La Messagerie Electronique Sécurisée

O. Nouali(1), N. Taboudjemat(2), Berbar, R. Saadi(1)
(1): Laboratoire des Logiciels de base,
(2) : Laboratoire Réseau et Systèmes distribués,
CE.R.I.S.T,
Rue des 3 frères Aïssiou, Ben Aknoun, Alger, Algérie
E-mail : (nnouali, onouali)@mail.cerist.dz

Introduction

De nos jours le monde s'est transformé en un petit village grâce à Internet qui véhicule à chaque seconde des milliers d'informations sous formes d'échanges de données, de transactions bancaires ou bien de messages de tout type qui sont transmis par les serveurs de messagerie. Les services de la messagerie électronique sont les services les plus utilisés sur le Net, ce qui fait d'eux la cible privilégiée des attaquants qui n'économisent aucun moyens pour arriver à leurs fins.
Les différentes menaces qui pèsent sur la messagerie électronique peuvent être intentionnelles ou accidentelles [SAH 98]. Les menaces accidentelles sont dues à une mauvaise manipulation du système par les utilisateurs. Par exemple, envoyer un message confidentiel à une mauvaise personne par erreur. Par contre, les menaces intentionnelles sont d'origine malfaisante. Elles peuvent être passives (telle que l'écoute) ou actives (telles que la modification du contenu du message ou la falsification des données relatives à l'expéditeur du message) . Les différentes attaques conduisent à une atteinte à la confidentialité, l'intégrité, et l'authenticité des échanges [MEI 99].

La cryptographie est l'un des mécanismes de base utilisés pour répondre à ses besoins de sécurité et d'ailleurs intervient dans plusieurs aspect de la sécurité d'Internet. Elle représente l'espoir de la sécurité des réseaux informatiques. Il s'agit d'un ensemble de techniques qui permettent de transformer un texte en clair en un texte inintelligible. Seul le destinataire autorisé pourra retrouver la forme d'origine du texte.

Cet article décrit une solution cryptographique pour sécuriser les messages électroniques et plus précisément les protéger contre les attaques qui peuvent se produire durant leur transmission sur le réseau, en offrant trois niveaux de sécurité (basse, moyenne et haute sécurité). Elle implémente un ensemble d'algorithmes cryptographiques très sûrs, efficaces et connus. De plus, pour renforcer la sécurité des algorithmes utilisés, nous avons apporté des modifications sur les modes de chiffrement.

I. Les outils cryptographiques

La cryptographie est la science de sécurisation des données utilisant des algorithmes cryptographiques. Elle permet de stocker ou de transmettre des informations sensibles à travers un réseau non sûr (non sécurisé). Ces algorithmes cryptographiques sont des

fonctions mathématiques utilisées dans des processus de chiffrement et de déchiffrement avec la collaboration d'une clé. Le chiffrement étant le processus de transformation d'un message clair en un message incompréhensible (chiffré) et le déchiffrement étant le processus inverse [ZIM 98].

I.1 Algorithmes symétriques et algorithmes asymétriques

Il existe principalement deux grandes classes d'algorithmes cryptographiques : la cryptographie symétrique ou à clé secrète, et la cryptographie asymétrique ou à clé publique.

La cryptographie à clé secrète utilise pour le chiffrement et le déchiffrement la même clé d'où le nom de cryptographie symétrique [SCH 97]. Les algorithmes les plus utilisés sont DES (Data Encryption Standard), TWOFISH, BLOWFISH, etc. [FRE 99, KLE 00, IBR 98]. Ces algorithmes utilisent la notion de chiffrement par bloc qui consiste à chiffrer un texte en le découpant en blocs et en appliquant la technique bloc par bloc selon un mode de chaînage de ces blocs. En effet, plusieurs modes ont été élaborés dont le mode ECB (Electronique Code Book) qui consiste à chiffrer chaque bloc indépendamment des autres, ou le mode CBC (Chaining bloc cipher) qui consiste à chaîner le dernier bloc chiffré avec le bloc clair en cour qui sera chiffré à son tour, et ainsi de suite [KLE 00, SAH 98].

Les algorithmes symétriques basés sur les opérations simples (arithmétiques et permutations) sont réputés pour être très rapides et très sûrs, mais présentent des problèmes de gestion et de distribution des clés. En effet, pour n utilisateurs il faut générer $(n^2-n)/2$ clés et les échanger de manière sûre et secrète sur le réseau. C'est pour cela que le concept d'algorithmes asymétriques a été crée. Il s'agit d'utiliser une paire de clés, une clé publique (appelée aini parce qu'elle est rendue publique) pour crypter et une autre clé secrète que seul le propriétaire détient pour décrypter [KLE 00]. Aucune clé ne peut être déduite de l'autre. Les algorithmes les plus utilisés sont RSA (Rivest Shamir Adleman), Diffie-Helman, etc…[SSH 00]. Ces algorithmes sont basés sur des théories mathématiques très complexes tels que le problème de factorisation d'entiers, le logarithme discret, le concept de courbes elliptiques, etc… Ils utilisent de très grandes clés pour assurer une certaine sécurité mais ils sont très lents pour le chiffrement (pour le même texte, il faut 100 fois le temps nécessaire pour un chiffrement symétrique). En pratique, une solution cryptographique consiste à combiner entre les deux méthodes de manière à profiter des avantages de l'une et de l'autre.

I.2 Les certificats numériques

Dans un environnement à clé publique il est vital, de s'assurer que les clés utilisées ne sont pas des contrefaçons. L'utilisateur peut se limiter à n'utiliser que les clés qui lui ont été remises physiquement par leur propriétaire. Mais ceci n'est pas une solution pratique(comment échanger des données avec des gens qu'on ne connaît même pas ? Ou

qui sont à des milliers de kilomètres ?). Le concept d'autorité de certification(AC) a été introduit pour résoudre ce problème. Il s'agit d'un organisme ou personne de confiance chargée de délivrer des certificats numériques qui simplifient la tâche d'établir la réelle appartenance d'une clé à son propriétaire supposé. Un certificat numérique est une information attachée à une clé publique permettant de vérifier l'authenticité de cette dernière [ZIM 98].

I.3 Les Signatures numériques

La méthode de base utilisée pour créer une signature numérique consiste à chiffrer l'information avec la clé privée de l'expéditeur. Si le destinataire arrive à la déchiffrer avec la clé publique de l'expéditeur alors l'information est authentique [ZIM98].

Le système décrit ici est très lent, de plus il produit un volume énorme de données. Il double au minimum la taille de l'information originale. En pratique donc, on ne chiffre pas un message avec sa clé privée, mais on signe une empreinte du message, calculée par une fonction de hachage à sens unique (qui permet de réduire la taille du texte à chiffrer). C'est beaucoup plus rapide et tout aussi fiable [ZIM98].

I.3 Fonction de hachage

Une fonction de hachage à sens unique porte plusieurs noms : fonction de compression, fonction de contraction, digeste, empreinte digitale, code correcteur cryptographique, code de vérification d'intégrité. Une fonction de hachage est une fonction mathématique qui convertit une chaîne de caractères de longueur quelconque en une chaîne de caractères de taille fixe (souvent de taille inférieure, cette chaîne est appelée empreinte).

Il est aisé de calculer l'empreinte à partir de la chaîne d'entrée mais il est presque impossible d'engendrer des chaînes qui ont une certaine empreinte tout comme il est mathématiquement impossible de remonter à la chaîne d'origine à partir de l'empreinte. Pour toutes fonctions de hachage la probabilité d'une collision est estimée à 2^n où n est la taille de la clé [KLE00].

Elle assure l'intégrité de telle manière que si un seul bit change, la sortie sera différente (avec une probabilité de collision égale à 2^n). Tant qu'une fonction de hachage sûre est utilisée, il n'y a aucun moyen de recopier la signature d'une personne ou d'altérer en quoi que ce soit un document signé. Le moindre changement provoquera l'échec de la vérification de la signature [ZIM98].

Parmi les algorithmes les plus connus, on cite SHA (160 bits) et MD5 (128 bits) [ROB94].

II. Architecture et fonctionnement du système

Le système réalisé est composé de cinq (05) modules comme le montre la figure1 :

- Un module de contrôle d'accès qui permet de gérer la création et l'accès aux différents comptes utilisateurs,
- un module de génération de clés qui s'occupe de la création de nouvelles paires de clés,
- un module de cryptage/décryptage qui se charge du chiffrement et du déchiffrement du message,
- un module de transmission pour acheminer un message à sa destination,
- enfin, un module de fermeture qui s'occupe d'archiver, de mettre à jour et de sécuriser toutes les données propres à l'utilisateur avant de fermer la session en cours.

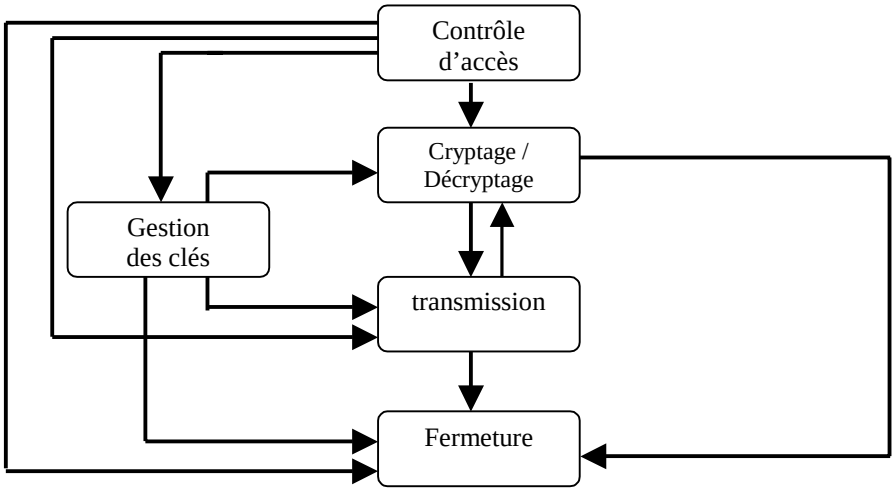

Figure 1 : Architecture du système

Pour assurer la confidentialité, le corps du message est compressé, puis crypté avec un ou deux algorithmes symétriques, selon le niveau de sécurité choisi (paragraphe IV). Une fonction de hachage est utilisée pour assurer l'intégrité du message. L'authentification et la non-répudiation sont assurées grâce à la signature numérique en utilisant un algorithme asymétrique.

Notre choix s'est porté sur les algorithmes symétriques BLOWFISH, TWOFISH, IDEA, TRIPLE DES et l'algorithme asymétrique RSA[DOB 95, WHI 98, KLE 00, ZIM 98].

Le schéma général d'un message chiffré est donné par la figure 3. Le message est partagé en deux parties. Le corps du message et les informations propres au système.

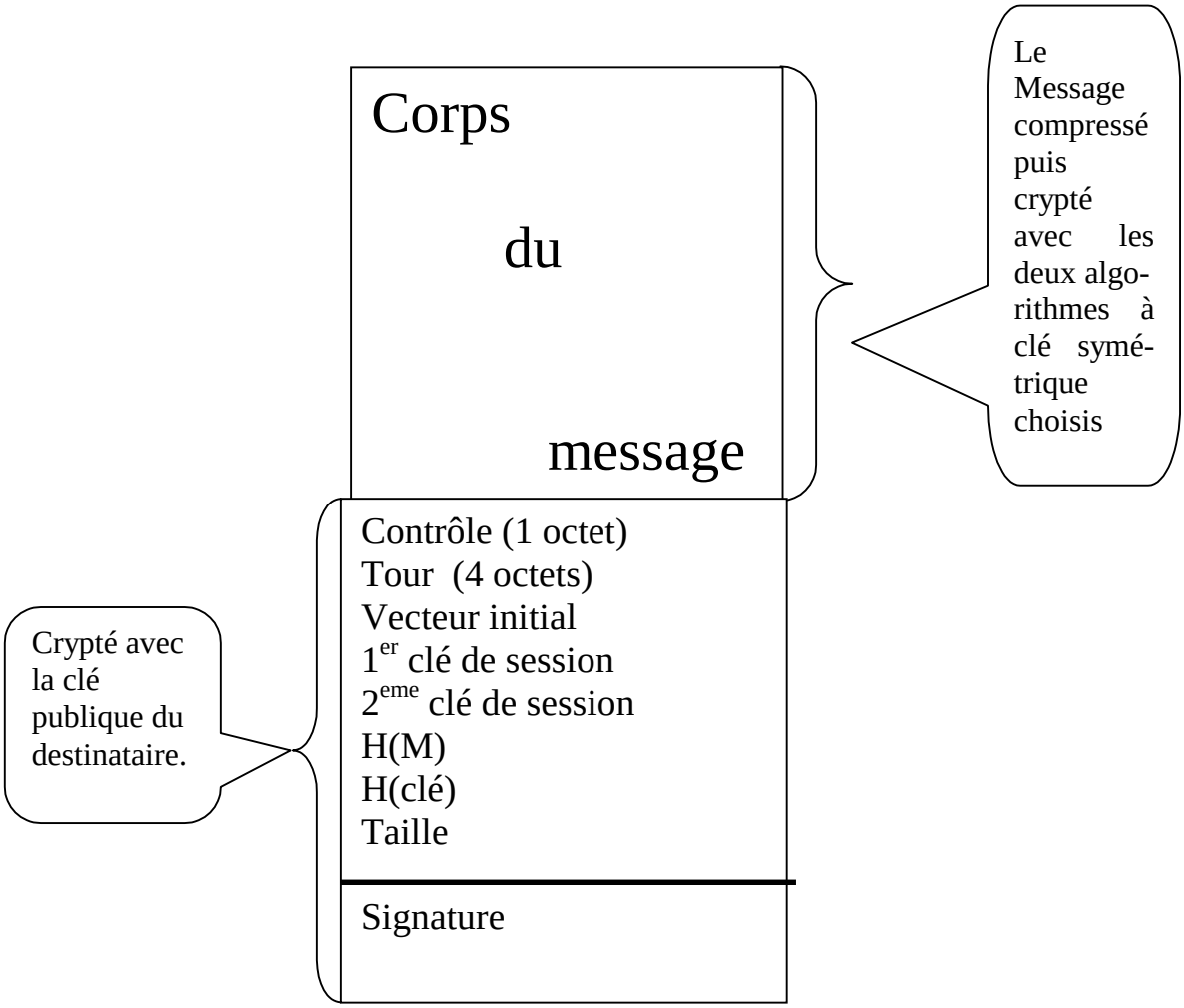

Figure 3 : Le schéma général d'un message chiffré.

- Le corps du message : Cette partie est chiffrée par un ou deux algorithme(s) symétriques selon le niveau de sécurité choisi par l'utilisateur.

- La partie système : Cette partie est cryptée avec un algorithme asymétrique RSA, et elle est composée des informations suivantes :

- **Contrôle (1 octet) :** Cet octet est utilisé pour indiquer les options du cryptage (confidentialité, intégrité, signature, niveau de sécurité).
- **Le tour (4 octets)(optionnel) :** C'est le nombre de blocs à crypter avec chaque algorithme (voir paragraphe III : Mode de chaînage CMBBC).
- **Le vecteur initial :** Il est utilisé pour commencer le chiffrement. Sa taille et son contenu différent selon le mode de cryptage.
- **Les clés de session :** Une ou deux clés sont générées, aléatoirement, et utilisées pour crypter le corps du message (réinitialisés pour chaque message). La taille des clés varie en fonction de l'algorithme utilisé (selon le niveau de sécurité).

- **H(M) (optionnel) :** Digeste de la fonction de hachage appliquée sur le message sur demande de l'utilisateur. La fonction de hachage utilisée est SHA.
- **H(key) (optionnel) :** Digeste de la fonction de hachage appliquée sur la clé de signature.
- **Taille (optionnel) :** Taille de la clé de signature.
- **Signature (optionnel) :** Le chiffrement du digeste de H(M) avec la clé secrète.

On remarque que le rôle du chiffrement asymétrique se limite à chiffrer les clés de session, le résidu de la fonction de hachage et les informations système. Pour réaliser le chiffrement asymétrique, l'algorithme RSA est utilisé avec des clés allant de 512 bits à 2048 bits.

III. Le mode de chiffrement

Le mode de chiffrement choisi pour notre système est le CBC. C'est l'un des meilleurs modes et l'un des plus utilisés [KLE 00, IBR 98]. Cependant, afin de renforcer la sécurité du chiffrement, nous avons effectué certaines modifications sur ce mode. Ces modifications ont donné naissance à deux nouveaux modes que nous avons appelé :

- CBBDC (Chaining Bloc Before Double Cipher),
- CMBBDC (Chaining and Mixing Bloc Before Double Cipher).

- Le mode CBBDC (Chaining Bloc Before Double Cipher)

La **première modification** qui a été opérée sur le CBC est de faire un XOR[KLE 00] entre les deux blocs en clair, au lieu de le faire entre le bloc en clair et le bloc chiffré précédemment comme c'est le cas dans le mode CBC. La **deuxième modification** a été de chiffrer le message en alternant les algorithmes. **De plus**, le choix du premier algorithme à appliquer se fait d'une manière aléatoire.

Si un attaquant arrive à découvrir l'une des clés, il ne pourra jamais lire le message ni même le bloc qu'il vient de déchiffrer et ceci pour deux raisons :
La première, est que le bloc précédent est chiffré avec une clé différente ce qui rend son déchiffrement impossible (sans la connaissance de la clé utilisée).
La deuxième, est qu'un XOR a été effectué entre le bloc piraté et le bloc en clair précédent. De ce fait, pour pouvoir lire le bloc piraté on aura besoin de connaître le bloc en clair précédent pour lequel un XOR a été appliqué avec un bloc chiffré par un algorithme différent.

- Le mode CMBBDC (Chaining and Mixing Bloc Befor Double Cipher)

Le mode CMBBDC est pratiquement le même que le mode CBBDC précédent à une différence près. Dans le mode précédent, le chiffrement se faisait d'une manière alternée bloc par bloc. Dans cette méthode, les blocs ne seront plus cryptés 1 à 1 mais, un nombre *n* appelé *Tour* va être généré aléatoirement de telle sorte que n blocs sont

cryptés avec l'algorithme 1 puis n blocs avec l'algorithme 2, etc. De plus, comme dans la méthode précédente, l'ordre de chiffrement des blocs sera brouillé par un chaînage de la manière suivante : bloc 1, bloc n+1, bloc 2, bloc n+2, …,bloc n, bloc 2n,... De cette manière, l'attaquant ne pourra jamais connaître ni la position des blocs ni l'algorithme avec lequel a été crypté chaque bloc. Ce qui rend le déchiffrement encore plus difficile.

Le nombre n est généré comme suit: *n = 1 + nb modulo P*, *avec P = taille (message)/2*, et *nb* un nombre aléatoire généré à partir de l'heure en milli secondes et la position de la souris à l'écran. De cette manière, n sera toujours supérieur à 1 et inférieur à P ce qui veut dire qu'il y aura toujours au moins deux rondes [BAR 00].
Le seul inconvénient de cette méthode est qu'elle est plus lente que la précédente, ce qui ralentit le chiffrement. De ce fait elle a été utilisée comme option pour les deux modes moyenne et haute sécurité.

IV Les différents niveaux de sécurité

Le système est doté de trois niveaux de sécurité:

- Bas niveau de sécurité
- Moyen niveau de sécurité
- Haut niveau de sécurité

Chacun de ces trois niveaux, utilise un certain nombre d'algorithmes répartis comme suit :

IV.1 Mode Basse sécurité

Ce mode utilise l'un des quatre algorithmes suivant en mode CBC simple. Il offre une sécurité suffisante pour un échange de messages personnels :

- BLOWFISH avec une clé d'une taille de 256 bits et des blocs d'une taille de 128 bits.
- TWOFISH avec une clé d'une taille de 256 bits et des blocs d'une taille de 128 bits.
- 3-DES avec une clé d'une taille de 192 bits et des blocs d'une taille de 64 bits.
- IDEA avec une clé d'une taille de 128 bits et des blocs d'une taille de 64 bits.

IV.2 Mode moyenne sécurité

Ce mode utilise les deux algorithmes suivants en mode CBBDC ou CMBBDC selon le choix de l'utilisateur.

- 3-DES avec une clé d'une taille de 192 bits et des blocs d'une taille de 64 bits.
- IDEA avec une clé d'une taille de 128 bits et des blocs d'une taille de 64 bits.

Afin de préserver un niveau fiable de sécurité dans ce mode, le chiffrement avec RSA exigera une clé d'une taille supérieure à 1024 bits et inférieure à 2048 bits.

IV.3 Mode Haute sécurité

Ce mode utilise les deux algorithmes suivant en mode CBBDC ou CMBBDC selon le choix de l'utilisateur.

- BLOWFISH avec une clé d'une taille de 256 bits et des blocs d'une taille de 128 bits.
- TWOFISH avec une clé d'une taille de 256 bits et des blocs d'une taille de 128 bits.

Dans ce mode la taille de la clé doit être supérieure à 2000 bits et inférieure à 2048 bits

V Chiffrement / Déchiffrement

V.1 Chifrement

Avant de chiffrer un message, il est compressé avec PK-ZIP qui a pour but de réduire la taille à fin de rendre le chiffrement plus rapide, mais plus encore réduire la redondance afin de rendre la cryptanalyse plus difficile. Ensuite, l'utilisateur choisit un niveau de sécurité selon ses besoins, et sélectionne les options de sécurité offertes par le système :

- La confidentialité, permet le chiffrement du message avec l'un des niveaux sélectionnés.
- L'intégrité est assurée par l'application de la fonction de hachage SHA qui délivre un résidu de 160 bits.
- La signature, permet d'identifier l'auteur du message (authentification et non-répudiation).

Une fois les options choisis, l'opération de chiffrement est lancée en passant par les étapes suivantes:

- Créer le fichier cible dont le nom a été spécifié par l'utilisateur.
- Sauvegarder les bits de contrôle.
- Calculer le tour.
- Générer les clés de session à partir des paramètres aléatoires suivants :
  - ✓ L'heure en milli secondes.
  - ✓ La position de la souris à l'écran.

  Le système récupère 05 octets de ces paramètres auxquels il va appliquer l'algorithme MD5[1][KLE 00] quinze fois consécutives, ce qui générera une suite de 256 bits à partir de laquelle sera extraite la clé de session.

Calculer le vecteur initial. Il diffère selon les algorithmes et les niveaux de sécurité choisis. Pour le mode basse sécurité, chaque algorithme s'occupe du calcul de ses propres clés. Tandis que pour les niveaux haute et basse sécurité, le vecteur initial est calculé à partir des clés de session.

[1] MD5 est une fonction de hachage puissante.

- Appliquer une fonction de hachage au message si l'utilisateur a sélectionné cette option.
- Appliquer une fonction de hachage sur la clé de signature afin de faciliter la recherche au niveau du destinataire.
- Insérer la taille de la clé de signature.
- Insérer la signature.
- Crypter toutes les informations précédentes avec l'algorithme RSA en utilisant la clé sélectionnée par l'utilisateur.
- Crypter le message avec les algorithmes symétriques.
- Concaténer les deux blocs dans un seul fichier.

Enfin, le message est envoyé grâce au module de transmission.

V.2 Le déchiffrement

L'opération de déchiffrement passe par les étapes suivantes :

- Le système décrypte, avec la clé secrète de l'utilisateur, le 1er bloc et récolte les octets de contrôle nécessaires au décryptage.
- Calcule le nombre des blocs à décrypter avec le RSA en fonction des bites de contrôle.
- En cas de présence d'une signature, il récupère la clé publique nécessaire au déchiffrement de la signature en effectuant une recherche parmi les clés disponibles dans le trousseau du destinataire et ceci en utilisant le digeste. S'il trouve la clé alors il calcule la taille du bloc de signature et déchiffre la signature.
- Il vérifie cette dernière en comparant la fonction de hachage avec le résultat obtenu lors du déchiffrement de la signature.
- Enfin, il déchiffre le message reçu en fonction des bits de contrôle et affichera un compte rendu.

Conclusion

Dans cet article, nous avons traité des problèmes de sécurité auxquels peuvent être confrontés les messages électroniques durant leur phase de transfert sur le réseau. En effet, Le système développé a été conçu dans le but d'offrir à l'utilisateur la possibilité de sécuriser ses e-mails contre d'éventuelles attaques pouvant porter atteinte à leur confidentialité (l'écoute du trafic et espionnage), leur intégrité (modification du message), leur authentification et par la même celle de leur expéditeur (imitation) et aussi leur non-répudiation (le fait de nier l'émission d'un message). En fonction de l'importance des informations à protéger, il a la possibilité de choisir un parmi trois niveaux de sécurité selon ses besoins ou ceux de son application:

- ✓ Le haut et moyen niveau qui utilisent comme algorithmes « Blowfish » et « Towfish », « Triple DES » et « IDEA » respectivement avec les deux modes de chaînages CBBDC et CMBBDC.
- ✓ Le bas niveau qui utilise les quatre algorithmes précédents, mais séparément en mode CBC.

Il faut souligner qu'en plus d'avoir basé la mise en œuvre de notre système sur les algorithmes les plus performants (en particulier en terme de robustesse et de rapidité) actuellement connus dans le domaine, nous avons proposés et implémentés des mécanismes nouveaux renforçant la sécurité du mode de chiffrement. En effet, sur la base de modifications opérées sur le mode de chaînage CBC (Chaining Bloc Before Cipher) qui est l'une des techniques la plus utilisée, nous avons aboutit à deux nouveaux modes CBBDC (Chaining Bloc Before Double Cipher) et CMBBDC (Chaining and Mixing Bloc Before Double Cipher) dont nous avons montré l'apport par rapport au mode CBC dans le paragraphe V.2. Comme continuité à ce travail, nous prévoyons entre autre de valider et de vérifier les performances de ces deux modes de chaînage à travers leur évaluation dans un environnement sujet à des problèmes de sécurité et plus précisément à des attaques de cryptanalyse.

Références Bibliographiques

[BAL 91] R.Balter & S.Krakowiak & J.P.Banâtre
Construction des systèmes d'exploitation répartis, INA France 1991.

[BER 00] B. Berbar, R. Saadi, N. Nouali, O. Nouali
Protection de la messagerie électronique par des méthodes cryptographiques, mémoire de fin d'études d'ingénieurs, USTHB, CERIST, Octobre 2000

[DOB 95] Dr. Dobb's Journal
http://www.contentepane.com/blowfish/analysis.html
Septembre 1995.

[FAQ 96] Mathematical cryptology
http://www.faqs.org/faqs/cryptography-faq/
Février 96.

[FRE 99] Miller freeman
The politics of cryptography
http://www.performancecomputing.com/features/9910f1.shtml
October 1999

[GDI 98] Groupe de discussion : Politique cadre en matière de cryptographie
http://strategie.ic.gc.ca/ssgf/cy00008f.html
Février 1998.

[GUT 97] Barbara Guttman & Robert Bagwill
Internet Security Policy : A Technical Guide. NIST Special publication 800-xx, 31 juillet 1997.

[HAE 97] Reto E.Haeni
Information warfare, an introduction, Rapport de recherches, George Washington Univerity, Janvier 1997.

[IBR 98] Bertrand Ibrahim
Rapport Internet et sécurité, Université de Genève
05/06/1998.

[KAM 97] Iren Kam
Network security Department of Computer Science Rensselaer Polytechnic Institute USA
www.cs.rpi.edu/~iren/project.html
1997.

[KAU 00] Lorraine Kauffman
Certicom ECC challenge inroduction, Public Relations Manager, Certicom Corp
http://www.certicom.com/ellypyticcurves/f001.htm
Janvier 2000.

[KLE 00] Matthieu Klein
PGP et applications cryptographiques
Rapport de licence EEA sur les techniques de chiffrement
http://www.hrnet.fr/~matthieu/crypto/crypto.html
03/04/2000.

[MAR 99] Marshall D.Abrams and Harold J.Podell.
Cryptography, 1999.

[MEI 99] Carolyn Meinel, Ronald Rivest
La sécurité sur Internet N°260
http://www.pourlascience.com/numerous/pls-260/internet.html
Juin 99 .

[MEN 97] Alfred J.Menzus, Paul C.Van Oorschot, Scott A.Vanstone
Handbook of applied cryptography, By CRC Press, Inc
1997.

[SAH 98] Nabil Sahli
A synthesis of state of the art in Internet security and guidelines for developing countries, Proc of 5th conf of Computer Communication, Africom CCDC'98, Tunisia, 20-22 october 1998.

[SCH 97] Bruce Schneier
Cryptographie appliquée : algorithmes protocoles et codes sources, Traduction de Laurent Vienol 2eme édition Paris 1997 ISBN : 2-84180-036-9.

[SEC 99] Introduction à la sécurité
http://xtream.online.fr/project/securite.html
1999.

[SSH 00] SSH communication security
http://www.ssh.fi/tech/crypto/algorithms.html
Janvier 2000.

[WHI 98] Bruce Schneier, JhonKelsey, Neils Ferguson, Doug Whiting,

David Wagner
Twofish : a 128 bits cypher
http://www.counterpane.com/twofish.ps.zip
Juin 1998.

[ZIM 98] Phill Zimmerman
Introduction à la cryptographie
1998.