques passives basées sur :
    - La corrélation des données : contenu du message et sa longueur. Avec les Mixes la solution est de crypter les messages entrants et de leur rajouter du padding (bourrage) afin que tous les messages sortants aient la même longueur.
    - La corrélation temporelle : si les messages entrants sortent du Mix dans le même ordre, l'espion peut alors lier l'émetteur et le récepteur. La solution est d'adjoindre un faux trafic au vrai quand cela est nécessaire dans une limite de temps donnée.
- 2 attaques actives :
    - Isolation et identification : en connaissant le nombre de messages n à transmettre pendant un temps donné, l'espion peut envoyer au Mix n-1 messages afin d'en isoler un seul. Mais bien qu'il ne puisse pas décoder le message, l'espion pourra

lier l'émetteur du message et le récepteur. Cette technique est très dure à mettre en place, car elle nécessite un contrôle total de l'entrée et de la sortie du Mix.

- Rejeu de message : l'espion peut aussi enregistrer un certain nombre de messages allant au Mix, puis les lui renvoyer. La solution consiste à placer un ticket d'horodatage dans les messages échangés.

Lorsqu'on utilise un réseau de remailers, 2 configurations sont possibles :

- En chaîne : on utilise le principe du routage onion (onion routing qui sera étudié par la suite).
- En cascade : le chemin pris par les données n'est pas connu par l'utilisateur, chaque Mix suivant est choisi par le Mix courant.

Les remailers fournissent un très bon niveau d'anonymat et si le cryptage est utilisé le niveau de confidentialité est élevé. Mais reste toujours l'inconvénient de l'insécurité de connexion puisque le fournisseur de ce service peut intercepter les messages des utilisateurs puisqu'on est obligé de lui révéler la destination. Ceci réduit la sécurité de la chaîne entière, et le problème de la sécurité de connexion va se poser.

4.2. La navigation anonyme sur le web :

Il existe plusieurs outils développés pour la navigation anonyme sur le web. Le but est de ne pas fournir l'adresse IP ou d'autres informations sur l'utilisateur au serveur web consulté.

4.2.1. Anonymizer :

(www.anonymizer.com). C'est un web-proxy qui élimine par filtrage les entêtes des identités et les adresses sources à partir des requêtes envoyées. L'anonymat offert par anonymizer est plutôt faible, puisqu'on peut facilement déterminer qui a demandé quoi.

4.2.2. Les Web-Mix [8]:

Le Web-Mix est une architecture utilisant le système de Mix vu auparavant mais orienté vers la navigation anonyme et non observable. Il est constitué de 3 parties :

- JAP : Java Anon Proxy ;
- un réseau de Mixes ;
- un proxy de sortie.

Ces trois parties constituent alors un tunnel anonyme pour les données du client échangées avec le serveur Web.

Le JAP est un proxy installé directement chez l'utilisateur, qui se connecte directement au premier proxy MIX via Internet. Il possède les fonctionnalités suivantes :

- Enregistrement du client.
- Filtrage des données venant de l'extérieur (cookie, Java script, Active X)
- Découpage des données venant de l'utilisateur pour les Mixes.
- Génération et réception d'un faux trafic lorsque l'utilisateur ne fait rien.

Les Web-Mixes sont utilisés en mode cascade comme décrit auparavant, ils sont simplement connectés entre eux. Ils reprennent les mêmes fonctionnalités que les Mixes précédents mais dans une optique temps réel. Le réseau constitue alors un support qui empêche toute observabilité. On ne peut savoir quelle requête apparaît à cet utilisateur grâce au faux trafic mélangé avec le vrai.

4.2.3. Crowd [6]:

Le système Crowd est un autre moyen pour assurer la navigation anonyme sur le web. L'idée principale de Crowd est que les gens peuvent être anonymes lorsqu'ils sont mélangés dans une foule. Les utilisateurs de Crowd émettent leurs requêtes à travers une foule (crowd) qui peut être un groupe d'internautes parcourant un logiciel Crowd. Les requêtes sont aléatoirement envoyées à un serveur Crowd en éliminant les entêtes qui incluent les informations sur l'origine de cette requête de telle sorte que personne ne connaîtra l'origine de cette requête.

4.3. Onion Routing [6]:

Onion routing est une technique pour protéger une variété de service Internet contre d'éventuelles attaques en l'occurrence l'analyse de trafic et l'interception des communications par des intrus.

L'analyse de trafic est utilisée pour déterminer qui est entrain de parler et à qui sur un réseau. Par exemple dans un réseau à commutation de paquets, les paquets ont une entête utilisée pour le routage et une partie contenant l'information utile. L'entête qui devrait être visible dans le réseau révèle la source et la destination du paquet. Même si l'entête est obscure dans quelque chemin, le paquet peut être traqué puisqu'il se déplace à travers le réseau. Le cryptage de la partie contenant la partie utile est aussi inefficace, par ce que le but de l'analyse de trafic est d'identifier les deux parties communicantes et ne s'intéresse pas au contenu de la conversation.

5. Conclusion :

L'anonymat est un concept qui fait face à la fois à l'espionnage et à l'analyse de trafic. Une connexion anonyme ne doit pas révéler au réseau et aux observateurs du réseau qui est entrain de parler et à qui, et si les données qui trafiquent sur le réseau ne contiennent pas des informations d'identification, la communication est aussi anonyme.

A travers ce chapitre nous avons défini la notion d'anonymat et sa relation avec la confidentialité ainsi que les différentes techniques permettant d'assurer l'anonymat et la confidentialité de plusieurs services Internet.

Chapitre 4

Onion Routing

1. Introduction :

En utilisant l'analyse de trafic, il est possible de déterminer qui est entrain de parler à qui dans un réseau public. Par exemple dans un réseau à commutation de paquets, les paquets ont une entête utilisée pour le routage et une partie contenant l'information utile (charge utile). L'entête qui devrait être visible dans le réseau révèle la source et la destination du paquet. Même si l'entête est obscure dans quelques chemins, le paquet peut être encore traqué puisqu'il se déplace à travers le réseau . Le cryptage de la charge utile est aussi inefficace, parce que le but de l'analyse de trafic est d'identifier les deux parties communicantes et ne s'intéresse pas au contenu de la conversation.

Pour faire face à ce type de menaces, il a fallu penser à une méthode qui peut assurer l'anonymat sur un réseau public, d'où a pris naissance " Onion Routing ".

Onion Routing est une infrastructure qui fournit des connexions anonymes à temps réel et bidirectionnelles, résistant à l'analyse de trafic et l'espionnage. Cette technique a été proposée par le " Naval Research Laboratory " des USA, et elle a été implémentée par Sun Microsystems pour assurer l'anonymat dans plusieurs services Internet (E-mail, le Web, le transfert de fichiers, Remote logins).

Dans ce présent chapitre, nous allons décrire cette technique.

2. Objectifs de cette technique [9]:

Onion Routing est une infrastructure qui :

- Complique l'analyse de trafic.
- Sépare l'identification du routage.
- Supporte plusieurs applications différentes.

Sans des liens dédiés entre chaque nœud et l'utilisation de tous les liens, l'analyse de trafic peut en principe être toujours efficace, mais il peut être plus coûteux. Onion routing accomplit ce but en séparant l'identification du routage. Au lieu de contenir des informations sur la source et la destination, les paquets se déplaçant le long d'une connexion anonyme contiennent seulement l'information sur le prochain et le précédent nœud. Ces connexions anonymes peuvent remplacer les connexions par socket puisque celles-ci sont généralement

utilisées pour supporter les applications fonctionnant sur Internet (comme navigation Web, Remote login et l'E-mail ...etc.) et les connexions anonymes par Onion routing peuvent aussi supporter une grande variété d'applications non modifiées en utilisant des proxies qui servent d'interfaces entre les applications et le réseau Onion routing.

La conception d'Onion est très conservatrice puisqu'elle suppose que le réseau public est vulnérable. En particulier, on assume que :

- Tout le trafic est visible.
- Tout le trafic peut être modifié.
- Les Onion routers peuvent être corrompus.
- Les Onion routers corrompus peuvent coopérer.

3. Onion routing :

3.1. La structure [9]:

Onion routing a deux parties : une infrastructure réseau qui supporte les connexions anonymes et les interfaces proxy qui relient ces connexions.

3.1.1. L'infrastructure réseau :

Le réseau public contient un ensemble d'onion routers (routeurs utilisés pour les connexions anonymes par onion routing). Chaque onion routers est relié à un petit ensemble d'onion routers voisins. La communication entre deux onion routers voisins s'effectue par des simples connexions socket dont les entêtes de paquets de données échangés sont cryptés par des fonctions cryptographiques connues par ces derniers, et les paquets sont routés à travers plusieurs sauts par le protocole IP.

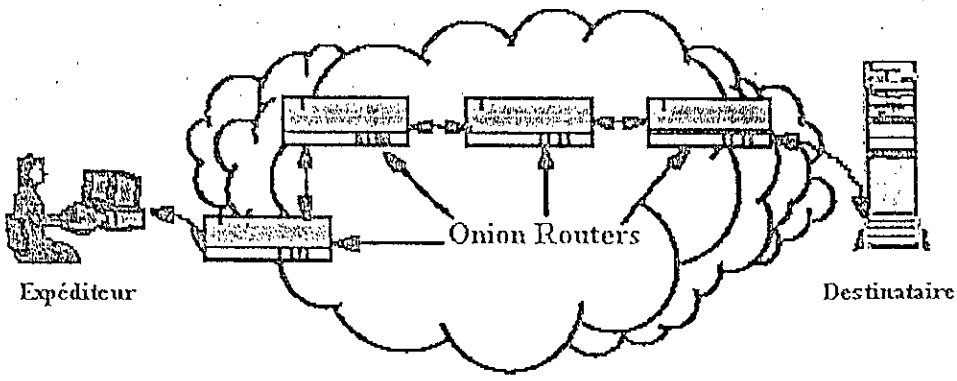


Figure 4.1 : L'infrastructure réseau d'onion routing

Une connexion anonyme est routée à travers une séquence d'onion routers voisins (voir figure 4.1). Les segments communs de ces routes sont multiplexés sur une simple connexion entre deux voisins. Le rôle d'onion router est de faire passer la donnée d'une connexion à une autre après avoir appliqué des opérations cryptographiques appropriées.

3.1.2. L'interface proxy :

Les proxies interfacent entre les applications et l'infrastructure réseau. Dans Onion routing, les fonctions des interfaces proxies se divisent en deux : une partie relie l'expéditeur à la connexion anonyme et une autre partie relie la connexion anonyme au destinataire. Imaginons par exemple : un expéditeur dans sa station de travail utilise un navigateur web, lorsqu'il clique sur un lien hypertexte le navigateur web envoie une requête http correspondant à l'URL de ce lien à un certain proxy onion routing au lieu de l'envoyer directement au destinataire. Soit W ce proxy. W analyse la requête et choisit aléatoirement une route à travers plusieurs onion routers (par exemple W-X-Y-Z), W envoie alors l'onion qu'il a construit le long de cette route. L'onion contient donc le paquet de données à envoyer vers la destination plus des instructions sous forme de couches aux onion routers initialement choisis pour construire la connexion anonyme. Le dernier onion router de la route (Z) fonctionne aussi comme un proxy onion routing pour le destinataire. Z fait passer donc les données de la connexion anonyme vers le destinataire et inversement (figure 4.2)

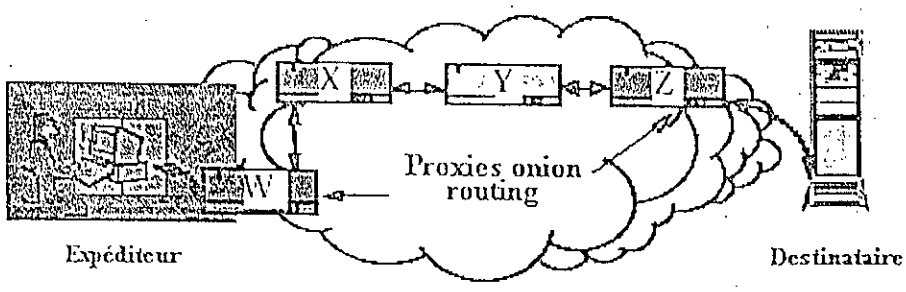


Figure 4.2 : Les interfaces proxy d'onion routing

Au lieu d'une simple connexion socket entre l'expéditeur et le destinataire, onion routing requière une connexion socket entre l'expéditeur et son proxy onion routing, une connexion anonyme entre le proxy expéditeur et le proxy du destinataire, et une connexion socket entre le proxy du destinataire et le destinataire lui même.

Il existe plusieurs configurations du réseau onion routing. Dans une configuration de base, un site qui est concerné par l'analyse de trafic devrait contrôler un proxy onion routing pour protéger la communication entre ce site et ses utilisateurs (figure 4.3). Ce proxy doit aussi fonctionner comme un onion router intermédiaire dans d'autres connexions anonymes, s'il n'est pas utilisé de cette manière, les observateurs peuvent surveiller le flux de données venant du proxy et le tracent au site sensible. Cependant, si le proxy onion routing est aussi un onion router intermédiaire ils ne peuvent pas avoir des informations concernant le site sensible.

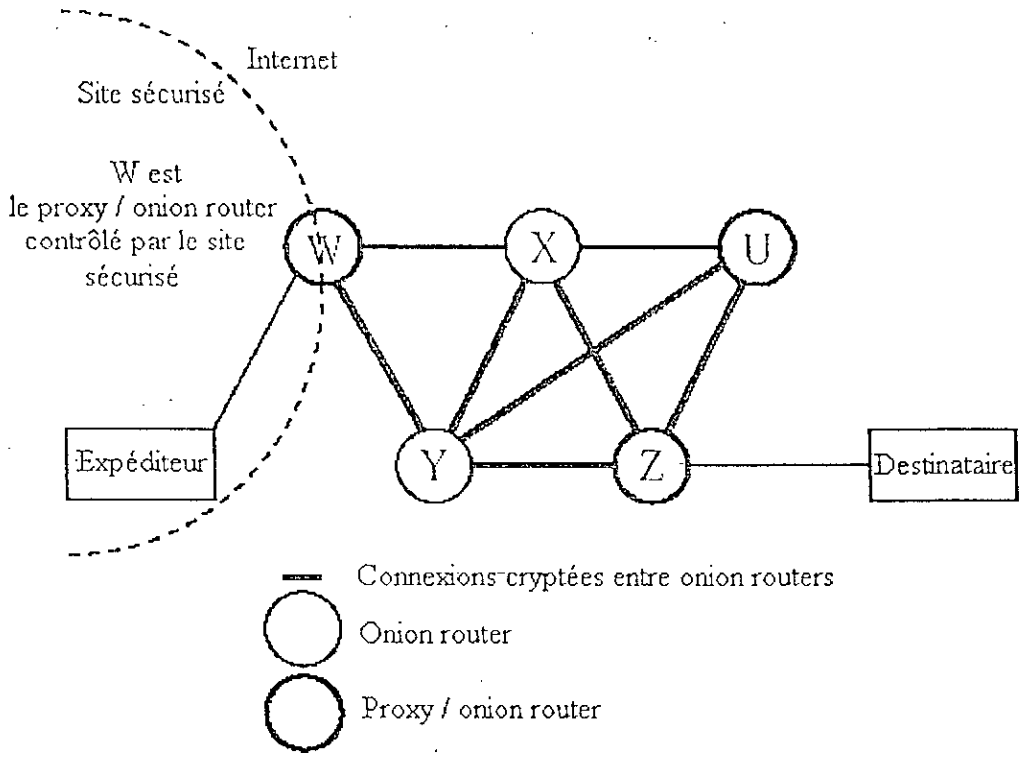


Figure 4.3 : Topologie de routage contrôlée par un site sécurisé

3.2. Le détail d'onion [9,10]:

Pour débuter une session entre un expéditeur et un destinataire, le proxy de l'expéditeur choisit une série de nœuds (onion routers) pour former une route à travers le réseau, et construit un onion qui encapsule les informations concernant cette route. La figure 4.4 illustre un onion construit par le proxy de l'expéditeur W pour une route anonyme vers X et Y. Le proxy de l'expéditeur W envoie donc l'onion le long de cette route et établit un circuit virtuel entre lui même et le proxy du destinataire Z.

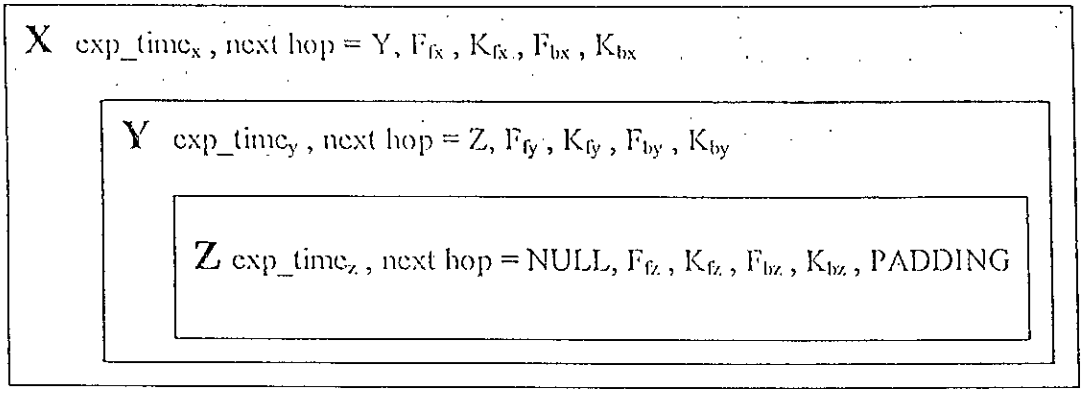


Figure 4.4: Structure d'onion

La structure des données dans l'onion est composée de plusieurs couches cryptées enveloppant la charge utile qui est au centre de l'onion. La structure de l'onion dépend de la route choisie par le proxy de l'expéditeur vers le proxy du destinataire.

En se basant sur cela, le proxy de l'expéditeur crypte en premier lieu pour le proxy du destinataire (la couche la plus basse), puis pour le nœud qui le précède dans la route, et continue de cette manière jusqu'au premier nœud à qui il va envoyer l'onion. Il va utiliser dans chaque couche une fonction cryptographique asymétrique pour crypter les informations utiles au nœud correspondant par la clé publique de ce dernier, et une fonction cryptographique symétrique et donc une seule clé pour le cryptage et décryptage pour la charge utile (ou payload) à envoyer au prochain nœud. La paire (fonction, clé) utilisée sera spécifiée dans la partie information à l'intérieur de l'onion afin que les onion routers puissent décrypter la charge utile reçue et l'envoyer correctement au prochain nœud. Dans ce cas, chaque onion router connaîtra qui lui a envoyé cet onion et à qui il doit le passer. Mais il ne connaîtra rien sur les autres nœuds constituant la chaîne ainsi que sa place dans cette chaîne (sauf s'il est le dernier).

Ce que le nœuds P_x reçoit comme données , ressemble à ceci :

{ exp time, next hop, F_f, K_f, F_b, K_b, Payload }PK_x

Ici PK_x est la clé publique du nœud P_x et Payload veut dire charge utile, en assumant que P_x possède la clé de décryptage. Le message décrypté contient un temps d'expiration de l'onion (exp time), le prochain nœud de routage auquel la charge utile va être envoyée (next hop), deux paires fonction/clé spécifiant l'opération cryptographique utilisée et la clé qui va être appliquée à la donnée à envoyer le long du circuit virtuel {F_f, K_f} et {F_b, K_b} telles que la

paire $\{F_f, K_f\}$ est appliquée aux données circulant dans le sens direct (de l'expéditeur vers destinataire), la paire $\{F_b, K_b\}$ est appliquée aux données circulant dans le sens opposé, et enfin la charge utile (payload).

Si le nœud récepteur est le proxy du destinataire alors le champ next hop = NULL.

Le temps d'expiration est utilisé pour détecter les faits de rejouer dont les nœuds corrompus pouvant être utilisés pour traquer les messages. Chaque nœud garde une copie de l'onion pendant son temps d'expiration, s'il reçoit une autre copie du même onion pendant ce temps là il l'ignore et s'il reçoit un onion après son temps d'expiration il l'ignore aussi.

Notons qu'à chaque nœud l'onion se réduit puisqu'une couche sera pelée. Ceci est un problème puisqu'on peut facilement déterminer la route suivie par les données en observant la diminution monotone de la taille de l'onion. Pour éviter ce genre de problèmes, des bits aléatoires sont ajoutés à la place des couches pelées à la fin de la charge utile avant l'envoi de l'onion. Aucun nœud, sauf le dernier, ne connaîtra la quantité de bits (padding) ajoutés à la charge utile puisqu'il ne connaît pas sa place dans la chaîne, il décrypte simplement ces bits avec le reste de l'onion sans faire la différence entre eux.

Même avec une taille constante de l'onion on peut tracer son chemin, à moins que tous les les onions aient la même taille, donc on fixe une taille de l'onion. Pour maintenir cette taille constante pour cacher la longueur de la chaîne, le proxy de l'expéditeur va bourrer la charge utile en fonction de la taille de l'onion. Ainsi, n'importe quel onion qui arrive au proxy du destinataire aura toujours la même quantité de bits ajoutés que ce soit initialement ou en cours de route.

3.3. Création du circuit virtuel [10]:

Le but dans l'envoi de l'onion est de produire un circuit virtuel avec des connexions cryptées entre les différents onion routers. Pour créer un circuit virtuel entre les différents nœuds, les messages transmis entre eux doivent contenir un identificateur de chaque nœud, une commande de type : create, destroy, data) et les données à transmettre. N'importe quelle autre commande différente de celle-ci est considérée comme une erreur et le nœud qui reçoit une telle commande l'ignore à moins qu'il envoie une commande "destroy" à travers ce circuit virtuel. La commande "create" accompagne l'onion. Quand un nœud reçoit une telle commande avec un onion, il choisit un identificateur et envoie un autre message de création contenant cet identificateur au prochain nœud et l'onion après avoir pelé sa couche. Le nœud

store aussi l'identificateur qu'il reçoit et son identificateur qu'il envoi comme une paire jusqu'à la destruction du circuit. La figure 4.5 illustre un circuit virtuel qui construit l'onion de l'exemple pris précédemment (pour le chemin W-X-Y-Z).

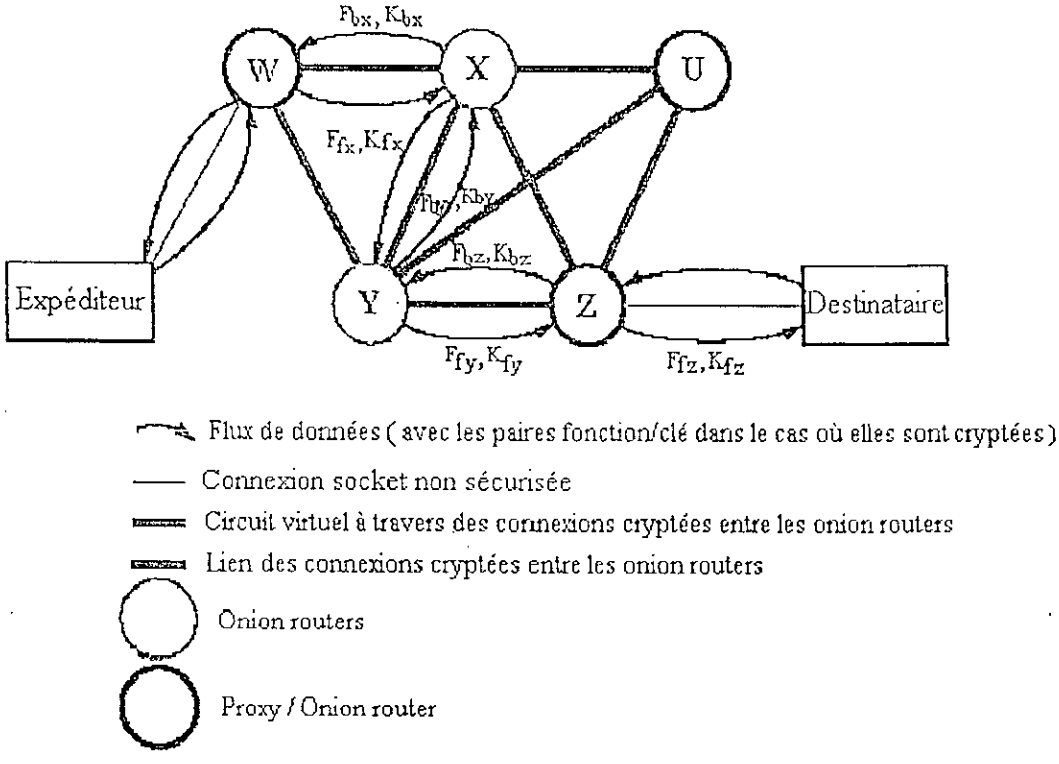


Figure 4.5 : Le circuit virtuel

3.4. Construction et transition de l'onion [9]:

En se basant sur l'exemple pris précédemment, les données envoyées par l'expéditeur à travers le circuit virtuel construit sont pré-cryptées à plusieurs reprises par son proxy W comme suit : (Voir figure 4.6)

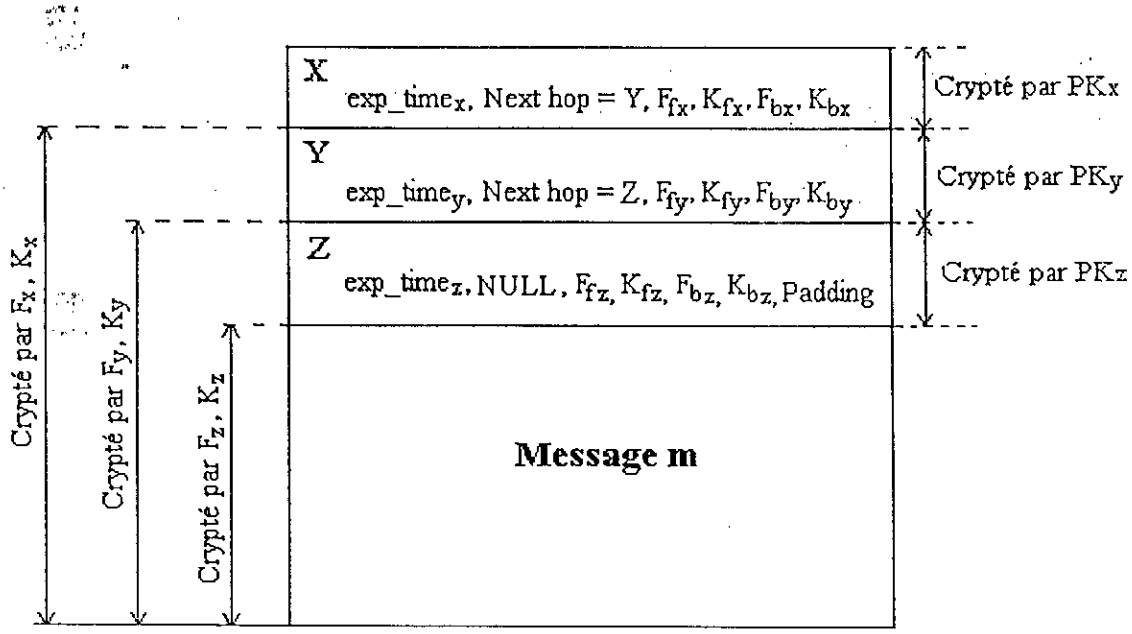


Figure 4.6 : *Construction des couches de l'onion*

Pour le proxy du destinataire Z, W crypte les données de son expéditeur par une fonction cryptographique symétrique F_z avec la clé secrète K_z. Puis il crypte à l'aide d'un algorithme asymétrique la donnée suivante : {exp time, next hop, F_f, K_f, F_b, K_b}, par la clé publique de ce proxy (Z). Ceci va constituer la couche la plus basse de l'onion. Il va procéder de la même manière pour chaque onion router jusqu'au premier (dans ce cas c'est X) qu'il va lui envoyer un onion complet contenant les couches cryptographiques qui vont être pelées jusqu'à ce que le destinataire reçoive le message en clair comme suit :

Quand un onion router reçoit les données dans cette connexion, avant de les envoyer au prochain nœud correspondant, il décrypte d'abord par sa clé privée la couche qui lui correspond. Celle ci contient les informations nécessaires dans lesquelles il trouvera la fonction cryptographique et la clé secrète avec laquelle il décryptera le reste de la donnée reçue et l'enverra au prochain nœud. Dans ce cas, s'il trouvera une paire {F_f, K_f} ceci veut dire que les données transitent dans le sens direct, sinon (s'il trouve {F_b, K_b}) les données transitent dans le sens inverse (voir figure 4.7).

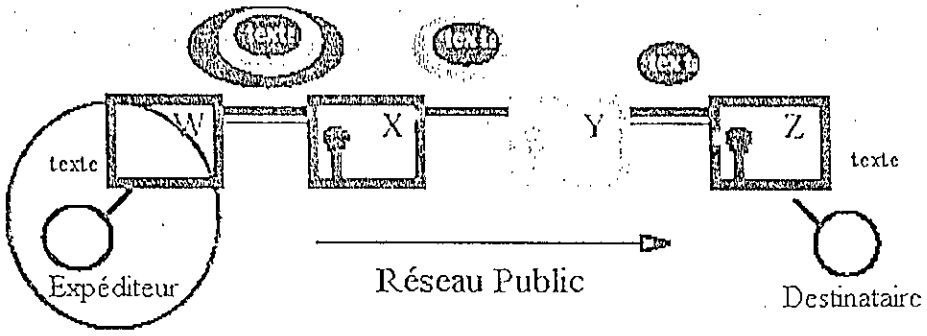


Figure 4.7 : Transition des données dans le sens direct

Remarque :

1- Les algorithmes symétriques sont connus par leur rapidité avec le seul défaut de la clé partagée qui peut être révélée. De leur côté, les algorithmes asymétriques sont robustes mais ils sont trop lents et génèrent des données de grande taille. C'est pour cela qu'on crypte les données envoyées par des fonction symétriques et les informations concernant chaque couche par un algorithme asymétrique.

2- Les identificateurs utilisés pour créer le circuit virtuel sont cryptés en utilisant des algorithmes à clé partagée par les onion routers.

3.5. Destruction de la connexion anonyme [9]:

Lorsqu'une connexion socket est interrompue, la connexion anonyme doit être détruite. Un onion router qui décide de couper la connexion envoie un message de destruction dans les deux sens le long de la connexion anonyme, il efface aussi ses propres tables de correspondances. Un onion router qui reçoit un message de destruction est obligé aussi d'effacer ses propres tables et d'envoyer le message dans la même direction.

4. Conclusion :

Onion routing est une architecture qui fournit une connexion anonyme entre un émetteur et un récepteur, en temps réel et bidirectionnelle pour résister aux tentatives d'espionnage ainsi

qu'à l'analyse de trafic. Cette connexion anonyme peut remplacer les connexions socket dans une large variété d'applications utilisant des proxies telles que : Le Web, le transfert de fichiers, remote logins, le courrier électronique, telnet…etc.

En fait, le but principal d'onion routing est de compliquer l'analyse de trafic en se basant sur une technique très efficace pour cela. Cette technique a été traitée le long de ce chapitre.

Onion Routing a été implémenté par sun solaris, et les proxies onion routing qui servent d'interface entre l'utilisateur et le réseau des onion routers ont été implémentés pour supporter plusieurs services tels que : le web (http), le courrier Electronique (SMTP), le remote login (RLOGIN), le transfert de fichiers (FTP), sachant que chaque proxy et onion router ont été conçu suivant le protocole de la méthode. De ce fait, on remarque bien que l'implémentation de cette technique n'est pas à la portée de tout le monde, seules les grandes compagnies qui trouvent un intérêt dans son implémentation peuvent la faire. Pour les Internautes, Onion routing n'est qu'un service proposé par son concepteur qui peut être accessible à travers le site www.onionrouter.net. Donc, le problème de la connexion anonyme se reposera encore, puisque l'anonymat entre expéditeur et fournisseur du service est rompu puisqu'il peut connaître la destination.

Chapitre 5

Les remailers

1. Introduction :

Nous pouvons souhaiter envoyer un courrier de façon anonyme c'est à dire sans divulguer notre identité au destinataire ou toute personne souhaitant connaître la destination de nos courriers. Bien sûr, nous pouvons modifier notre adresse e-mail dans notre navigateur mais l'entête de notre courrier comporte de nombreuses informations dont l'adresse IP.

A partir de notre adresse IP, les bases d'interrogation spécialisées peuvent divulguer notre identité ou celle de notre fournisseur d'accès (qui enregistre nos connexions dans ses fichiers log).

De plus, notre fournisseur d'accès Internet (FAI) peut lire nos courriers, il convient donc, pour une plus grande confidentialité, en plus de cacher notre adresse IP, de crypter nos messages entrants et sortants. C'est le rôle dévolu aux remailers ou ré-acheminеurs, qui cryptent nos courriers sortants tout en dissimulant les informations pouvant nous identifier, dont notre adresse IP puisqu'ils la substituent par les siennes. Nous allons étudier ceci donc à travers ce chapitre.

2. Qu'est ce qu'un remailer ?

Anglicisme pour "Ré-expéditeur ou ré-acheminеur Anonyme ". Un remailer est un serveur qui nous propose de recevoir notre courrier et de le réexpédier en faisant comme si c'était lui qui en était à l'origine, en dissimulant notre identité. Du coup, nous sommes anonymes. Le remailer le plus connu fut "anon.penet.fi" (voir chap. 3 type 0), mais il a été attaqué de toutes parts par tous ceux qui n'aiment pas que des gens puissent s'exprimer aussi librement, et il a fini par fermer. Depuis, d'autres systèmes ont été mis en place, qui sont nettement plus sophistiqués (il s'agit des types 1 et 2).

3. Y a-t-il beaucoup de remailers ?

La réponse est oui, puisqu'il existe actuellement une cinquantaine de remailers situés un peu partout dans le monde scindés en deux familles (types) : les cypherpunks et les Mixmaster. Ces derniers découpent les messages en morceaux de taille constante, rendant la

surveillance des messages plus difficile et le niveau de sécurité s'en trouve accru par rapport aux premiers.

Nous pouvons trouver une liste de remailers à l'adresse :

http:/www.cs.berkeley.edu/~raph/remailer~list.html.

La page affichée est illustrée par la figure 5.1.

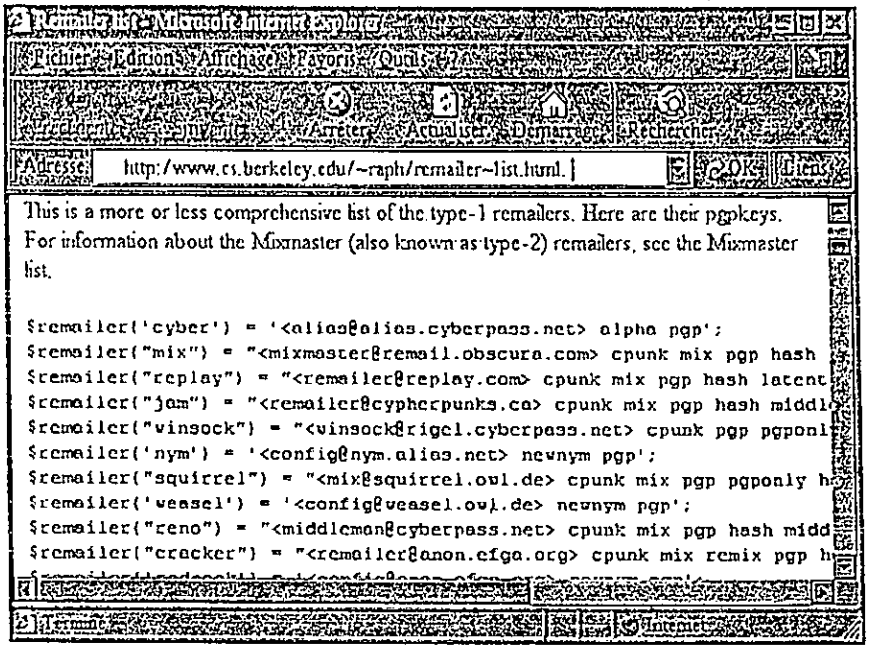


Figure 5.1 : la liste des remailers sur le site www.cs.berkeley.edu

On trouvera dedans une liste de vingt remailers, leurs caractéristiques, ainsi qu'une table de résultats tests effectués sur certains.

On trouvera aussi une autre liste à l'adresse :

http:/www.anon.efga.org/type-I (cypherpunk) remailer list.html (voir figure 5.2).

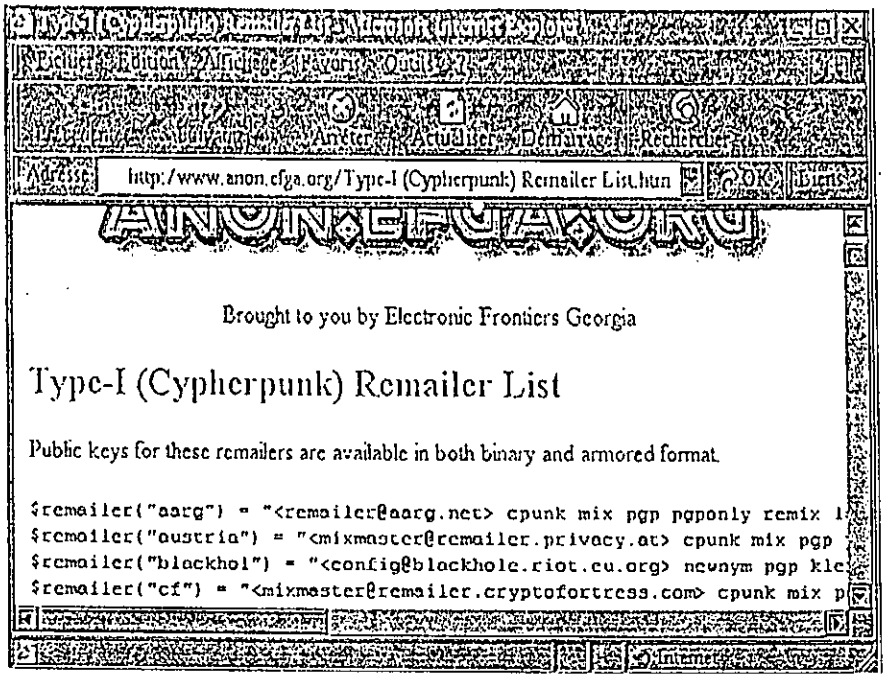


Figure 5.2 : la liste des remailers sur le site www.anon.efga.org

Cette liste contient 42 remailers avec leurs différentes caractéristiques et tests. De plus, vu que la plupart de ces remailers acceptent les messages cryptés par PGP (il y a même certains qui n'acceptent que les messages cryptés par PGP et rejettent les autres), il existe dans ces deux pages des liens qui nous permettent de télécharger directement le fichier pubring.pgp qui contient les clés privées de tous les remailers figurant dans les deux listes.

On remarque aussi que chaque remailer de la liste réalise beaucoup de fonctions, si on prend l'exemple : middleman@cyberpass.net, on trouvera à côté ses différentes fonctions : cpunk mix pgp hash middle latent cut ek reord . Telles que :

- **cpunk** : c'est à dire qu'il accepte tout message avec l'entête Request-Remailing-To: destination. Cette entête est acceptée par la majorité.
- **mix** : Il peut accepter les messages en format de mixmaster.
- **pgp** : il accepte les messages cryptés par PGP ; il accepte aussi les messages non cryptés.
- **hash** : Il supporte le symbole *##* par lequel toute information peut être ajoutée à l'entête du message telle que l'objet (par exemple : ''demande de renseignement'').
- **middle** : c'est à dire que ce remailer crée sa propre chaîne par d'autres remailers à travers lesquels le message va être envoyé. Il est conseillé d'utiliser ce type de remailer au milieu de la chaîne choisie initialement.
- **latent** : Il accepte l'entête Latent-time : option pour ajouter le temps de réponse.

- cut : Il accepte l'entête Cutmarks :option pour voir les modifications effectuées par le remailer sur le message envoyé.
- ek : Il peut crypter le message reçu avec la clé publique ajoutée avec l'entête Encryt-key : clé.
- reord : Réordonne les messages entrants et sortants pour compliquer les tentatives d'analyse de trafic.

Il existe cependant beaucoup d'autres fonctions dans la liste qui renforce l'anonymat de nos messages et, en utilisant ces remailers en chaîne, on va certainement rendre la traçabilité des chemins suivis par nos messages très difficile et même impossible ; plus le nombre de remailers utilisés en chaîne est élevé, plus notre anonymat est fort.

Pour plus d'informations concernant ces listes voir les annexes B et C.

4. Communications anonymes [12]:

L'Internet nous fournit la possibilité d'envoyer des messages Electroniques anonymes et ceci, grâce à la disponibilité des outils puissants de cryptographie. Les e-mails anonymes envoyés à travers plusieurs remailers ne peuvent pas être tracés si le chemin suivi est convenablement protégé. En plus, deux personnes ou plus peuvent communiquer sans savoir l'identité de chacun.

Un prospectus anonyme traditionnel demande une stratégie d'écriture et de distribution qui évitent de faire un lien entre le prospectus et l'auteur. Si le prospectus risque d'attirer l'attention de quelqu'un possédant des techniques modernes, il est soucieux d'éviter les marques telles que le papier distinctif ou les empreintes digitales. Contrairement, les communications sur Internet sont toutes numériques, les marques d'identification qu'elles transportent sont des informations insérées par l'expéditeur, le logiciel de l'expéditeur ou par des intermédiaires qui ont relié le message quand il est en transit. D'habitude, un e-mail, par exemple, arrive avec l'adresse de retour de l'expéditeur et l'information du routage décrivant le chemin pris à partir de l'expéditeur jusqu'au destinataire.

Il n'y a pas de raison pour laquelle le message révèle l'identité de l'expéditeur, c'est dans ce but que les remailers ont été conçus.

Les remailers varient mais tous les programmes de remailing (ré-acheminement) ont une caractéristique commune : tous les remailers suppriment les informations d'identification des

e-mails rentrants, substitue une entête prédéfinie identifiant le remailer comme étant un expéditeur ou utilisant une étiquette telle que anonymous@anonymous.

Par l'utilisation des outils cryptographiques disponibles et le routage du message à travers une série de remailers, l'utilisateur peut s'assurer de trois choses :

- Aucun remailer n'est capable de lire le texte du message parce qu'il est crypté plusieurs fois, la lecture du texte demande la participation de tous les remailers, en utilisant leurs historiques (fichiers log).
- Ni le destinataire ni autres remailers dans la chaîne (sauf le premier) ne peut identifier l'expéditeur du texte sans la coopération de tous les remailers précédents.
- Ainsi il est impossible (vue la complexité) pour le destinataire du message de connaître son expéditeur.

Pour la protection de la chaîne, nous devons avoir confiance à ces remailers. Si un seul remailer de la chaîne est honnête, toute la chaîne est sécurisée, et si les remailers se trouvent dans des pays différents, il est très coûteux de forcer les remailers à divulguer leurs historiques (fichiers log). En plus, il y a des remailers qui par principe refusent de garder les fichiers log, d'où il y a de fortes chances que les informations nécessaires n'existent pas.

5. Utilisation des remailers pour assurer l'anonymat [12]:

Si A veut envoyer un e-mail à un remailer avec l'instruction d'envoyer l'e-mail à B. Le remailer supprime l'adresse de retour identifiant A et envoie le message à B en prétendant qu'il est de la part de nobody@remailer.com. Si le remailer possède des fichiers log contenant les adresses mails de A et B, le chemin sera tracé, donc l'anonymat n'est pas assuré.

Pour garantir un grand niveau d'anonymat, on doit compliquer le chemin et ceci en se basant sur la technique d'onion routing.

Le technologie Internet actuelle permet le routage des messages à travers une série de remailers anonymes, cette technique est appelée "Chained remailing", si un seul remailer de la chaîne est honnête, A peut s'assurer que personne ne pourra lier A et B ceci par l'utilisation à la fois du cryptage et du chaînage. Cependant, on assume que l'utilisation de ces deux techniques n'est pas suffisante pour déjouer un espion qui est capable de tracer les messages entrants et sortants des différents remailers sur une période de temps. Pour cela, les remailers doivent compliquer l'analyse de trafic en utilisant la technique des Mixmasters précédemment

expliquée afin d'éviter d'éventuelles attaques de corrélation entre les messages entrants et les messages sortants. Le cryptage assure que le premier remailer ne peut pas lire et lier A à B.

Supposons que A décide d'envoyer un message anonyme à B à travers les remailers R_1, R_2, R_3. Chaque remailer possède une clé publique pour le cryptage par PGP que l'on a trouvé dans un fichier pubring.pgp disponible sur Internet. A veut s'assurer qu'aucun membre de la chaîne ne connaît le chemin complet (entièrement) des autres remailers manipulant le message. Chaque membre de la chaîne connaît l'identité du remailer précédent duquel le message est arrivé, et l'identité du prochain remailer auquel le message doit être envoyé.

Ainsi A a besoin de R_1, le premier membre de la chaîne qui élimine toutes informations reliant A au message. R_1 sait seulement que le message doit être envoyé à R_2 et ignore le chemin après cela. R_2 est le deuxième membre de la chaîne, il connaît seulement que le message vient de R_1 et doit être envoyé à R_3 puis, R_3 envoie enfin le message à B sans connaître son expéditeur.

Pour atteindre cet objectif, A doit crypter le message plusieurs fois pour avoir un onion (message dans le noyau couvert par plusieurs couches résultant de l'opération de cryptage), utilisant les clés publiques de R_1, R_2 et R_3. Chaque remailer recevant le message va le décrypter avec sa clé privée. Ceci va lui permettre d'avoir en clair uniquement la prochaine destination. L'onion qui va être envoyé à R_1 est illustré par la figure 5.3.

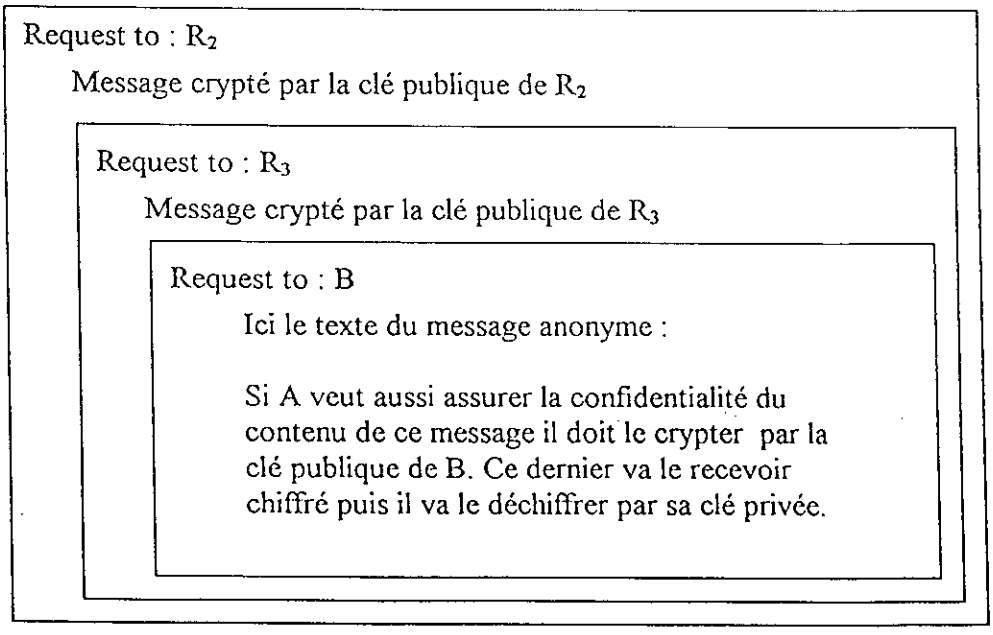


Figure 5.3 : Structure du message anonyme

6. Inconvénients :

L'inconvénient majeur des remailers est le temps de réponse comme on a pu le constater en consultant les deux pages web précédemment présentées. Bien qu'il existe des remailers qui répondent en moins de 5 minutes, il y en a d'autres qui répondent en plus de 2 jours. Ceci limitera donc le choix des remailers ainsi que le nombre de remailers dans une chaîne.

De plus, certains fournisseurs de ce service disent que leurs remailers ne gardent pas les fichiers log. Ceci reste à confirmer ! Néanmoins, il suffit qu'un seul remailer de la chaîne ne garde pas ces fichiers pour éviter la rupture de notre anonymat.

7. Conclusion :

Ainsi nous avons étudié dans ce chapitre un autre type de service Internet qui s'occupe de l'anonymat des courriers électroniques. Les remailers sont des serveurs qui ont été implémentés en se basant sur plusieurs techniques dans le but de compliquer les différents types d'attaques sur le trafic de données circulant sur un réseau. Nous avons vu que le point fort de ces derniers est de les utiliser en chaîne mais avec un nombre limité à cause de leurs temps de réponse.

Ce qu'on peut dire c'est que l'anonymat absolu n'existe jamais car il existe toujours des techniques sophistiquées pour rompre l'anonymat de certaines personnes qui utilisent ces services pour commettre des actes illégaux. Mais vu le coût de ces techniques, c'est seulement à l'échelle gouvernementale que l'on peut dépenser de l'argent pour mener des enquêtes profondes afin de détecter ces malfaiteurs. A part cela, personne ne trouvera son intérêt dans le gaspillage d'argent et de temps pour tracer le chemin des messages anonymes.

Néanmoins, reste le problème du fournisseur de ce service (L'ISP), puisque pour bénéficier de son service nous sommes obligés de révéler la destination de nos messages. La solution est de dissimuler la destination avant d'envoyer le messages au remailer. Ceci pourra être fait si nous pouvons prendre en main le choix de la chaîne de remailers utilisés pour l'envoi des messages au niveau de notre PC, dans ce cas nous pouvons construire notre propre onion (message + couches cryptographiques appropriées) puis l'envoyer à un remailer quelconque. Ainsi l'adresse de destination sera cachée dans le noyau du message et pour le remailer la destination sera le premier remailer choisi dans la chaîne initiale. Cette solution

fait l'objet de notre application qui sera présentée dans le prochain chapitre, elle a été nommée Chain telle que Chain est un programme qui permet l'envoi des messages anonymes sans prendre la peine d'utiliser un service Internet de ce genre, puisqu'il permet de choisir la chaîne des remailers par nous même, construire les couches cryptographiques appropriées et envoyer directement le résultat à un autre remailer au choix sans ouvrir notre boite de courrier électronique (voir chap.6).

Chapitre 6

Réalisation

1-Introduction :

Dans cette partie nous présentons la conception de notre application pour l'envoi des messages anonymes sur Internet en utilisant une chaîne de remailers comme il a été décrit au chapitre précédent.

2- Présentation générale du logiciel :

Notre logiciel est un simple client de messagerie qui nous permet d'établir une connexion directe avec le serveur de messagerie sortante auquel nous sommes inscrits, et d'envoyer le message vers la destination désirée sans faire appel à notre boite de courriers électronique.

En plus, il nous offre la possibilité d'assurer notre anonymat, en proposant une liste de remailers de laquelle nous devons choisir certains pour constituer une chaîne entre la source et la destination. Cette chaîne sera protégée par des couches imbriquées de cryptographie faites à l'aide du logiciel PGP.

Ce logiciel a été implémenté en langage Builder C++ version 5.0 et compilé sous Windows 98.

Après un simple clic sur l'icône de son exécutable 

, nous apercevrons l'affichage suivant :

Figure 6.1 : Affichage avant l'apparition de l'interface du logiciel

Puis, l'interface générale de notre application se présentera comme suit :

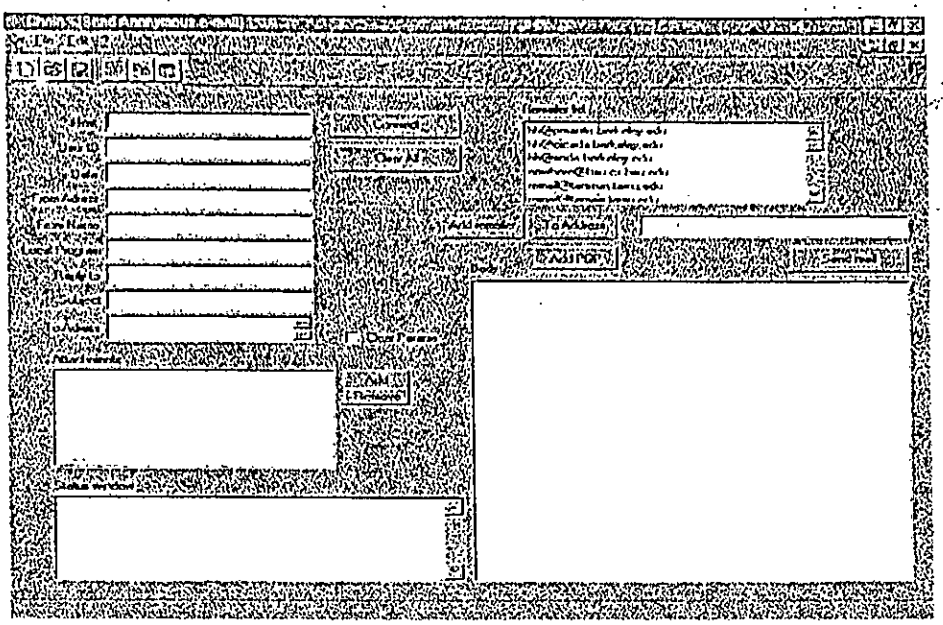


Figure 6.2 : L'interface graphique

Comme l'illustre la figure 6.2, l'interface de notre application comporte les éléments suivants :

- Une barre de menus.
- 6 boutons de raccourcis : 3 du menu File et 3 du menu Edit.
- 10 champs de saisies des paramètres nécessaires pour l'envoi des messages.
- Une liste box .
- Un menu optionnel obtenu par clic droit sur la liste box.
- Une zone de texte.
- Deux zones d'affichage de l'état de connexion et des noms de fichiers d'attachement.
- 8 boutons pour effectuer les différentes opérations.

2.1. La barre des menus :

La barre de menus de notre logiciel est composée de 3 boutons menus :

- File

- Edit
- ?

Figure 6.3 : La barre des menus

a) Le menu File :

Ce menu est composé de 4 commandes :

- New
- Open
- Save
- Exit

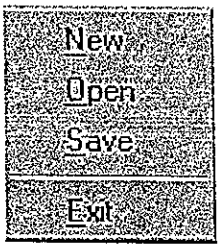


Figure 6.4 : Commandes du menu File

a.1 New :

Cette commande permet d'ouvrir une nouvelle page vide dans la zone de texte en effaçant tout ce qui a été écrit avant.

a.2 Open :

Cette commande permet d'importer n'importe quel texte écrit dans un fichier bloc note, Word pad ou Word 97 ou 2000, et le mettre dans la zone de texte de notre logiciel.
Il résulte de la commande Open, l'ouverture de la boite de dialogue suivante :

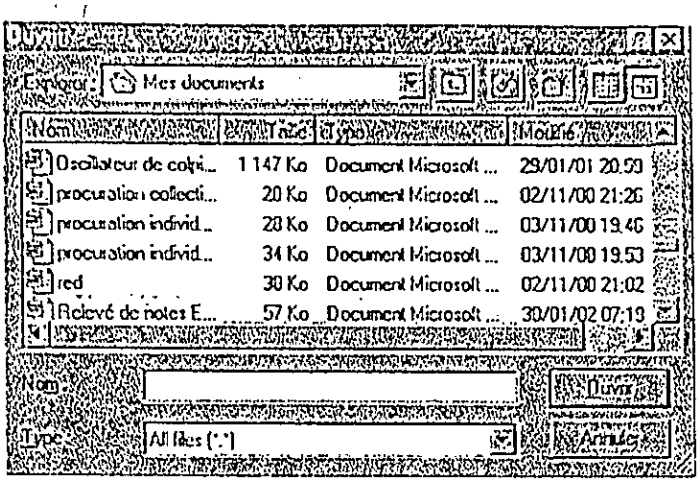


Figure 6.5 : La boite de dialogue Open

a.3 Save :

Cette commande permet d'enregistrer le contenu de la zone de texte dans un fichier sous un format texte.

a.4 Exit :

Elle permet de quitter l'application.

b) Le menu Edit :

Ce menu comporte 3 commandes :

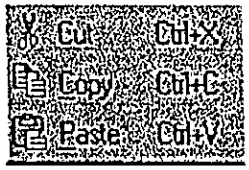


Figure 6.6 : Commandes du menu Edit

- La commande Cut : Pour effacer le texte des zones de saisie.
- La commande copy : pour copier le texte de n'importe quelle zone de saisie vers une autre.
- La commande Paste : Pour coller le texte copié dans le champ ou la zone voulue.

c) Le menu ?:

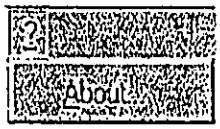


Figure 6.7 : Commande du menu ?

Ce menu contient une commande About qui affichera une fenêtre contenant des informations concernant le logiciel.

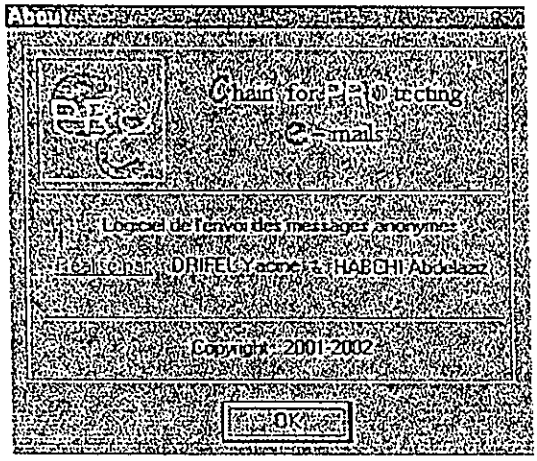


Figure 6.8 : La fenêtre About

La fenêtre se ferme par un simple clic sur le bouton OK.

2.2. Les boutons de raccourcis :

Figure 6.9 : Les boutons de raccourcis

Les trois premiers boutons sont des raccourcis pour les commandes New, Open et Save du menu File. Les trois autres sont des raccourcis pour les commandes Cut, Copy et Paste du menu Edit.

2.3. Les champs de saisie des paramètres d'envoi des messages :

Du côté gauche de l'interface apparaîssent 9 champs de saisie. Ils sont illustrés par la figure suivante :

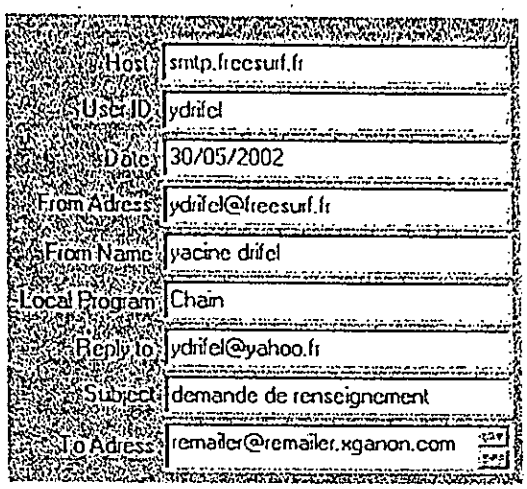


Figure 6.10 : Les champs des adresses et des paramètres d'envoi

Les champs Host, User ID, From Address et To Address sont obligatoires :

- Dans le champ **Host** on doit mettre l'adresse du serveur de messageries sortantes (le serveur SMTP) de l'hébergeur de notre boîte de courriers électroniques. Dans l'exemple de la figure 6.10 l'hébergeur de la boite ydrifel@freesurf.fr est www.freesurf.fr, et son serveur des e-mails sortants est smtp.freesurf.fr alors que le serveur des e-mails entrants est pop3.freesurf.fr.
- Dans le champ **User ID** on doit mettre le nom d'utilisateur de la boîte e-mail afin de s'identifier lors de la connexion au serveur. Dans l'exemple de la figure 6.10 on doit remplir le User ID par ydrifel qui est le nom d'utilisateur de la boite ydrifel@freesurf.fr.
- Dans le champ From Address on met notre adresse e-mail. Bien sûr il s'agit de l'adresse de la boite utilisée par l'application pour l'envoi du message. Dans l'exemple de la figure 6.10, il faut mettre ydrifel@freesurf.fr.
- Dans le champ **To address** on doit mettre l'adresse de destination du message envoyé. Cette adresse peut être celle d'un remailer si on veut envoyer un message anonyme, sinon une adresse personnelle dans le cas contraire.

Le reste des champs de saisie sont optionnels, ils sont souvent utilisés dans le cas de l'envoi non anonyme de nos messages :

- Dans le champ **Date** on peut mettre la date de l'envoi des messages.
- Dans le champ From Name on peut mentionner le nom complet de l'expéditeur du message.
- Dans le champ **Local Program** on peut informer le destinataire sur le nom du programme utilisé pour envoyer le message.
- Le champ **Replay To** est utilisé lorsque l'expéditeur demande une réponse du destinataire ; dans ce cas il doit mentionner l'adresse de la boîte e-mail où il veut recevoir la réponse.
- En fin le champ **Subject** qui signifie objet du message envoyé (exemple : Demande de renseignement).

Le dernier champ se trouve du côté droit de l'interface près du bouton To Address.

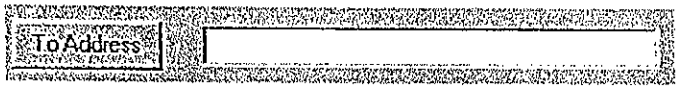


Figure 6.11 : Champ d'adresse anonyme

Ce champ est utilisé dans le cas de l'envoi de messages anonymes. Ici on doit mentionner l'adresse de destination, alors que dans l'autre champ To Address précédemment présenté on doit écrire l'adresse du premier remailer auquel on va envoyer le message traité.

2.4. La liste Box :

Elle contient une liste de remailers utilisés pour l'envoi des messages anonymes. Cette liste est divisée en deux groupes séparés par une ligne vide. Les remailers du premier groupe n'acceptent que les messages en clair car ils ne possèdent pas de clé publique PGP. Ils sont utilisés pour garantir l'anonymat de la source par rapport à la destination seulement. Par contre, les remailers du deuxième groupe possèdent leurs propres clé publique PGP, et donc ils sont utilisés pour assurer l'anonymat du chemin complet.

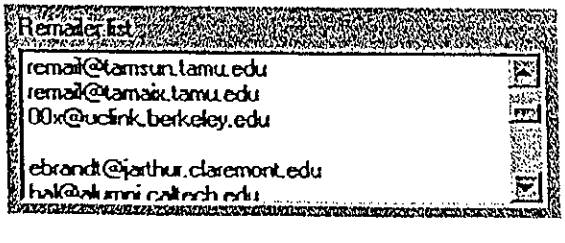


Figure 6.12 : La liste Box

2.5. Le menu Optionnel :

Il est obtenu par un clic droit sur la liste des remailers. Il comporte les commandes illustrées par la figure suivante :

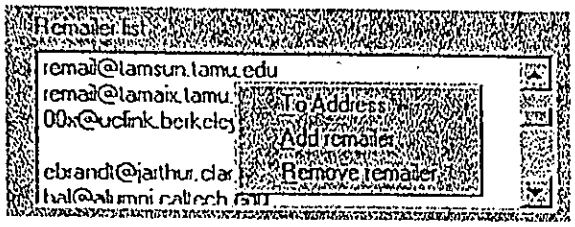


Figure 6.13 : Le menu optionnel

- La commande **To Address** permet de copier l'adresse du remailer choisi de la liste dans le champs To address. Si on choisit par exemple le remailer : remailer@xganon.com, après sa sélection depuis Remailer list , on clique sur la commande To Address.

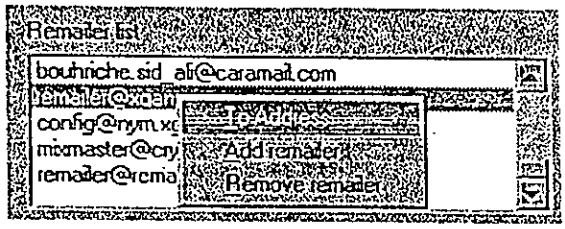


Figure 6.14: La commande To address

Ceci donnera le résultat suivant :

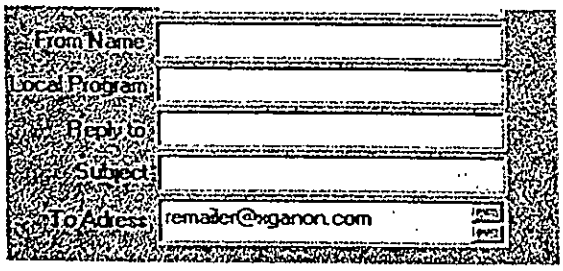


Figure 6.15 : Résultat de La commande To address

- La commande **Add Remailer** permet d'ajouter un nouveau remailer à la liste présente. Il suffit de cliquer sur la commande Add Remailer du menu optionnel, ceci va afficher une fenêtre dans la quelle on doit saisir l'adresse du nouveau remailer.

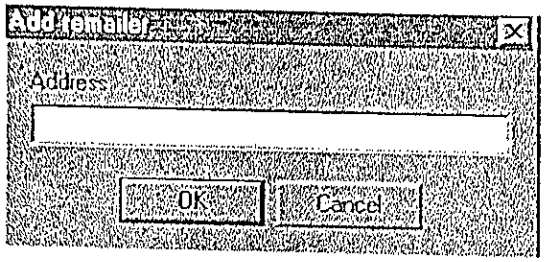


Figure 6.16 : Résultat de La commande Add Remailer

- La commande **Remove Remailer** permet de supprimer un remailer de la liste présente.

2.6. La zone de texte (Body):

C'est la zone dans laquelle le message va être écrit crypté et traité convenablement pour qu'on puisse l'envoyer.

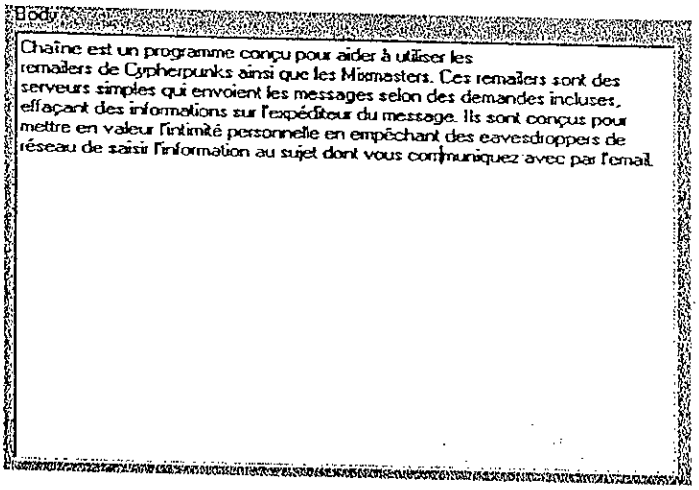


Figure 6.17 : La zone de texte

Le message écrit dans cette zone peut être crypté de l'extérieur par PGP et le résultat sera affiché dedans en écrasant le précédent.

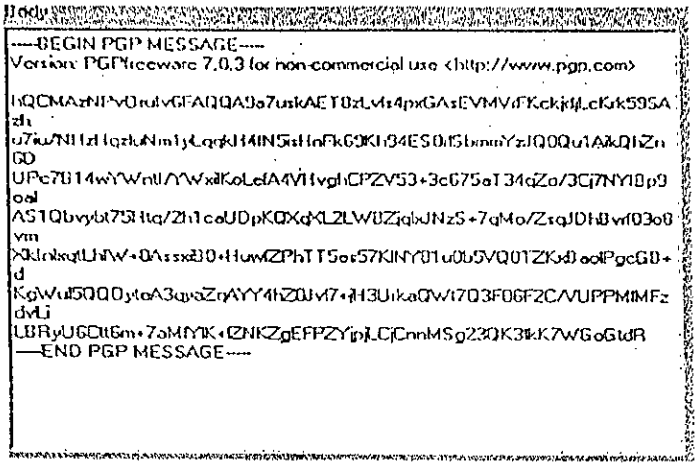


Figure 6.18 : Résultat de cryptage du message précédent par PGP

2.7. La zone "Attachments" :

Dans cette zone s'afficheront les noms et le chemin d'accès des fichiers d'attachement à envoyer avec le message. Elle est accompagnée par les deux boutons Add/Remove qui seront présentés dans les prochains paragraphes.

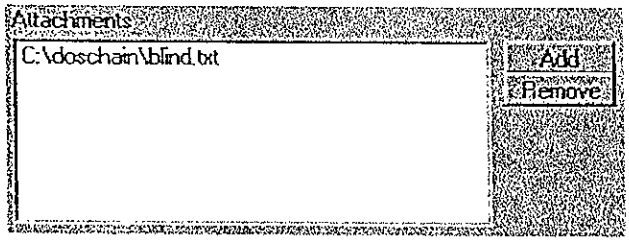
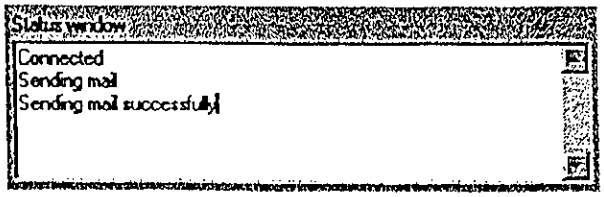


Figure 6.19 : La zone attachments

2.8. La zone "Status Window" :

Dans cette zone s'affichera les états de connexion entre ce programme client et le serveur de messagerie (connected, null remote address, ...etc), ainsi que l'état d'envoi des messages à la destination (Sending mail, Sending mail successfully, Recipient..... not found, ..etc).

Figure6.20 : La zone Status Window



2.9. Les boutons opérationnels:

a) Le bouton Connect : 

Un simple clic sur ce bouton permet de se connecter au serveur dont l'adresse est mentionnée dans le champ Host. Une fois connecté, le message "Connected" est affiché dans la zone.status window (voir figure 6.20).

b) Le bouton Sendmail :

L'utilisation de ce bouton permet l'envoi du message contenu dans la zone de texte vers le serveur des e-mail sortants qui va à son tour l'envoyer vers la destination dont l'adresse est écrite dans le champ to address. Bien sûr, cette opération s'effectue après l'opération de connexion. Lorsqu'on clique sur le bouton Sendmail, le message Sending mail s'affichera dans la zone status Window en-dessous du message connected. Ceci veut dire que l'opération de l'envoi est entrain de se dérouler. Une fois l'e-mail envoyé, dans la zone status Window on apercevra le message Sending mail successfully (Voir figure 6.20).

c) Le bouton Clear All :

Ce bouton permet d'effacer tous les champs de paramètres après avoir activé le champ Clear params 

d) Le bouton Add Remailer :

Ce bouton ajoute au dessus du corps du message à envoyer l'entête :

::

Request-Remailing-To: Adresse du remailer choisi.

Ceci va dire au remailer qui le précède dans la chaîne, que tout ce qui est en dessous de cette entête est à envoyer au remailer dont l'adresse est indiquée dans cette entête. Si on choisit l'exemple de mixmaster@cryptofortress.com, on aura le résulat suivant :

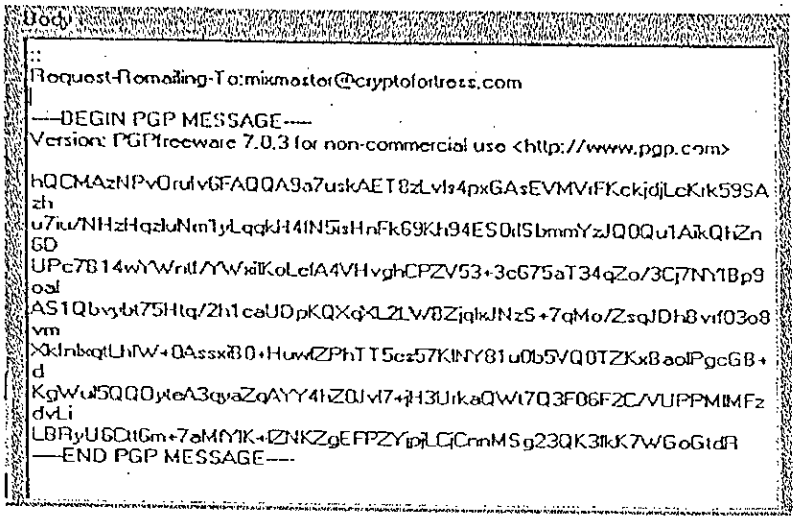


Figure 6.21 : Résultat de l'utilisation du bouton Add remailer

e) Le bouton Add PGP :

Ce bouton est utilisé pour ajouter au dessus du message crypté par PGP l'entête :

::

Encrypted : PGP

Ceci permet de dire au remailer recevant cette partie que celle-ci est cryptée par PGP. Le résultat est illustré par la figure 6.22.

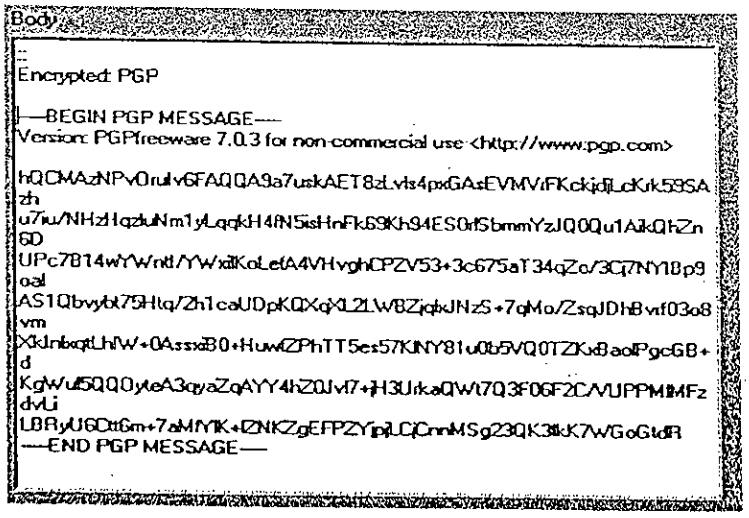


Figure 6.22 :Résultat de l'utilisation du bouton Add PGP

f) Le bouton To Address :

Ce bouton permet d'ajouter au corps du message l'entête :

: :

Request-Remailing-To: Adresse de la destination.

L'adresse de destination est lue à partir du champ d'adresse situé à côté de ce bouton.

C'est la première opération effectuée sur le message à envoyer. Cette entête sera lue par le dernier remailer de la chaîne.

Par exemple si la destination est habchi2002@yahoo.fr, On doit écrire cette dernière dans le champ approprié,

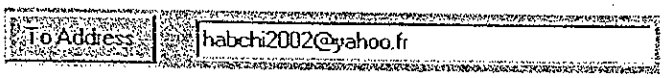


Figure 6.23 : Saisie de l'adresse de destination

puis on clique sur le bouton To address. Le résultat apparaît sur la figure 6.24.

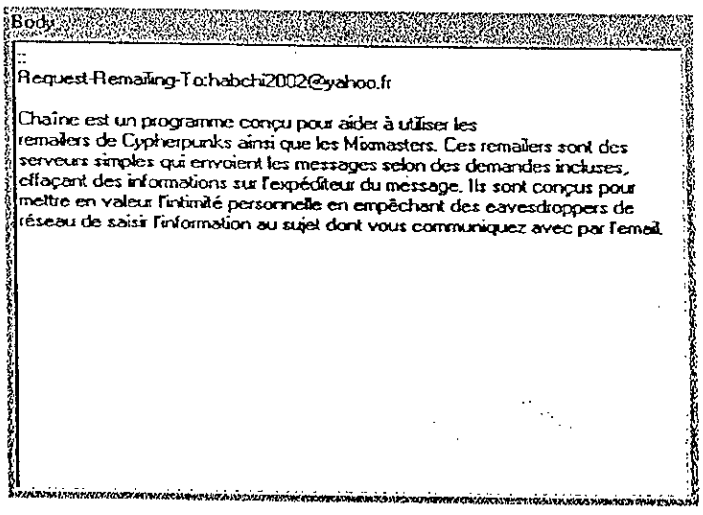


Figure 6.24 : Résultat de l'utilisation du bouton To address

g) Les boutons Add et Remove :

Ils accompagnent la zone "attachments", Le bouton Add permet d'ouvrir une boîte de dialogue pour choisir les fichiers d'attachement à envoyer avec le message. Une fois choisis, les noms et les chemins d'accès des fichiers d'attachement seront affichés dans la zone "attachments" .

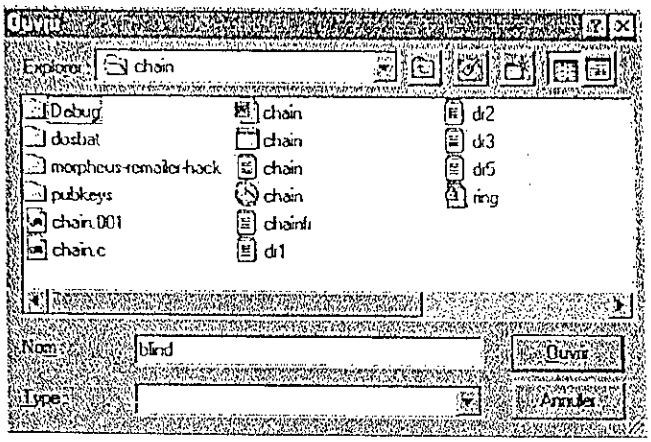


Figure 6.25 : Boîte de dialogue obtenue par utilisation du bouton Add

Alors que le bouton remove permet d'effacer toute la sélection et de vider le contenu de la zone "attachments".

3. Exemples de tests et résultats :

3.1. Exemple 1 :

- **Traitement avant l'envoi du message :**

Dans cet exemple nous allons envoyer un message d'essai à partir de la boîte e-mail ydrifel@freesurf.fr vers la destination habchi2002@yahoo.fr, en utilisant une chaîne constituée de deux remailers : il s'agit de remailer@remailer.xganon.com et anon@riot.eu.org dans l'ordre indiqué. Ceci veut dire que nous allons crypter deux fois ; une pour le "riot" et le tous pour le "remail.xganon". Pour cela nous devons tout d'abord remplir les champs de paramètres comme l'illustre la figure 6.26.

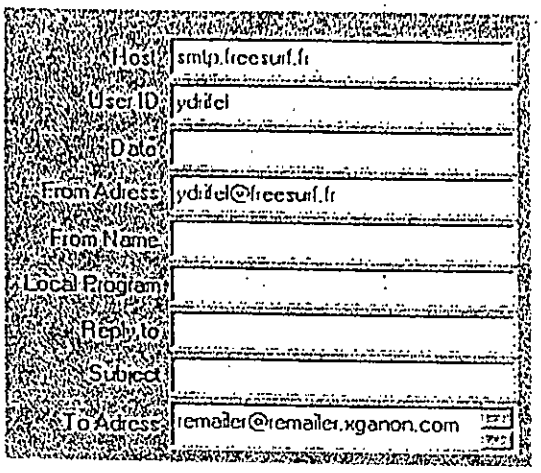


Figure 6.26 : Remplissage des champs de paramètres d'envoi

Puis, nous ouvrons le fichier texte dr2.txt où nous avons écrit le message.

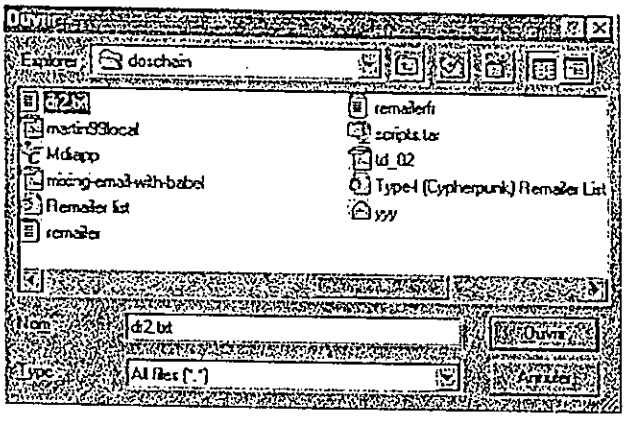


Figure 6.27 : Ouverture du fichier contenant le message à envoyer

Le résultat de l'ouverture s'affichera dans la zone de texte (body).

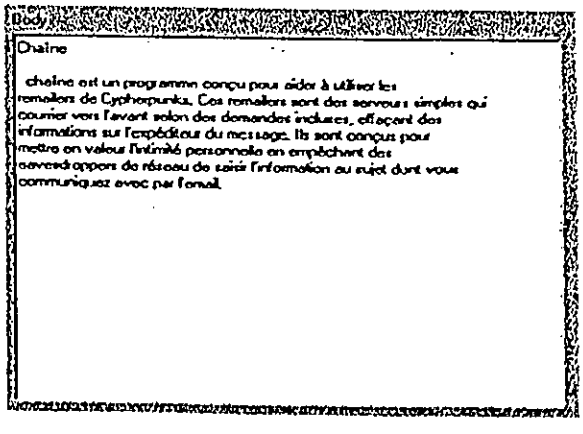


Figure 6.28 : Le message à envoyer

Après, nous devons saisir l'adresse de destination dans la barre d'adresses à côté du bouton To Address comme suit :

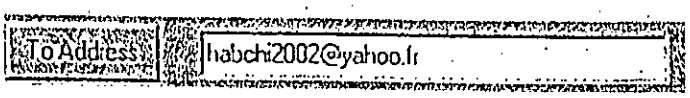


Figure 6.29 : Saisie de l'adresse de destination dans le champ To Address

Cette adresse sera ajoutée comme une entête au dessus du message après avoir cliqué sur le bouton To Address, comme le montre la figure 6.30 :

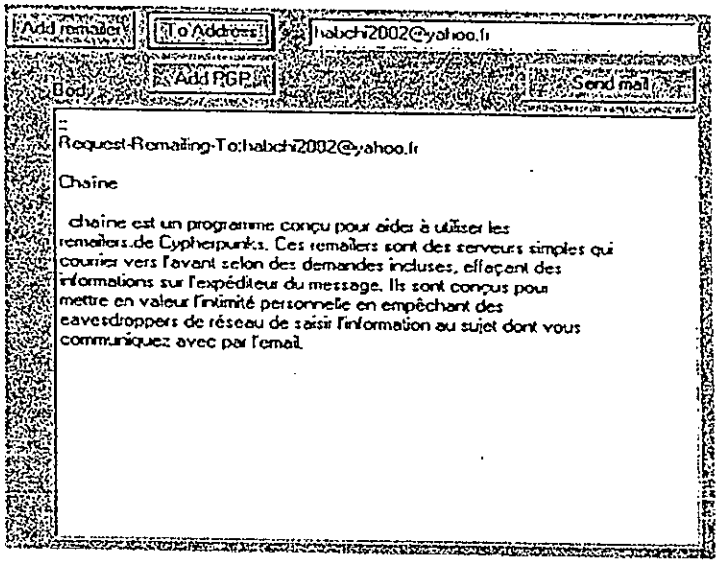


Figure 6.30 : L'entête du bouton To Address

Le résultat sera crypté pour le dernier remailer de la chaîne qui est anon@riot.eu.org. Pour cela nous executons la commande Encrypt à partir du menu Current Window du logiciel PGP, comme le montre la figure 6.31 :

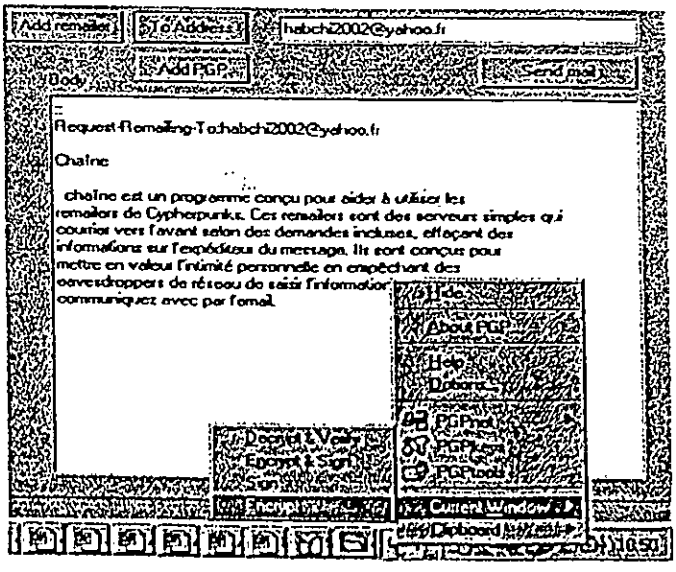


Figure 6.31 : Utilisation de PGP pour Crypter les messages

Une fenêtre PGPtray-Key s'affichera directement après l'exécution de cette commande pour pouvoir choisir la clé publique du remailer correspondant, par laquelle s'effectue la première opération de cryptage.

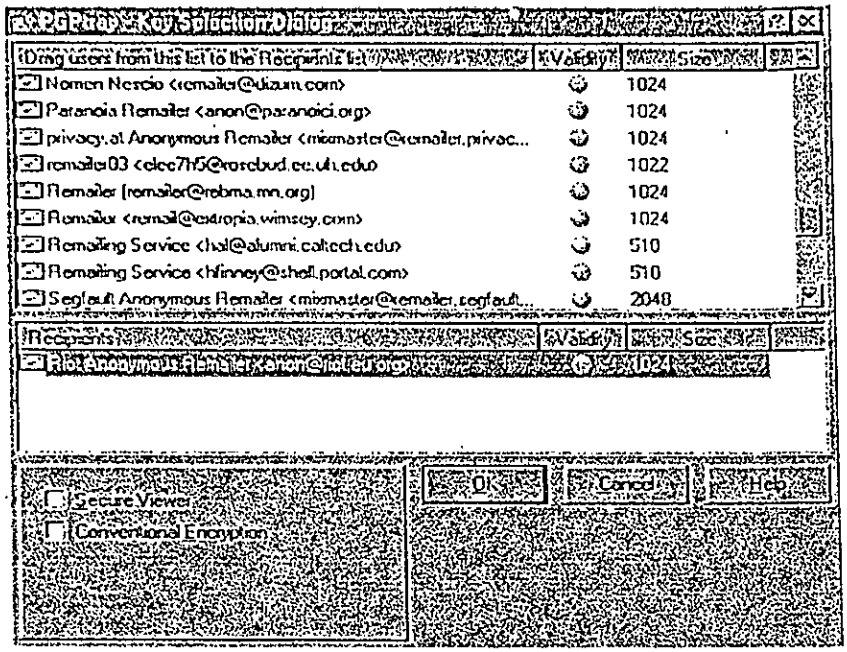


Figure 6.32 : La fenêtre PGPtray-key

Une fois terminé, le remailer "riot" doit savoir que le message reçu est crypté par PGP pour qu'il puisse le décrypter par sa clé privée. Pour cela, nous devons ajouter l'entête

::

Encrypted : PGP par un simple clic sur le bouton Add PGP.

Le résultat est illustré par la figure 6.33.

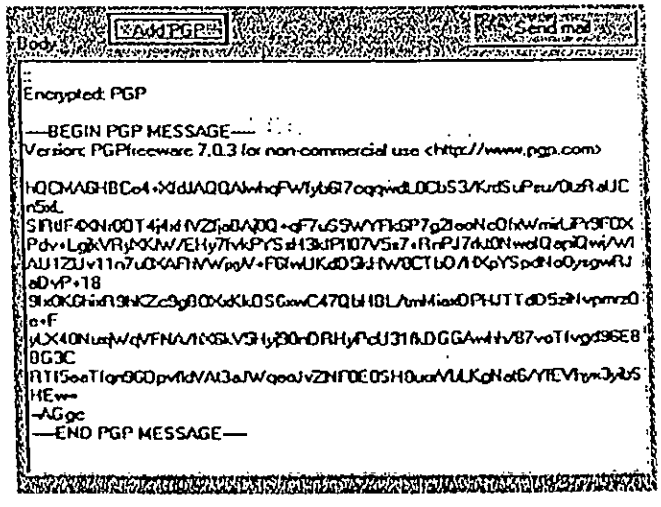


Figure 6.33 : Résultat final de la première opération de cryptage

Au résultat est ajoutée une entête contenant l'adresse du remailer riot, cette entête est utile pour le premier remailer de la chaîne car elle va lui montrer l'adresse du prochain remailer. L'entête est ajoutée après avoir sélectionné d'abord l'adresse du premier remailer à partir de la liste (anon@riot.eu.org) puis cliquer sur le bouton Add Remailer.

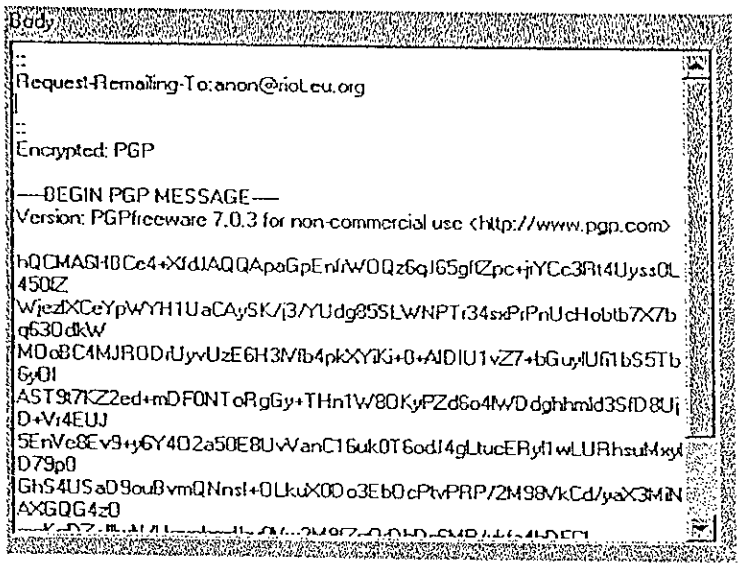


Figure 6.34 : Message devant être reçu par Remailer.xganon

La deuxième opération de cryptage s'effectuera de la même manière sur le résultat précédent, sauf que dans ce cas il faut choisir la clé publique du "remailer.xganon". Ainsi le message résultant est constitué d'un noyau qui est le message en clair plus deux couche de cryptographie.

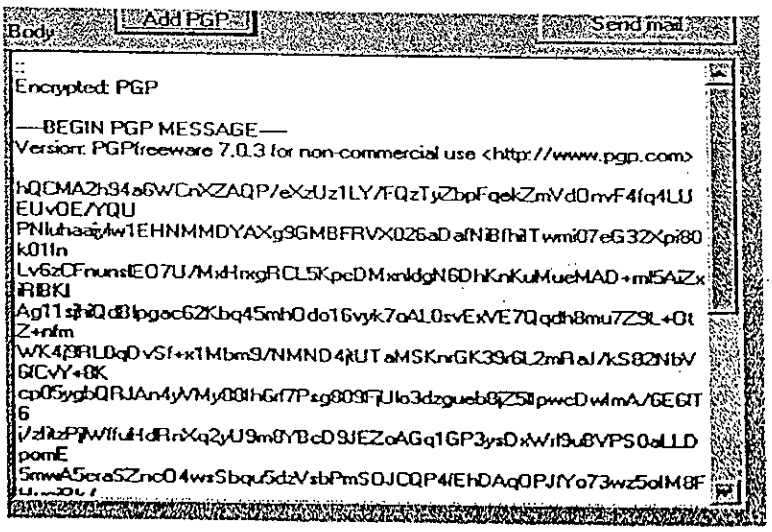


Figure 6.35 : Résultat final du traitement

Restera donc l'étape de l'envoi du résultat final. Pour cela, nous devons cliquer sur le bouton Connect pour établir une connexion avec notre serveur SMTP, puis nous attendons l'affichage du message "Connected" dans la zone status window. Une fois affiché, un simple clic sur le bouton Sendmail permettra l'envoi du message au premier remailer de la chaîne. Ceci est confirmé par l'affichage de la phrase "sending mail successfully" dans la zone status window comme nous l'avons déjà montré dans la figure 6.20.

- **Résultat :**

Lorsque le destinataire ouvre sa boîte e-mail (habchi2002@yahoo.fr), il apercevra l'arrivée d'un message anonyme tel que dans la case expéditeur il lira "anonymous user". Puis, lorsqu'il ouvre le lien correspondant il verra la page suivante :

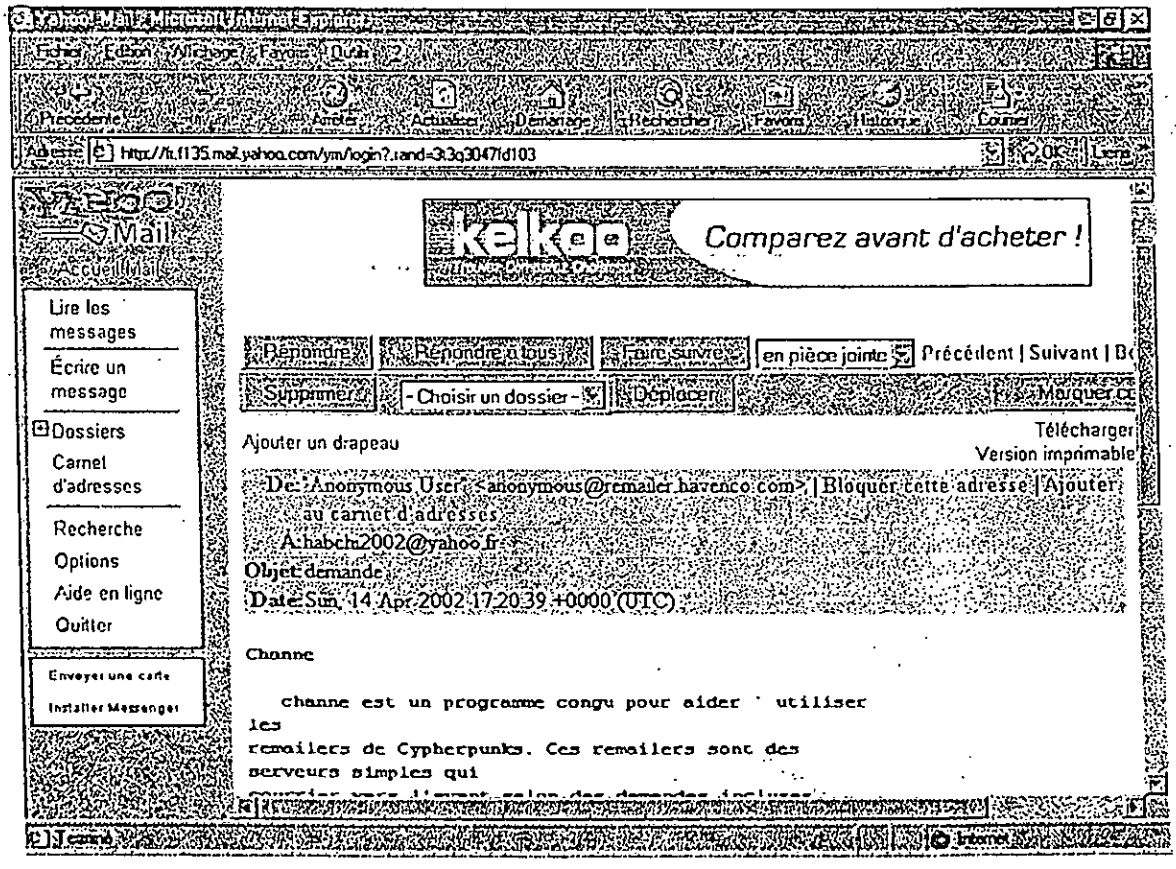


Figure 6.36 : Le message reçu par le destinataire

On remarque que dans le champ de l'expéditeur (De :.....), le destinataire voit qu'il a reçu ce message d'une personne anonyme, dont l'adresse est : anonymous@remailer.havenco.com. En fait, cette adresse est celle d'un remailer qui se trouve dans la liste précédemment présentée mais il n'est pas utilisé dans la chaîne initialement choisie pour l'envoi de ce message. D'où nous pouvons constater que le dernier remailer de la chaîne qui est "anon@riot.eu.org" possède la caractéristique middle qui veut dire que ce dernier envoie le message reçu à travers une autre chaîne de remailers dont le dernier est mix@remailer.havenco.com. Ce qui permet de renforcer notre anonymat. Cependant, le message reçu contient des erreurs de frappes à cause des multiples opérations de décryptage et d'envoi. Mais ceci est rarement rencontré puisque s'était la seule fois où nous avions reçu un message erroné parmi plusieurs essais réussis.

3.2. Exemple 2 :

Dans cet exemple nous allons envoyer le même message d'essai à travers trois remailers : il s'agit de :

remailer@remailer.xganon.com

mixmaster@cryptofortress.com

remailer@xganon.com

Dans l'ordre indiqué en haut nous devons appliquer les mêmes opérations vues dans l'exemple précédent, sauf que dans ce cas nous crypterons trois fois au lieu de deux fois ; puis nous enverrons le message résultant au premier remailer de la chaîne qui est remailer@remailer.xganon.com.

Une fois terminé, le message arrivera à la destination : tout d'abord le destinataire aperçoit que sa boite de réception contient un message non lu.

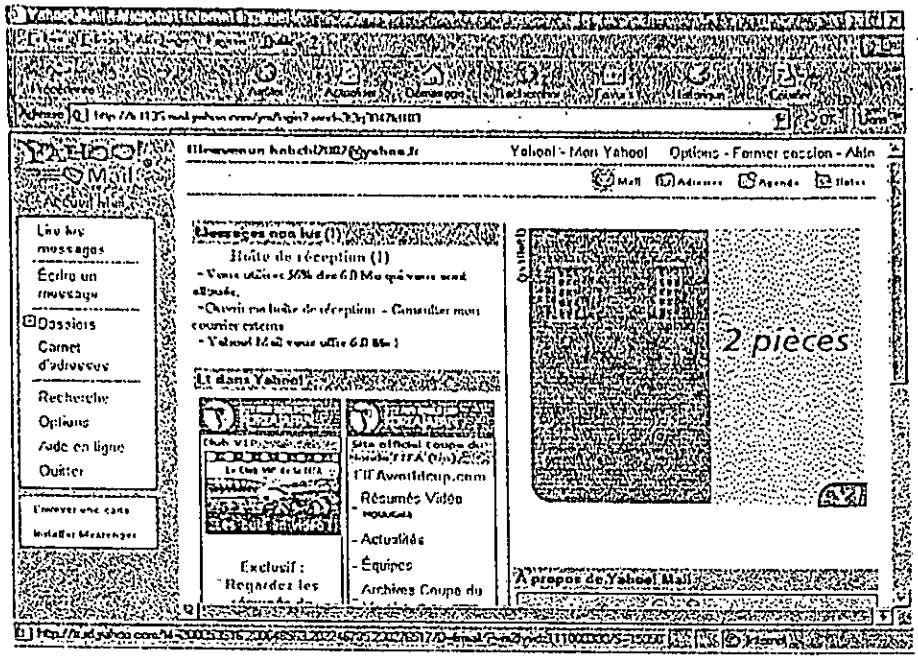


Figure 6.37 : Boite de réception du destinataire

Dans la boite de réception, le destinataire remarque que l'expéditeur du message reçu s'appelle xganon. Bien sûr il l'ignore du fait qu'il s'agit du nom d'un remailer.

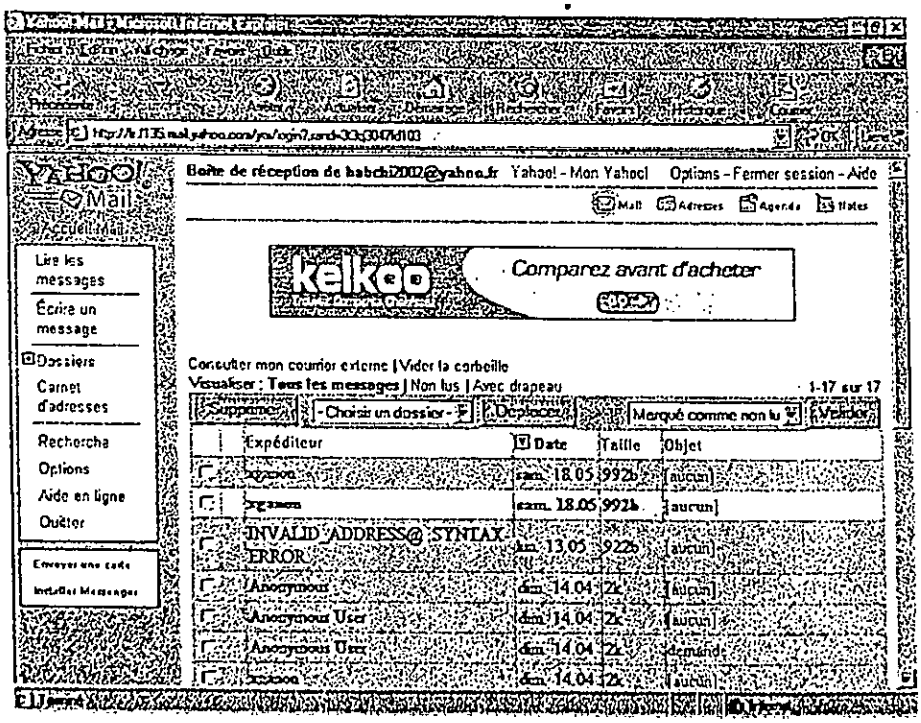


Figure 6.38 : Contenu de la boite de réception du destinataire

Après l'ouverture du message, il apercevra la page suivante :

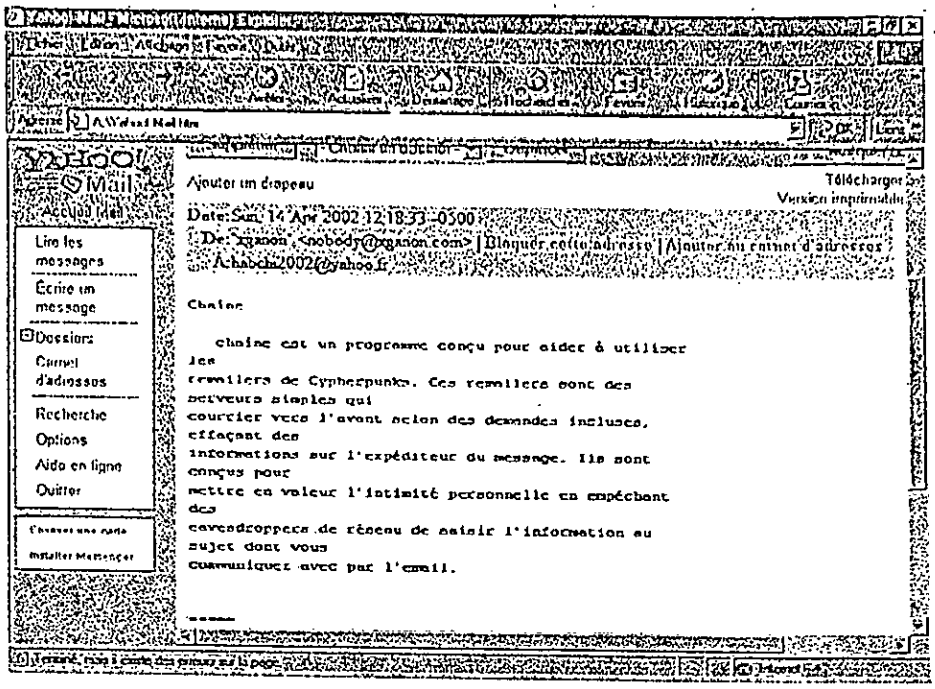


Figure 6.39 : Le message reçu par le destinataire

On remarque ici que le remailer xganon met son nom dans le champ du nom d'expéditeur contrairement à certains remailers qui mettent la phrase Anonymous user ou tout simplement Anounymous. On remarque aussi que le message est reçu sans erreur contrairement à l'exemple précédent.

Remarques :

- Les temps de réponse de chacune des chaînes utilisées dans les exemples 1 et 2, étaient inférieurs à 5 minutes, alors que dans d'autre chaîne nous avions reçu le résultat après 2 jours ce qui réduit le choix du nombre de remailers utilisés dans une chaîne.
- Nous avons utilisé pour le cryptage le PGP freeware version 7.0.3, cette dernière ne possède pas un exécutable direct mais un ensemble d'utilitaires accessibles depuis le menu PGP-tray. Ceci nous a empêché de l'appeler directement de l'intérieur du programme, nous étions obligé à chaque fois de l'appeler de l'extérieur. Néanmoins, cela n'empêche pas le bon fonctionnement du programme.

4. Conclusion :

Nous avons pu dans ce chapitre décrire les parties les plus importantes de la mise en œuvre de notre application. Nous avons pu aussi présenter quelques exemples de tests et leurs résultats. Il faut noter qu'il existe beaucoup d'autres tests qui n'ont pas été présentés. Ceci ne nous empêche pas d'avouer que nous avions trouvé beaucoup de difficultés relative aux mauvaises conditions de connexions.

Conclusion Générale

Conclusion générale :

L'envoi des messages anonymes ne devient plus un problème, grâce à l'existence à la fois des remailers et des outils cryptographiques développés qui permettent de dissimuler toute information reliant la source à la destination des messages.

A travers ce travail, nous avons pu réaliser une application permettant d'utiliser une chaîne de remailers librement choisis par l'expéditeur, avec un outil cryptographique pour l'envoi anonyme des messages sur Internet dont le but principal est d'empêcher toute tentative d'espionnage même par les fournisseurs de services.

Cependant, le problème d'anonymat se pose dans beaucoup d'autres services Internet en l'occurrence le world wide web, l'un des services les plus utilisés. En effet, en accèdant au disque dur de l'ordinateur d'un internaute, il est possible de trouver de nombreuses informations créées lors de sa connexion à partir du dossier historique, ou dans la liste des favoris, ou bien dans les cookies, petits programmes crées lors de la visite de certains sites web. Pour cela, diverses techniques ont été mises au point pour faire face à ce genre de problèmes. Donc plusieurs extensions à notre application sont envisageables à savoir : l'introduction d'une de ces techniques pour protéger la navigation anonyme sur le web.

Enfin, il est à noter que ce travail nous a permis d'approfondir nos connaissances théoriques et pratiques, spécialement dans le domaine de la sécurité informatique. Cela sans négliger les connaissances considérables acquises dans l'étude des différentes techniques d'anonymat.

Annexes

Le logiciel PGP (version 7.0.3)

1. La cryptographie avec PGP :

Le chiffrement fait appel à deux techniques, le cryptage symétrique (une seule clé pour chiffrer et déchiffrer) et le cryptage asymétrique qui utilise une clé publique non protégée qui chiffre mais ne déchiffre pas et une clé privée (jamais transmise et protégée) pour déchiffrer. PGP (Pretty Good Privacy) créé par l'américain Philip Zimmermann, combine les deux méthodes. Ce produit est disponible sur toutes les plates formes matérielles et les systèmes d'exploitations (windows, mac, unix, linux.....).

Il existe plusieurs utilitaires utilisant la technologie PGP :

- Le freeware comporte deux modules **PGPKeys** pour la gestion des clés et **PGPtools** pour le déchiffrement, le déchiffrement, la signature et l'effacement.

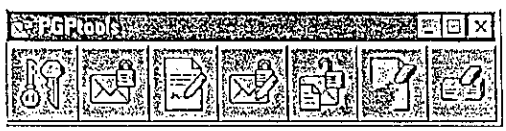


Figure A-1 : La fenêtre de PGPtools

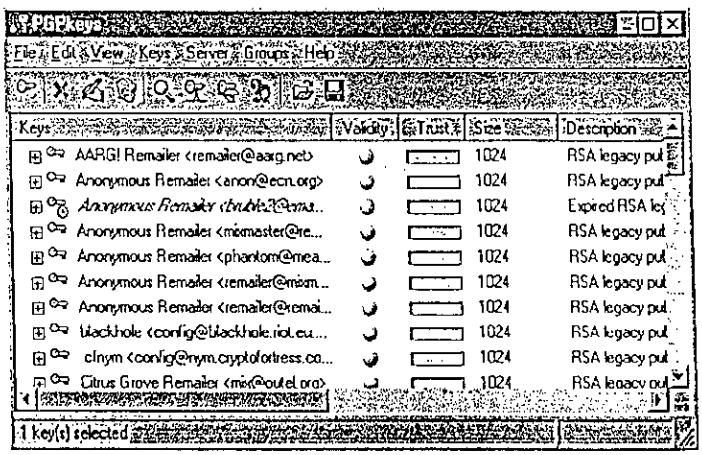


Figure A-2 : La fenêtre de PGPkeys

- **PGPdisk** sert au cryptage des données de tout ou partie d'un disque dur. Il n'est malheureusement plus disponible dans la version freeware, mais dans la version payante. Il permet de créer un volume PGPdisk qui, une fois monté permet d'être utilisé comme

disque dur externe. On peut y installer des applications et stocker des fichiers protégés par un mot de passe secret.

- **PGPnet** permet la mise en œuvre d'un réseau local dont les communications numériques sont sécurisées entre les postes où PGPnet est installé.

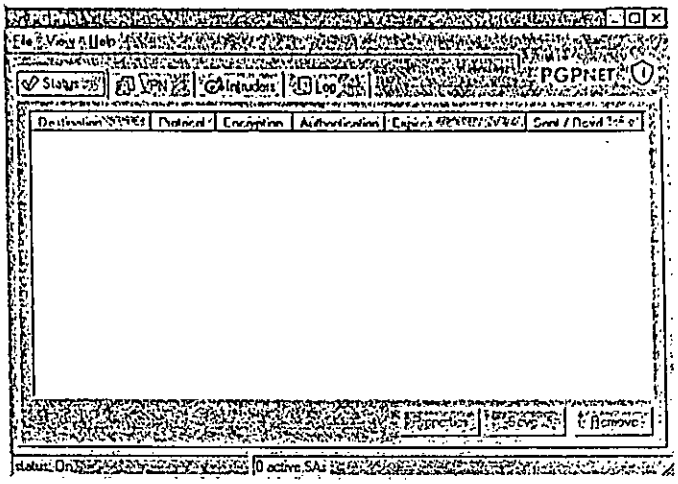


Figure A-3 : La fenêtre de PGPnet

- La version payante de *NAI* (Network Associates, la société qui distribue PGP) inclus des fonctions comme l'auto-décryptage qui permet de convertir des fichiers ou des dossiers en archive d'auto-décryptage pouvant être envoyés à des utilisateurs ne possédant pas PGP.

2. La procédure de cryptage PGP :

Avant de crypter un message, celui ci est comprimé par PGP pour limiter la durée de transmission et renforcer la sécurité cryptographique. Puis le programme crée une clé de cession aléatoire qui sert à crypter notre texte. Une fois générée, la clé de cession est codée avec la clé publique du destinataire du message. Ce destinataire va recevoir en plus du message chiffré, la clé de cession cryptée avec (sa propre clé publique).

3. La procédure de décryptage PGP :

Lors de la réception du message crypté, le destinataire se sert de sa clé privée pour récupérer la clé de cession aléatoire utilisée pour coder le message. Puisque l'expéditeur du message a créé la clé de cession (en fait la clé de cryptage) avec la clé publique du destinataire, ce dernier peut parfaitement la reconstituer à l'aide de sa clé privée car les clés

fonctionnent par paires. Une fois en possession de la clé de session aléatoire, il ne lui reste plus qu'à déchiffrer le message qui lui a été adressé.

4. Comment l'expéditeur peut-il entrer en possession de la clé publique du destinataire ?

La clé publique générée par PGPkey, peut être envoyée par e-mail, dans le corps du message ou en tant que pièce jointe, soit sur un serveur de clés publiques ou d'entreprise, qui stocke les clés publiques de nombreux utilisateurs de PGP. Il faut récupérer la clé publique de la personne à laquelle on veut envoyer le message chiffré. Pour cela il faut constituer un trousseau de clés publiques contenant les clés publiques authentifiées des interlocuteurs.

5. Comment être sûr que la clé publique est celle de son propriétaire supposé ?

Comparer l'empreinte numérique digitale (fingerprint) de la copie de la clé publique sur la machine avec celle de la clé d'origine. Cette empreinte est constituée d'une liste de mots générés aléatoirement par l'utilisateur et phonétiquement distincts pour être facilement compréhensible. En comparant cette liste de mots entre la copie de la clé et une disquette remise en main propre, il est possible d'authentifier le détenteur de la clé.

6. Comment peut-on être sûr de l'intégrité d'un document ?

PGP utilise un procédé d'authentification qui empêche la récupération de la signature d'un document pour la joindre à un autre document. Le message est transformé en un court résumé de 160 bits par une fonction de hachage. Toute modification du message transforme ce résumé qui est ensuite chiffré avec la clé privée de l'expéditeur. Le texte en clair et sa signature cryptée sont envoyés au destinataire du message. Ce dernier utilise la clé publique de l'expéditeur du message pour vérifier si la signature est correcte.

Une fois installée (PGP), la première étape consiste à créer la clé publique et la clé privée. On peut définir une date d'expiration de la clé publique (illimitée par défaut). Ensuite, saisissant la phrase clé complexe (passphrase) qui protège la clé privée ; elle sera utilisée à chaque utilisation de la clé privée, lors de l'envoi et de la réception des messages cryptés. La clé privée secrète est stockée dans le fichier **sering.pkr** , et le trousseau de clés publiques est stocké dans le fichier **pubring.pkr**

La liste des remailers obtenue de la page www.cs.berkeley.edu

Voici donc la liste complète des adresses remailers figurant dans la page web du site www.cs.berkeley.edu, sachant que cette page est mise à jour régulièrement dès qu'un nouveau remailer est mis au point.

- **La liste :**

$remailer{'cyber'} = '<alias@alias.cyberpass.net>

$remailer{"mix"} = "<mixmaster@remail.obscura.com>

$remailer{"replay"} = "<remailer@replay.com>

$remailer{"jam"} = "<remailer@cypherpunks.ca>

$remailer{"winsock"} = "<winsock@rigel.cyberpass.net>

$remailer{'nym'} = '<config@nym.alias.net>

$remailer{"squirrel"} = "<mix@squirrel.owl.de>

$remailer{'weasel'} = '<config@weasel.owl.de>

$remailer{"reno"} = "<middleman@cyberpass.net>

$remailer{"cracker"} = "<remailer@anon.efga.org>

$remailer{'redneck'} = '<config@anon.efga.org>

$remailer{"bureau42"} = "<remailer@bureau42.ml.org>

$remailer{"neva"} = "<remailer@neva.org>

$remailer{"lcs"} = "<mix@anon.lcs.mit.edu>

$remailer{"medusa"} = "<medusa@weasel.owl.de>

$remailer{"McCain"} = "<mccain@notatla.demon.co.uk>

$remailer{"valdeez"} = "<valdeez@juno.com>

$remailer{"arrid"} = "<arrid@juno.com>

$remailer{"hera"} = "<goddesshera@juno.com>

$remailer{"htuttle"} = "<h_tuttle@rigel.cyberpass.net>

On trouvera également dans cette page en plus des caractéristiques de chaque remailer qui sont nombreuses, une table résumant les temps de réponses de quelques remailers testés :

Remailer	Adresse e-mail	Historique	Temps de réponse	% de bon fonctionnement
hera	goddesshera@juno.com	----------	5:03:45	99.86%
nym	config@nym.alias.net	+*#**#**####	00:34	95.82%
redneck	config@anon.efga.org	#*###*+#****	02:00	95.44%
Mix	mixmaster@remail.obscura.com	+++ +++++++*	19:18	95.27%
squirrel	mix@squirrel.owl.de	-- ---+---	2:34:19	95.16%
cyber	alias@alias.cyberpass.net	*++***+ ++	11:26	95.11%
replay	remailer@replay.com	**** ***	10:06	94.93%
arrid	arrid@juno.com	----.------	8:50:34	94.41%
bureau42	remailer@bureau42.ml.org	---------	3:38:29	93.53%
cracker	remailer@anon.efga.org	+ +*+*+*+	16:32	92.80%
jam	remailer@cypherpunks.ca	+ +*-++++	24:14	92.79%
winsock	winsock@rigel.cyberpass.net	-..-..----	9:59:18	92.22%
neva	remailer@neva.org	------****+	1:03:02	90.39%

Les significations des symboles utilisés dans la case de l'historique sont données comme suit :

- # : réponse en moins de 5 minutes.
- * : réponse en moins d'une heure.
- + : réponse en moins de 4 heures.
- - : réponse en moins de 24 heures.
- . : réponse en moins de 2 jours.
- _ : réponse en plus de 2 jours.

La liste des remailers obtenue de la page www.anon.efga.org

Voici aussi la liste complète des adresses remailers figurant dans la page web du site www.anon.efga.org, sachant que cette page est mise à jour régulièrement dès qu'un nouveau remailer est mis au point.

- **La liste :**

$remailer{"aarg"} = "<remailer@aarg.net>
$remailer{"austria"} = "<mixmaster@remailer.privacy.at>
$remailer{"blackhol"} = "<config@blackhole.riot.eu.org>
$remailer{"cf"} = "<mixmaster@remailer.cryptofortress.com> $remailer{"cfnym"} = "<config@nym.cryptofortress.com>
$remailer{"citrus"} = "<mix@outel.org>
$remailer{"cmeclax"} = "<cmeclax@ixazon.dynip.com>
$remailer{"cracker"} = "<remailer@gacracker.org>
$remailer{"cripto"} = "<anon@ecn.org>
$remailer{"cthulu"} = "<mixmaster@cthulu.joatcrafts.org>
$remailer{"dingo"} = "<dingo1@dingoremailer.com>
$remailer{"dismix"} = "<mix@disastry.dhs.org>
$remailer{"dizum"} = "<remailer@dizum.com>
$remailer{"elvis"} = "<elvis@jpunix.com>
$remailer{"farout"} = "<farout@nuther-planet.net>
$remailer{"freaky"} = "<freaky@bigpond.net.au>
$remailer{"frog3"} = "<frog3remailer@frogadmin.yi.org> $remailer{"harmless"} = "<harmless@minder.net>
$remailer{"havenco"} = "<mix@remailer.havenco.com>
$remailer{"italy2"} = "<italyremailer@iol.it>
$remailer{"lcs"} = "<mix@anon.lcs.mit.edu>
$remailer{"lefarris"} = "<remailer@lefarris.dns2go.com> $remailer{"lemuria"} = "<mix@nox.lemuria.org>
$remailer{"matrix"} = "<matrix@underground.dnsalias.net>
$remailer{"notatla"} = "<mixclient@notatla.demon.co.uk> $remailer{"nullify"} = "remailer@mixmaster.nullify.org

$remailer{"nym"} = "<config@nym.alias.net>

$remailer{"paranoia"} = "<anon@paranoici.org>

$remailer{"passthru"} = "<mixer@immd1.informatik.uni-erlangen.de>

$remailer{"randseed"} = "randseed@melontraffickers.com

$remailer{"redneck"} = "<config@redneck.gacracker.org>

$remailer{"riot"} = "<anon@riot.eu.org>

$remailer{"rot26"} = "<rot26@mix.uucico.de>

$remailer{"segfault"} = "<mixmaster@remailer.segfault.net>

$remailer{"senshi"} = "<senshiremailer@gmx.de>

$remailer{"shinn"} = "<remailer@freedom.gmsociety.org>

$remailer{"squirrel"} = "<mix@squirrel.owl.de>

$remailer{"tonga"} = "<remailer@cypherpunks.to>

$remailer{"xganon"} = "<remailer@xganon.com>

$remailer{"xganon2"} = "<remailer@remailer.xganon.com>

$remailer{"xgnym2"} = "<config@nym.xganon.com>

$remailer{"xgmail"} = "<config@mail.xganon.org>

On trouvera également dans cette page en plus des caractéristiques de chaque remailer qui sont nombreuses, une table résumant les temps de réponses de quelques remailers testés :

Remailer	Adresse e-mail	Historique	Temps de réponse	% de bon fonctionnement
cripto	anon@ecn.org	++--++++*+++	27:16	100.00%
xgmail	config@mail.xganon.org	###########	00:23	100.00%
redneck	config@redneck.gacracker.org	###########	00:35	99.99%
xganon2	remailer@remailer.xganon.com	###########	00:18	99.99%
segfault	mixmaster@remailer.segfault.net	++++*++*++**	14:22	99.99%
cracker	remailer@gacracker.org	*++++++++++	24:22	99.99%
havenco	mix@remailer.havenco.com	*-**********	10:18	99.99%
xgnym2	config@nym.xganon.com	###########	00:20	99.99%
blackhol	config@blackhole.riot.eu.org	#####*#+######	02:05	99.99%
austria	mixmaster@remailer.privacy.at	+-*****+*****	09:57	99.98%
paranoia	anon@paranoici.org	***+****+**++	10:24	99.98%

dismix	mix@disastry.dhs.org	+*+*+*+++++++*	32:48	99.97%
cmeclax	cmeclax@ixazon.dynip.com	---+++----+-	1:14:15	99.97%
dingo	dingo1@dingoremailer.com	+++*++++++++**	22:08	99.96%
farout	farout@nuther-planet.net	----------	8:28:31	99.88%
nullify	remailer@mixmaster.nullify.org	++ -+++++++++	20:35	99.86%
dizum	remailer@dizum.com	+**___.-****	9:20:45	99.86%
riot	anon@riot.eu.org	---.__-----	7:32:09	99.83%
randseed	randseed@melontraffickers.com	*+******* ***	07:18	99.74%
xganon	remailer@xganon.com	*+**********	08:06	99.71%
aarg	remailer@aarg.net	+ **********	07:52	99.60%
nym	config@nym.alias.net	# *######## **	04:02	99.60%
notatla	mixclient@notatla.demon.co.uk	--..-------	9:38:08	99.50%
squirrel	mix@squirrel.owl.de	--------- -	3:51:42	98.93%
shinn	remailer@freedom.gmsociety.org	+ ++++**+*+*	23:24	98.88%
citrus	mix@outel.org	+ + ++ +++	37:17	98.60%
cfnym	config@nym.cryptofortress.com	# *### #####	00:17	98.54%
harmless	harmless@minder.net	+ -+__.-**	1:35:17	98.37%
matrix	matrix@underground.dnsalias.net	_.---_.-*+	1:38:18	98.08%
cthulu	mixmaster@cthulu.joatcrafts.org	+ ++++ +++	47:12	96.31%
senshi	senshiremailer@gmx.de	_ ---_----	3:50:19	96.12%
lemuria	mix@nox.lemuria.org	+*+++**** +	13.32	94.16%
italy2	italyremailer@iol.it	_ ---	4:11:66	92.48%
frog3	frog3remailer@frogadmin.yi.org	_.--+*+***	2:08:04	90.55%

Les significations des symboles utilisés dans la case de l'historique sont les mêmes que celles données dans l'annexe B.

Liste des serveurs POP et SMTP de certains fournisseurs du service mail sur Internet.

Le site	Le serveur POP	Le serveur SMTP
www.9online.fr	pop.9online.fr	smtp.9online.fr
www.club-internet.fr	pop3.club-internet.fr	mail.club-internet.fr
www.francenet.fr	mail.francenet.fr	mail.francenet.fr
www.free.fr	pop.free.fr	smtp.free.fr
www.freesbee.fr	pop.freesbee.fr	smtp.freesbee.fr
www.freesurf.fr	pop.freesurf.fr	smtp.freesurf.fr
www.infonie.fr	pop.infonie.fr	smtp.infonie.fr
www.libertysurf.fr	pop.libertysurf.fr	smtp.libertysurf.fr
www.magic.fr	pop2.magic.fr	smtp2.magic.fr
www.m6net.fr	pop.m6net.fr	mail.m6net.fr
www.netclic.fr	pop.netclic.fr	mail.netclic.fr
www.net-up.com	pop.net-up.com	mail.net-up.com
www.oreka.fr	mail.oreka.fr	mail.oreka.fr
www.sympatico.ca	pop1.sympatico.ca	smtp1.sympatico.ca
www.teaser.fr	pop.teaser.fr	smtp.teaser.fr
www.wanadoo.fr	pop.wanadoo.fr	smtp.wanadoo.fr
www.worldonline.fr	pop3.worldonline.fr	smtp:worldonline.fr
www.worldnet.fr	pop.worldnet.fr	smtp.worldnet:fr

Bibliographie

[1] Christophe Bidan, Valéric Issarny : Un aperçu sur des problèmes de sécurité dans les systèmes Informatiques, IRISA, publication interne N° 959 Octobre 1995.

[2] ISO : Systèmes de traitement de l'information — Interconnexions de systèmes ouverts — Modèle de référence de base, partie 2 : Architecture de sécurité, ISO7498-2 :1989 (F)

[3] Tomas Olovsson : A structured approch to computer security, Department of computer Engineering, Chalmers University of technology, technical report N°122, 1992.

[4] L.Khelladi, K.Bouguessa : Conception et réalisation d'un système de vote électronique en Java, mémoire de PFE N°71/2001 USTHB 2001.

[5] B.Schnenier : Cryptographie appliquée, traduction de S.Vaudenay, International Thomson Publishing Company, Paris 1997.

[6] D.Goldschlag, M.Reed, and P.Syverson: Privacy on the Internet, INET97, Kuala Lumpur, Indinisia, June, 1997.

[7] Mari Korkea-Aho: Anonymity and Privacy in the Electronic World, Department of computer science, Helsinky University of Technology, Technical report November 1999.

[8] O.Fouache: Privacy Protocols, DEA Réseaux et systèmes distributes, Institut Eurécom, Juin 2001.

[9] D.Goldschlag, M.Reed, and P.Syverson: Proxies for anonymous routing, in Proc.12th An. Computer Security Applications Conf, San Diego, CA, 1996, pp. 95-104.

[10] D.Goldschlag, M.Reed, and P.Syverson: Hiding Routing Information, New York, Springer-Verlag, pp. 1996, 137-150.

[11] D.Goldschlag, M.Reed, and P.Syverson: Anonymous Connection And Onion Routing, IEEE Journal on Selected Area In Communications. Vol 16 N°04, May 1998.

[12] D.Goldschlag, M.Reed, and P.Syverson: Private Web Brosing, Journal of Computer

[13] A.Michael Froomkin: Flood Control on the Information Ocean: Living With Anonymity, Digital Cash, and Distributed Databases. Published at 15 U. Pittsburgh Journal of law and commerce 395, 1996.

[14] The Onion Routing home page. URL: http://www.onion-router.net/

[15] Proxymate. URL: http://www.proxymate.com/

[16] The Anonymazer. URL: http://www.anonymazer.com/

[17] D.Chaum: Untracable Electronic Mail, Return Addresses, and Digital Pseudonyms, Communications of the ACM, v.24, N°02, Feb.1981, pp.84-88.

[18] Les dossiers pratiques de www.anonymay.org: Comment être anonyme sur le Web ? Edition 2 du 01 Septembre 2000.