UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE D'ANTANANARIVO

DEPARTEMENT ELECTRONIQUE

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR

Spécialité : ELECTRONIQUE

Option : INFORMATIQUE APPLIQUÉE

VERIFICATION DU LOCUTEUR PAR RECONNAISSANCE VOCALE

Présenté par :

RASAMIJAONA TOJOMANGA ALAIN MICHEL

Soutenu le : 15 JUIN 2011

N° d'ordre : 05/EN/IA /2010

Année Universitaire : 2009-2010

UNIVERSITE D'ANTANANARIVO

ECOLE SUPERIEURE POLYTECHNIQUE D'ANTANANARIVO

DEPARTEMENT ELECTRONIQUE

MEMOIRE DE FIN D'ETUDES EN VUE DE L'OBTENTION DU DIPLOME D'INGENIEUR

Spécialité : **ELECTRONIQUE**

Option : **INFORMATIQUE APPLIQUÉE**

VERIFICATION DU LOCUTEUR PAR RECONNAISSANCE VOCALE

Présenté par :

RASAMIJAONA TOJOMANGA ALAIN MICHEL

Soutenu le : **15 JUIN 2011**

Devant le Jury :

- Monsieur **RASTEFANO Elisée** *Président*
- Monsieur **RATSIMBA Mamy Nirina** *Examinateur*
- Monsieur **RATSIMBAZAFY Guy Predon Claude** *Examinateur*
- Monsieur **HERINANTENAINA Edmond Fils** *Examinateur*

Rapporteur :

Monsieur **ANDRIAMANANTSOA Guy Danielson**

Année Universitaire **: 2009-2010**

REMERCIEMENTS

Aussi vif que soit mon désir de réussir, cet ouvrage n'a pas pu être réalisé sans l'aide de Dieu Tout Puissant que je loue pour sa bonté éternelle.

Par ailleurs, je tiens à exprimer ma profonde gratitude et reconnaissance à ceux qui n'ont pas ménagé leurs efforts et leurs précieux temps en m'apportant gracieusement aide et contribution dans mon travail en particulier :

- *Monsieur RATSIMBA Mamy Nirina, Chef du Département Electronique, qui s'est toujours montré soucieux non seulement du bon déroulement de ma formation mais aussi de ma bonne éducation durant mes cinq années d'études à l'Ecole Supérieure Polytechnique d'Antananarivo, et aussi en tant que membre de jury.*
- *Monsieur RASTEFANO Elisée, qui a voulu présider la soutenance de ce mémoire*
- *Les membres du jury ici présents :*
  - *Monsieur RATSIMBAZAFY Guy Predon Claude*
  - *Monsieur HERINANTENAINA Edmond Fils*
- *Monsieur ANDRIAMANANTSOA Guy Danielson, mon encadreur qui m'a présenté le thème de ce mémoire de fin d'études. Ses orientations et ses conseils m'ont permis d'achever ce projet.*
- *Au corps enseignant du département Electronique pour les formations et les supports les plus appropriés durant ma vie estudiantine.*
- *A tous mes amis, mes familles, pour leurs aides de toutes sortes.*

Que la grâce de Dieu soit continuellement sur vous.

TojoManga

RESUME

Le contrôle d'accès au sein d'un bâtiment ou d'un système informatique exige aujourd'hui des systèmes d'authentification plus sûrs et simples à réaliser, le système d'authentification par voix en est l'un. Cet ouvrage se base à la conception et à l'élaboration d'un système de vérification d'identité par voix: le logiciel « RecSpeaker version 1.0 », dont le premier chapitre parle des outils utiles à la réalisation comme la voix, la production de la voix, le phonème et la prosodie. Ensuite la biométrie et les techniques biométriques sont mentionnées, suivies de l'introduction sur la reconnaissance vocale.

Le deuxième chapitre se fixe sur : l'échantillonnage d'un signal qui est la première étape de la numérisation, la quantification consistant à convertir le signal échantillonné sous forme binaire et le codage pour associer à un ensemble de valeurs discrètes un code composé d'éléments binaires.

Le troisième chapitre montre le principe de la reconnaissance du locuteur avec ses conditions nécessaires et les modes pour reconnaître le locuteur. Puis le principe de vérification du locuteur suivi des différentes phases de la procédure de vérification sont traités.

Et le dernier chapitre décrit le logiciel « RecSpeaker version 1.0 » un logiciel de vérification d'identité d'un individu et de protection d'un fichier quelconque dans le disque dur grâce à la voix d'une personne préenregistrée.

SOMMAIRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ABREVIATIONS

API : Application Programming Interface.

DCB : Décimal codé binaire

DEA : Data Encryption Algorithm

DES : Data Encryption Standard

EM : Expectation-Maximisation.

FFT : Fast Fourier Transform.

GDI : Graphical Device Interface

GMM : Gaussian Mixture Model.

LSB : Least Significant Bit

MFCC : Mel-frequency Cepstral Coefficient.

RC4 : Rivest Cipher 4

RSA : Rivest Shamir Adlem

SQL : Structured Query Language

TFD : Transformée de Fourier Discrète.

INTRODUCTION

La voix est devenue aujourd'hui une modalité biométrique grâce aux caractères acoustiques faciles à déterminer en elle d'une part et la simple manipulation de ces caractères de l'autre. De ce fait, la biométrie vocale peut concurrencer aux autres biométries (empreinte digital, iris, …) en vue de sa simplicité, sa fiabilité et sa facilité à réaliser. Beaucoup de domaines comme télécommunication, bureautique, services et commerce, utilisent désormais cette biométrie. L'authentification par voix ou la vérification d'identité par voix est le cas pratique de la biométrie vocale. A part le contrôle d'accès au sein d'un bâtiment ou d'un système informatique s'ajoutent les applications comme les transactions téléphoniques et les applications criminalistes qui sont les applications adéquates de la reconnaissance vocale (reconnaissance du locuteur). De plus face à l'évolution de l'électronique, de la mathématique et de l'informatique, la technologie en matière de biométrie ne s'arrête pas de se progresser surtout la biométrie vocale qui devient un outil de reconnaissance d'identité et de sécurité très connu mondialement.

Ce mémoire de fin d'étude intitulé « ***VERIFICATION DU LOCUTEUR PAR RECONNAISSANCE VOCALE*** » a pour objectif de concevoir et d'élaborer un système de reconnaissance vocale ou un système de reconnaissance du locuteur pour vérifier vocalement l'identité d'une personne. Ce qui nous amène à la réalisation du logiciel « ***RecSpeaker version 1.0*** » possédant deux fonctions : la vérification d'identité d'un individu par sa voix et la protection d'un fichier quelconque vocalement.

Ce rapport est divisé en quatre chapitres : le premier chapitre parle la généralité sur la reconnaissance vocale, le second chapitre est orienté sur la numérisation d'un signal, le troisième chapitre explique brièvement la vérification du locuteur suivi de la réalisation du logiciel « ***RecSpeaker version 1.0*** » en quatrième chapitre.

Chapitre 1 : GENERALITE SUR LA RECONNAISSANCE VOCALE

1.1 Voix

Définition

La voix est l'ensemble des sons caractérisés par deux fonctions mécaniques de base : la phonation qui consiste à la production d'un phénomène acoustique et l'articulation qui consiste à la modulation de ce dernier.

1.2 Parole

1.2.1 Caractéristiques

La parole est un moyen de communication naturel de l'humain avec une efficacité très importante.

Elle se distingue des autres sons par ses caractéristiques acoustiques qui ont leur origine dans les mécanismes de production. Elle apparait physiquement comme une vibration de pression de l'air causée par le système articulatoire. Les sons de la parole sont produits soit par des vibrations des cordes vocales (c'est la source de voisement), soit par une turbulence créée par l'air s'écoulant rapidement dans une constriction ou lors du relâchement d'une occlusion du conduit vocal (c'est la source de bruit).

1.2.2 Production de la parole

Essentiellement, il y a trois étapes pour le processus de la phonation :

- Premièrement, il faut avoir une énergie respiratoire suffisante pour mettre en mouvement les cordes vocales et générer des bruits.
- Les cordes vocales vibrant, provoquent la naissance des sons voisés.
- Enfin, une gestuelle articulatoire au niveau du conduit vocal et fosses nasales se réalise.

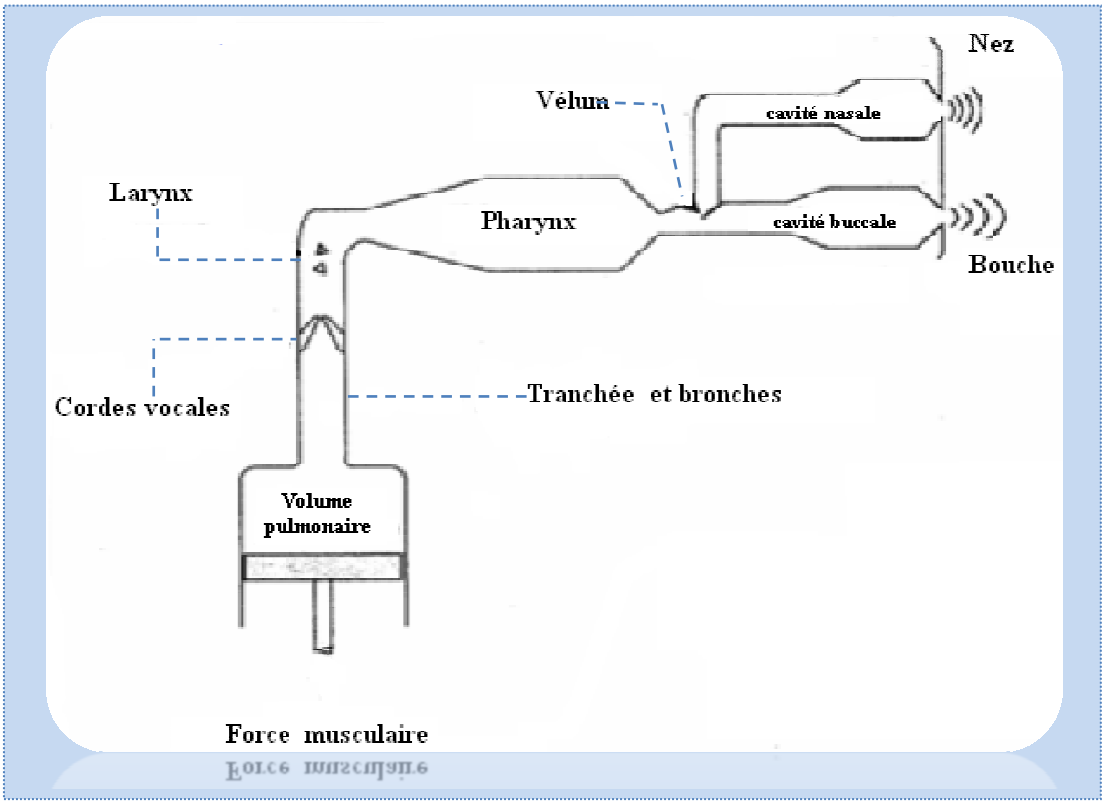

Figure 1.1 : **Production de la parole**

1.2.3 Signal de la parole

Le signal vocal est caractérisé par:

- Sa fréquence
- Son intensité (ou le niveau sonore)
- Son timbre (ou « la richesse » du signal)

a. Fréquence

La fréquence qui est l'inverse de la période **T**, est le nombre d'oscillation dans une seconde.

On peut aussi avoir la fréquence **F** à partir de la formule suivante :

$$F=C/\lambda$$

, avec **C** la célérité ou la vitesse du son en **m/s** et **λ** la longueur d'onde en **m**.

- **Fréquence fondamentale**

Le signal de la parole comprend un son fondamental et des harmoniques dont les rapports de fréquences avec la fondamentale sont des quotients de nombres entiers. Toute vibration sonore peut être décomposée en une somme de fonctions sinusoïdales élémentaires dont les périodes plus courtes sont proportionnelles avec sa propre période (c'est la décomposition en « série de Fourier »).

- **Fréquence harmonique**

C'est la fréquence multiple de la fréquence fondamentale $\mathbf{F_0}$, c'est-à-dire

$f_n = nF_0$ avec **n** ϵ {2,3,…}. Une harmonique correspond à une fonction trigonométrique sinusoïdale dont la fréquence est un multiple de la fréquence de la fonction périodique décomposée. La somme de toutes les harmoniques d'une fonction périodique reconstitue la fonction.

Prenons par exemple un signal acoustique s(t) qui est la superposition de trois sinusoïdales pures dont la fréquence fondamentale est f = 440Hz et de fréquences harmoniques f_2 = 880Hz, f_3 = 1320Hz, d'équation :

$$\mathbf{s(t) = \sin 2\pi f_1 t + \sin 2\pi f_2 t + \sin 2\pi f_3 t}$$

Les graphes de cette équation en fonction du temps et de la fréquence sont illustrés par la fig 1.2.

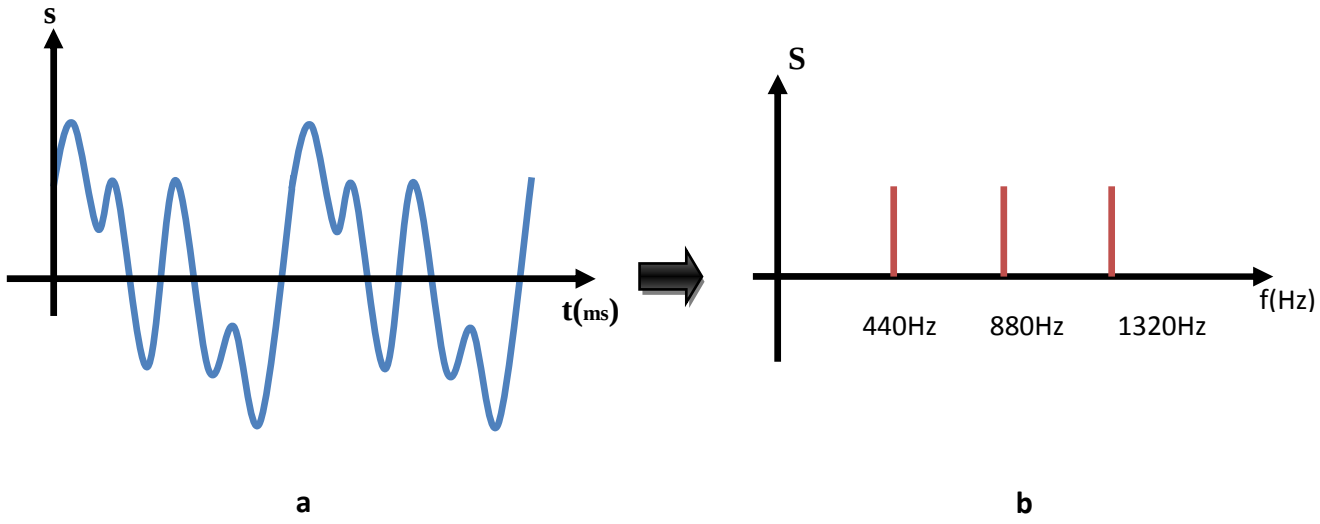

Figure 1.2 : **Signal acoustique s, a : Représentation en fonction du temps, b : Représentation fréquentielle**

b. Intensité

Le son est une onde qui se propage de façon omnidirectionnelle. L'énergie acoustique **E** (en joule) produite par la source sonore est repartie sur une surface sphérique de plus en plus grande au fur et à mesure de la propagation de l'onde sonore. L'intensité notée **I**, est une fonction dépendante de la puissance **P** de l'émetteur et la distance notée **r**, qui sépare la source et le lieu d'écoute **[1]**.

L'intensité est donnée par la formule :

$$I = \frac{P}{4\pi r^2}$$

, avec **P** est en **W** et r en **m** => I en **W/m²**.

Elle caractérise aussi le volume du signal mais dépend de l'amplitude du signal considéré.

En acoustique, elle s'exprime en décibel en raison de deux facteurs :

- Les valeurs obtenues sont faciles à manipuler (elles ne sont trop grandes ni trop petites)
- La perception humaine sur l'intensité sonore se fait de façon logarithmique.

L'intensité acoustique (en **dB**) est alors définie comme suit :

$$L = 10 \log I/I_0$$

Où $\mathbf{I_0}$ le son le plus faible que l'on puisse entendre pour un signal de fréquence 1 kHz. Elle est appelée aussi intensité de référence.

$$I_0 = 10^{-12}\ W/m^2$$

Pour que l'oreille perçoive un son dit audible, l'intensité sonore **I** doit être :

$$10^{-12}\ W/m^2 < I < 25\ W/m^2$$

Où 10^{-12} $\mathbf{W/m^2}$ est la limite de sensibilité de l'oreille et 25 $\mathbf{W/m^2}$ borne supérieure de l'intensité sonore, correspond à une destruction de l'oreille.

c. <u>Timbre</u>

Il correspond à la richesse d'un signal sonore d'un instrument ou d'une voix d'une personne. Il est caractérisé par ses fréquences harmoniques, ses nombres et ses amplitudes. C'est pour cette raison qu'une même note ne produira pas le même son avec deux instruments différents. On dit qu'un son est riche lorsqu'il possède beaucoup d'harmoniques et pauvre lorsqu'il a moins d'harmoniques.

1.3 Phonème

Le phonème est la plus petite unité phonique distinctive, qui n'est défini sur une base acoustique, ni articulatoire, ni perceptuel, mais sur le plan fonctionnel. Ainsi, les phonèmes n'ont pas d'existence indépendante, c'est-à-dire, ils constituent un ensemble structuré dans lequel chaque élément est intentionnellement différent de tous les autres et la différence étant à chaque fois porteuse de sens **[3].**

Voici un exemple des phonèmes de la langue française. Dans la langue française, il existe 36 phonèmes.

Tableau 1.1 : **Les phonèmes de la langue française**

CONSONNES	VOYELLES
Paie, baie, mais, fait, vais, ouais, taie, dais, nez, sait, huer, lait,….	*Lit, les, là, lin, lu, leu, leur, le, lent, loup, lot, lotte, long, ….*

1.4 Prosodie

La prosodie est la façon de décrire ou de présenter formellement les éléments de l'expression orale à savoir les tons, les accents, l'intonation et la qualité. Ces éléments transmettent des informations sur la signification d'un énoncé. En d'autres termes, elle désigne les phonèmes liés à l'évolution dans le temps des paramètres de hauteur, d'intensité et de durée.

La perception de hauteur est essentiellement liée à la fréquence fondamentale qui correspond au niveau physiologique de la production et à la fréquence de vibration des cordes vocales.

La perception d'intensité est essentiellement liée à l'amplitude et à l'énergie du son, mais partiellement dépend aussi avec sa durée.

La perception de durée correspond à son temps d'émission et sa durée acoustique. A noter que le terme « durée » est utilisé pour désigner à la fois le paramètre perceptif et le paramètre acoustique et le terme « longueur » comme synonyme de durée perçue est utile quand la distinction est importante **[4]**.

- **Mélodie**

Elle est constituée par la variation dans le temps de la fréquence fondamentale, ou de la hauteur si l'on se place du point de vue perceptif. L'enchainement des durées relatives (y compris les durées des silences) constitue le rythme.

Avec le modèle d'intonation, il existe quatre niveaux d'intonation : basse, moyenne, haute et aiguë. Cette modélisation met en jeu les trois modalités suivantes : l'interrogation, l'exclamation et l'affirmation.

Les substitutions entre les intonations dans une phrase de même contenu, entrainent des changements de sens. Cela montre notamment que l'intonation joue un rôle important pour la compréhension du message vocale.

- **Ton**

Le mot **ton** désigne le ou les niveaux de hauteurs observées dans une syllabe donnée. Le ton coïncide donc avec la partie de la courbe mélodique qui se rattache à une seule syllabe.

L'intonation d'un énoncé se présente comme une succession de ton. Et on distingue quatre niveaux de hauteur : haut, bas, infra-bas et suraigu.

- **Accent**

L'accent se situe par la manifestation d'intensité, de hauteur et/ou de durée, portant sur une syllabe.

L'équation suivante résume ce qui précède :

Prosodie = F_0 + énergie + durée (grandeur acoustique)

= hauteur + intensité +longueur (grandeur perçues)

=mélodie + rythme (structures) + accentuation

1.5 Notion de la biométrie

1.5.1 Identité

L'identité est une notion complexe, difficile à définir.

Du point de vue personnel, la caractérisation de l'identité prend en compte tout ce que l'individu considère comme faisant partie intégrante de lui et qui ne peut lui être enlevé.

Du point de vue externe, l'identité d'un individu est la façon dont il perçu par le monde qui l'entoure **[5].**

Pour identifier une personne, trois approches sont possibles:

Utiliser un identifiant : ce que l'on possède (carte, badge, document).

Utiliser une connaissance : ce que l'on sait (mot de passe).

Utiliser une biométrie : ce que l'on est.

1.5.2 Biométrie

a. Définitions

C'est la science qui étudie, à l'aide des mathématiques (statistiques, probabilités), les variations biologiques à l'intérieur d'un groupe déterminé.

Autrement dit, c'est une méthode permet d'identifier ou de vérifier l'identité (authentification) d'une personne sur la base de données reconnaissable et vérifiable qui lui est propre.

b. Authentification biométrique

La biométrie permet l'authentification d'individus à partir de leurs caractéristiques physiologiques ou comportementales qui doivent être :

– universelles : présentes chez tous les individus.

– uniques : spécifiques à chaque individu.

– permanentes : pour permettre une authentification au cours du temps.

– mesurables : pour permettre l'enregistrement et les comparaisons futures.

Avantages

L'authentification biométrique présente de nombreux avantages :

- Elle permet de s'affranchir des intermédiaires que constituent les clefs, cartes et autres codes personnels susceptibles d'être oubliés, perdus ou volés.
- Elle supprime le risque qui peut être occasionné par le prêt d'une clef ou la communication d'un mot de passe à un tiers.
- L'utilisation de données intrinsèques à l'utilisateur lui permet, de plus, de recourir à la biométrie en tout lieu et à tout moment.

c. Différents types de biométrie

- **Biométrie morphologique**

Elle décrit les individus par des mesures de leurs caractéristiques biologiques ou physiologiques qui sont moins sujettes à l'influence du stress que la biométrie comportementale.

Exemples : empreintes digitales, le réseau veineux de la rétine, l'iris, l'empreinte, etc.

- **Biométrie comportementale**

La biométrie comportementale mesure et caractérise des éléments qui sont propres aux comportements d'un individu.

Exemple : signature dynamique.

- **Biométrie mixte**

Parfois on ne peut pas distinguer exactement la biométrie morphologique avec biométrie comportementale. D'où le nom biométrie mixte.

Exemple : la voix, qui est utilisée de façon naturelle par les êtres humains pour reconnaître un individu, est une modalité comportementale qui peut subir les influences d'une pathologie, du stress ou même d'un changement émotionnel.

1.6 Reconnaissance vocale

La reconnaissance de la voix (caractéristique propre pour chaque individu) ou la reconnaissance vocale est un terme générique regroupant les problèmes relatifs à la reconnaissance du locuteur, basé sur le contenu de l'information dans le signal acoustique de la parole qui est la faculté de communiquer la pensée par le moyen de sons articulés émis par les organes de la phonation, tandis que la voix, elle représente l'ensemble de sons produits par le système articulatoire et phonatoire.

Dans la reconnaissance vocale, on cherche à trouver ce qui caractérise le locuteur dans le signal acoustique. Ici, l'individualité est présente, parce que le locuteur peut être reconnu aussi bien que par son timbre vocale, que par la hauteur de sa voix, la particularité d'élocution, l'intonation,….

Les systèmes de reconnaissance vocale se concentrent sur les caractéristiques de voix qui sont uniques à la configuration de la parole d'une personne. Les configurations de la parole sont constituées par une combinaison des facteurs comportementaux et physiologiques. Les mouvements des organes de production de la parole engendrent des variations de pression acoustique instantanée qui peuvent être captées par un transducteur (microphone) et transformées en variations de tension électrique.

Un enregistrement de la parole n'est ni un prélèvement direct ni une trace laissée sur une surface au contact d'une partie de son corps, il ne s'agit que de la capture indirecte de mouvements articulatoires complexes faisant intervenir les cordes vocales, la langue, le voile du palais, la mâchoire et les lèvres. La reconnaissance vocale est considérée comme une des formes les moins intrusives de la technologie biométrique, car elle n'exige aucun contact physique avec le capteur (microphone) du système automatique de reconnaissance **[6].**

Chapitre II: NUMERISATION D'UN SIGNAL

2.1 Introduction

L'importance des systèmes numériques de traitement de l'information ne cesse de croître (téléphone, télévision, radio, instrumentation, ….). Ce choix est souvent justifié par des avantages techniques tels que la grande stabilité des paramètres, une excellente reproductibilité des résultats et des fonctionnalités accrues. Le monde extérieur étant par nature « analogique », une opération préliminaire de conversion analogique numérique est nécessaire .Cette conversion est la succession de trois effets sur le signal analogique qui n'est autre que le signal de départ :

- L'**échantillonnage** : qui rend le signal analogique en signal discret
- **La quantification** pour associer à chaque échantillon une valeur
- **Le codage** pour associer un code à chaque valeur.

2.2 Echantillonnage

Il consiste à prélever à des instants précis, le plus souvent équidistants, les valeurs instantanées d'un signal.

Soit s(t) un signal analogique, continue dans le temps, est représenté par un ensemble de valeurs discrètes $s_e(t)$ tel que :

$$\mathbf{s_e(t) = s(\ n.\ Te)}$$

, avec **n** : un entier et $\mathbf{T_e}$: période d'échantillonnage.

Théoriquement, l'opération d'échantillonneur est souvent symbolisée par un interrupteur. La Figure 2.1 montre cette opération.

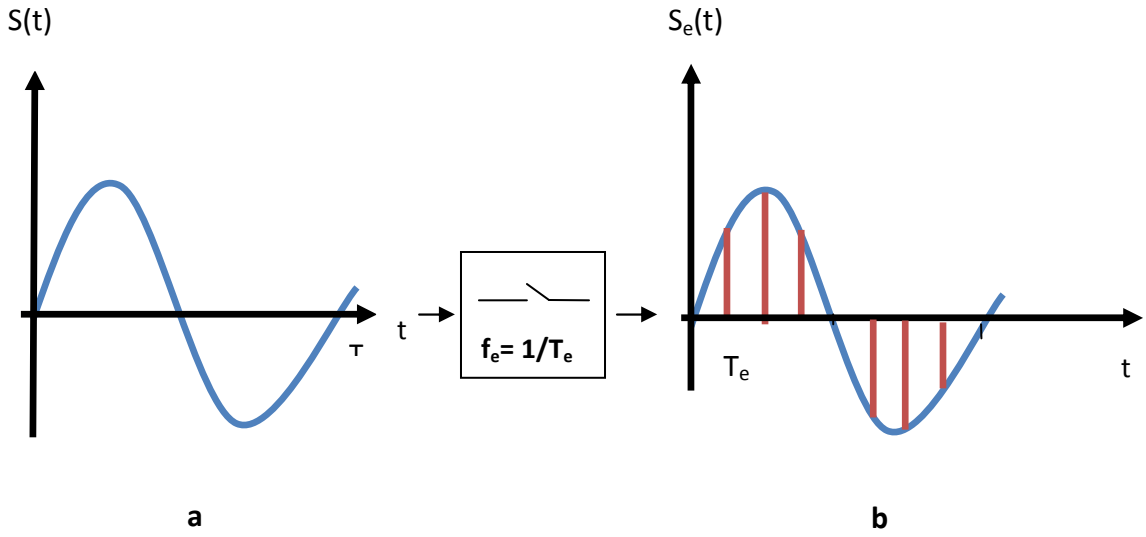

Figure 2.1: **Transformation du signal s(t) continu en signal échantillonné $s_e(t)$, a : signal analogique s(t) , b : signal échantillonné $s_e(t)$**

2.2.1 Echantillonnage idéal [7]

L'échantillonnage idéal est modélisé par la multiplication du signal continu s(t) et d'un peigne de Dirac de période T_e ; c'est-à-dire :

$$S_e(t) = s(t).\ \delta_{Te}(t) = s(t) \sum_{n\to-\infty}^{+\infty} \delta(t - nTe) = s(nT_e) \sum_{n\to-\infty}^{+\infty} \delta(t - nTe)$$

Donc, le spectre du signal échantillonné est le suivant :

$$S_e(f) = \frac{1}{Te} \sum_{n\to-\infty}^{+\infty} S(f) * \delta(f - nfe) \longrightarrow \boxed{\mathbf{S_e(f) = \frac{1}{Te} \sum_{n\to-\infty}^{+\infty} S(f - nfe)}}$$

On obtient donc un spectre infini qui provient de la périodisation du spectre du signal d'origine autour des multiples de la fréquence d'échantillonnage.

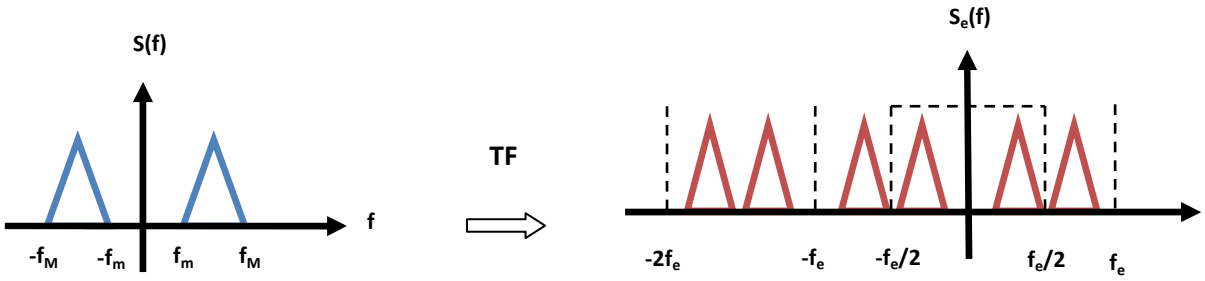

Figure2.2 : **Echantillonnage idéal**

Remarques :

- Sur le signal échantillonné, on voit qu'il est possible de restituer le signal original par un filtre passe-bas.
- Aussi, si $\mathbf{f_M} > \frac{\mathbf{fe}}{2}$, la restitution de l'originale sera impossible car il va apparaître un recouvrement spectral lors de l'échantillonnage ;

  avec $\mathbf{f_M}$ est la fréquence maximale du spectre du signal à échantillonner.

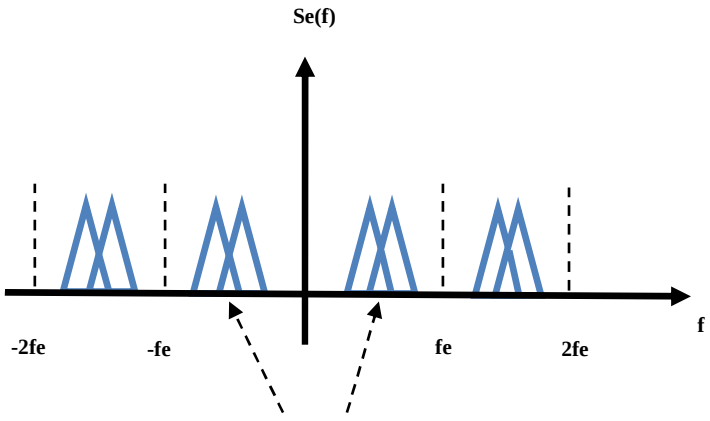

Figure 2.3 : **Recouvrement spectral**

Le théorème de **Shannon** montre que la reconstitution correcte d'un signal nécessite que la fréquence d'échantillonnage $\mathbf{f_e}$ soit au moins deux fois plus grande des fréquences $\mathbf{f_M}$ du spectre du signal :

$$\mathbf{fe > 2f_M}$$

2.2.2 Echantillonnage réel

L'échantillonnage réel est obtenu en commandant un interrupteur par un train d'impulsions étroites. Ce qui veut dire qu'il est impossible d'obtenir des échantillons de durée quasiment nulle. La modélisation de l'échantillonnage par un peigne de Dirac est donc erronée. En fait, chaque impulsion va avoir une durée très courte **λ**. L'échantillonnage peut donc être modélisé par la multiplication du signal par une suite de fonction rectangle ou porte de largeur λ **[7]**.

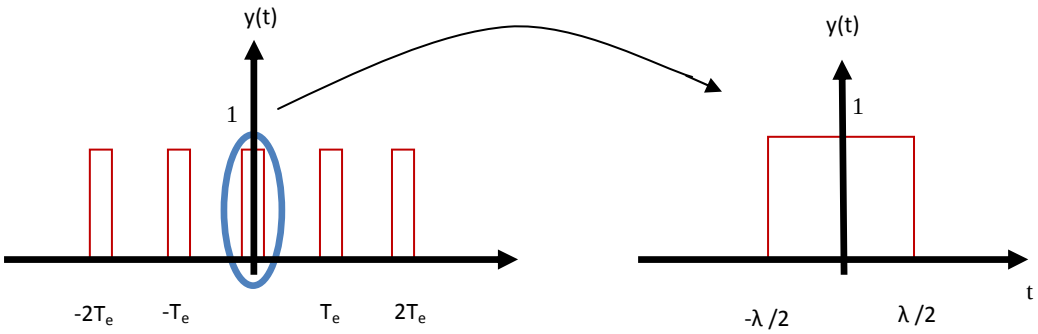

Figure 2.4 : **Signal porte de durée λ**

On a comme expression du signal d'échantillonnage :

$$y(t) = \sum_{k\to-\infty}^{+\infty} \text{rect}\ \frac{t-kTe}{\lambda} = \text{rect}\ (\frac{t}{\lambda}) * \sum_{k\to-\infty}^{+\infty}\ \delta(t-kTe)$$

Par conséquent, sa transformée de Fourier est égale à :

$$Y(f) = \lambda\ \text{sinc}(\lambda f)\ \frac{1}{Te} \sum_{k\to-\infty}^{+\infty}\ \delta(t-kfe)$$

Et comme l'expression du signal d'échantillonné est :

$s_e(t) = s(t).y(t)$

Sa transformée de Fourier devient :

$$S_e(f) = S(f) * Y(f) = S(f) * \frac{\lambda}{Te} \sum_{k\to-\infty}^{+\infty} \text{sinc}(\lambda f) \,.\, \delta(f - kfe)$$

$$\mathbf{S_e(f) = \frac{\lambda}{Te}\, sinc(\lambda f) \sum_{k\to-\infty}^{+\infty} S(f - kfe)}$$

On retrouve la même allure de spectre modulé en amplitude par une fonction en sinus cardinale.

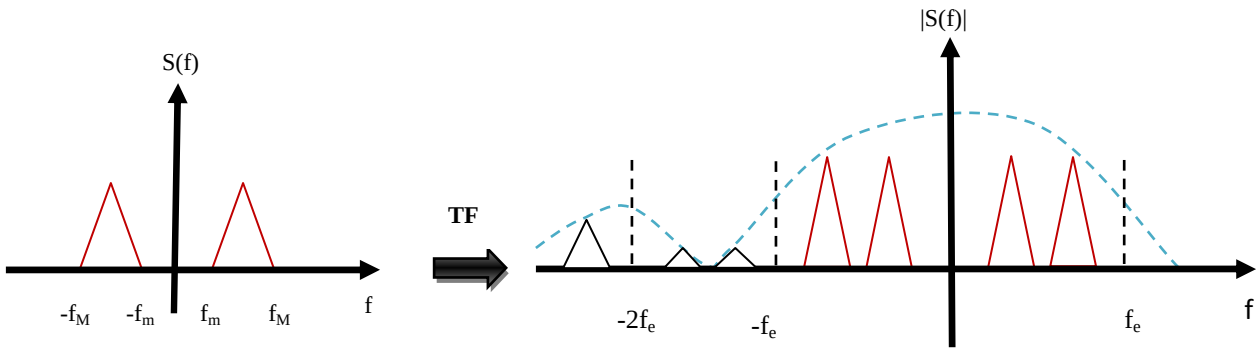

Figure 2.5 : **Spectre modulé en amplitude entrainant une fonction en sinus cardinale**

2.2.3 Echantillonnage-blocage

Pratiquement, on n'échantillonne pas un signal pour le reconstruire juste après. En effet, l'échantillonnage sert à prélever le signal à des instants multiples de T_e et ensuite convertir les échantillons sous forme binaire (8, 12, 16 bits, …) par l'intermédiaire d'un convertisseur analogique-numérique (**CAN**). Mais cette conversion n'est pas instantanée.

Donc il est nécessaire de procéder au blocage du signal pour avoir une conversion sans erreur si le signal à convertir varie trop rapidement. C'est pourquoi on utilise un échantillonneur-bloqueur puisqu'il mémorise le signal à convertir et le maintient constant pendant toute la durée de conversion.

L'effet de blocage peut être modélisé par une fonction porte décalée de λ/2 :

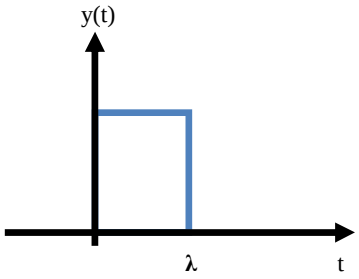

Figure 2.6 : **Fonction porte**

L'expression de cette fonction porte est définie comme suit :

$$y(t) = \sum_{k\to-\infty}^{+\infty} \text{rect}\left[\frac{t-\frac{\lambda}{2}-kTe}{\lambda}\right] = \text{rect}\left[\frac{t-\frac{\lambda}{2}}{\lambda}\right] * \sum_{k\to-\infty}^{+\infty} \delta(t-kTe)$$

L'échantillonnage-blocage consiste donc à la multiplication du signal par y(t). D'où la transformée de Fourier du signal échantillonné est défini de la manière suivante :

$$\mathbf{Se(f) = \frac{\lambda}{Te}\ sinc(\lambda f)\ \sum_{k\to-\infty}^{+\infty} S(f-kfe)\ e^{-j\pi f\lambda}}$$

2.3 Quantification

2.3.1 Définitions

La quantification consiste à associer à une valeur réelle **x** quelconque, une autre valeur $\mathbf{x_q}$ appartenant à un ensemble fini de valeurs et ce suivant une certaine loi : arrondi supérieur et arrondi le plus proche **[7]**.

L'écart entre chaque valeur $\mathbf{x_q}$ est appelé pas de quantification et le fait d'arrondir la valeur de départ entraine forcément une erreur de quantification que l'on appelle le bruit de quantification.

2.3.2 Quantification uniforme

La loi de quantification uniforme ou linéaire utilise un pas de quantification Δ constant entre chaque valeur $\mathbf{x_q}$.

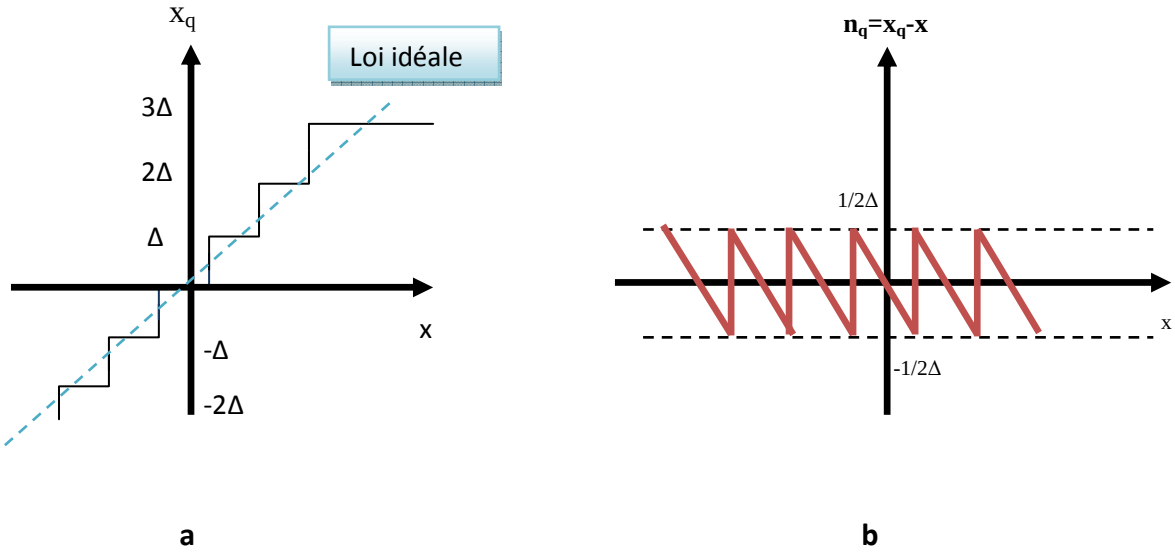

Figure 2.7 : **Quantification uniforme, a : signal quantifié, b :bruit de quantification**

L'évolution de bruit de quantification est une évolution en dent de scie avec une amplitude égale au quantum. Les caractéristiques du bruit de quantification varient en fonction de principe de quantification utilisée. Le bruit de quantification $\mathbf{n_q}$ est un signal aléatoire.

2.3.3 Quantification non linéaire

L'intérêt des techniques de quantification non linéaires réside dans le fait qu'elles permettent de coder de manière plus précise les valeurs qui apparaissent plus souvent. Dans la plupart des applications, on utilise la technique de quantification linéaire centrée mais

dans certains cas particuliers, il peut s'avérer intéressant d'opter pour une technique plus adaptée.

2.4 Codage

Le codage consiste à associer à un ensemble de valeurs discrètes un code composé d'éléments binaires. Le codage est la dernière étape et un outil très important à la numérisation d'un signal analogique

Les codes les plus connus et les plus utilisés sont : code binaire naturel, code binaire décalé, code complément à 2, code DCB, code Gray.

2.4.1 Code binaire naturel ou DCBN

Il est défini de la manière suivante :

$$N = \sum a_i \ 2^i$$

Avec $a_i = \{0,1\}$ et **i** ϵ IN.

Ce code se prête parfaitement au traitement des opérations arithmétiques. Ses inconvénients sont les suivants :

Il faut un grand nombre de bits pour exprimer un nombre dès que celui-ci est élevé.

Ce code peut introduire des erreurs lors du codage des grandeurs variant de façon ordonnée. En effet entre deux mots successifs de ce code, plusieurs bits pourront être amenés à changer simultanément **[8]**.

2.4.2 Code Gray

Le passage d'un mot code au suivant se fait par un changement d'état par un seul bit à partir de **LSB** (Least Significant Bit) si possible **[8]**.

Il est obtenu à partir d'un code binaire :

$$G_i = B_{i+1} \oplus B_i$$

Caractéristiques

- C'est un code cyclique, en effet, lorsqu'on passe du dernier au permier le principe est toujours vérifié.
- Ce code a la même densité que le code binaire naturel.
- Il existe des symétries dans le contribution des mots du code, d'où le nom du code binaire réfléchis ou code reflexe.
- Le code Gray n'est pas un code pondéré.
- Il y a une correspondance entre le code Gray et la table de Karnaugh.

2.4.3 Code DCB

Chaque élément d'un nombre décimal (chiffre décimal) est représenté par son équivalent binaire naturel à 4 bits.

Exemple :

1 9 8 7 => 0001 1001 1000 0111

Caractéristiques

- C'est un code pondéré.
- Il conserve les avantages du système décimal et du code binaire pur.
- Les mots codes sont plus longs qu'en code BCDN **[8]**.

Chapitre III : VERIFICATION DU LOCUTEUR

3.1 Reconnaissance du locuteur

La reconnaissance d'un locuteur ou la reconnaissance sur la base de la voix est une motivation ancienne. Elle a pour objectif la détermination de l'identité d'un locuteur ou d'identifier une personne à partir de sa voix, plus précisément à partir d'un signal de parole. Pour cela, on doit tester si la voix enregistrée provient vraiment d'un locuteur particulier ou non.

3.1.1 Conditions nécessaires :

Pour bien assurer une bonne qualité, acceptable et robuste du système de reconnaissance du locuteur, les caractéristiques suivantes sont nécessaires :

- Les locuteurs ne doivent pas déguiser leur voix. C'est-à-dire lors de l'enregistrement de la parole, ils doivent être en bonne santé et être concentrés.
- Pas de stress durant l'enregistrement.
- L'environnement doit être bien contrôlé. C'est-à-dire :
  - le micro d'enregistrement doit être le même pour tous les locuteurs.
  - L'endroit où l'enregistrement se déroule est obligatoirement un même local et bien calme.
- Des données de parole, enregistrées dans les mêmes conditions que le signal test, sont disponibles pour référencier un locuteur dans le système.
- Le contenu linguistique des messages inclut des mots connus du système, permettant à celui-ci de calculer une ressemblance entre voix en se basant sur des contenus comparables.
- L'usage d'un système de synthèse de la parole n'est pas autorisé.

3.1.2 Principe de base

Figure 3.1 : **Structure d'un système de reconnaissance du locuteur**

3.1.3 Modes de reconnaissance du locuteur

En reconnaissance du locuteur, il existe 2 modes tels que : le mode dépendant du texte et le mode indépendant du texte.

a. Mode dépendant du texte

En mode dépendant du texte, le texte prononcé par le locuteur durant l'enregistrement est le même que celui qu'il a prononcé lors de l'apprentissage ou de la vérification de sa voix. Il existe 3 niveaux de dépendance au texte et ils sont classés suivant les applications :

- système à texte libre
- système à texte suggérée
- système dépendant du vocabulaire ou système personnalisé dépendants du texte.

b. Mode indépendant du texte

En mode indépendant du texte, le locuteur peut prononcer n'importe quelle phrase pour être reconnu. L'avantage dans ce mode c'est qu'on ne trouve aucune contrainte sur le message que le locuteur doit prononcer ni sur la langue qu'il peut utiliser.

3.2 Vérification du locuteur

La vérification du locuteur consiste à vérifier l'identité proclamée par un individu par la comparaison d'un signal vocal et d'un modèle de référence du locuteur présumé, préalablement appris par le système, ou encore elle consiste à déterminer si un locuteur est bien celui qu'il prétend être.

Le principe de la vérification est simple, on cherche à trancher entre deux hypothèses bien distinctes : soit le locuteur est bien le locuteur autorisé, c'est à dire celui dont l'identité est revendiquée, soit le locuteur est un imposteur, c'est à dire qu'on a deux entrées : une identité et un accès de test **[6]**.

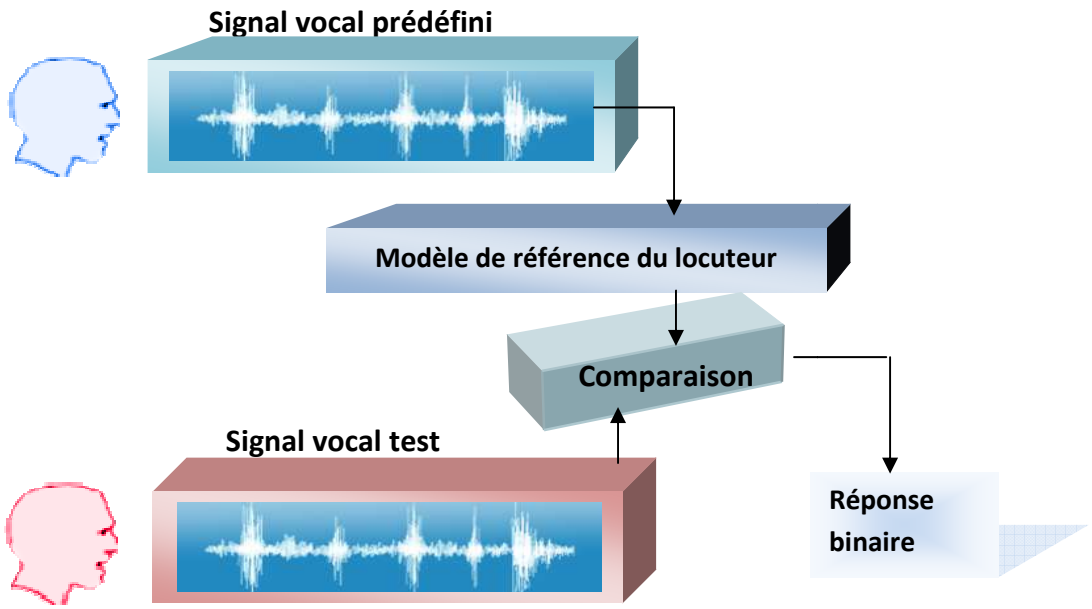

Figure 3.2 : **Principe de vérification du locuteur**

3.3 Procédure de vérification:

La vérification du locuteur est divisée en trois grandes principales phases : la phase de paramétrisation ou l'analyse acoustique, la phase de modélisation et la phase de décision.

Ici dans notre cas le locuteur a prononcé le même message (durant l'enregistrement et le test) ou les mêmes mots, c'est-à-dire que notre système de reconnaissance de locuteur est en mode dépendant du texte.

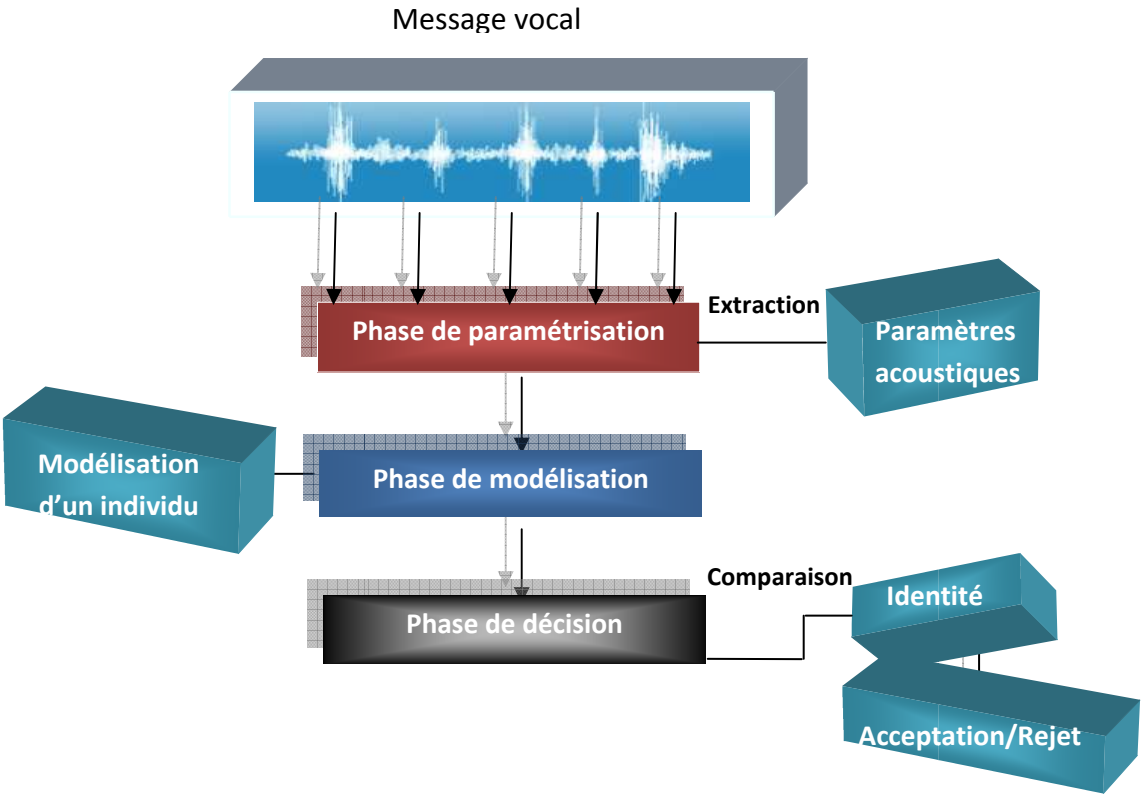

Figure 3.3 : **Différentes phases de vérification du locuteur**

3.3.1 Phase de paramétrisation

Les systèmes de reconnaissance du locuteur utilisent des représentations du signal de parole dans lesquelles le bruit et la redondance ont été réduits afin de ne conserver que les informations considérées comme utiles à la tâche spécifiée. Cette phase est appelée la phase de paramétrisation ou paramétrisation du signal de parole. C'est la phase la plus importante en reconnaissance du locuteur. Dans cette phase, les paramètres à identifier doivent être fréquents, facilement mesurables, pas trop sensibles à la variabilité intra-locuteur et robustes face aux imitateurs.

Pour pouvoir obtenir ces paramètres, la méthode la plus sûre et la plus utilisée est la méthode des **MFCC**s (Mel-frequency Cepstral Coefficients) qui seront calculés à l'aide de **TFD** (Transformée de Fourier discrète).

a. Transformée de Fourier discrète

La transformée de Fourier discrète ou **TFD** (Transformée de Fourier Discrète) permet de calculer la transformée de Fourier d'une suite d'échantillons au lieu d'une fonction continue.

On appelle la transformée de Fourier discrète d'une séquence x_n, où n=0, 1, …, N-1, la séquence X_k **[9]**. Elle est définie par:

$$X_k = \sum_{n=0}^{N-1} x_n \; e^{-\frac{2\pi i n k}{N}}$$

Où (x_n) est la suite des échantillons temporels, (X_k) la suite des échantillons fréquentiels pour k= [0, 1, 2,…, N-1], et N le nombre d'échantillons temporels.

b. FFT

La **FFT** (Fast Fourier Transform) est un algorithme rapide de **TFD** (Transformée de Fourier Discrète). Elle est très utilisée dans le traitement numérique du signal sonore. Grâce à elle on peut isoler les différentes fréquences qui composent tout son constitué par la superposition de plusieurs ondes sinusoïdales.

Le but de l'algorithme de Transformée de Fourier rapide est de calculer la **TFD** avec une complexité minimale ou d'évaluer les intégrales de Fourier en minimisant le nombre d'opération de façon à obtenir un algorithme rapide. L'algorithme le plus utilisé est celui de **Cooley-Tuckey [10]**.

Algorithme de Cooley-Tuckey :

En posant

$$X[k] = \sum_{n=0}^{N-1} x[n] \; .W_N^{nk} \qquad \text{avec} \qquad W_N = e^{-\frac{2\pi i}{N}}$$

Propriétés de W_N :

$$W_N^{2nk} = e^{-\frac{2\pi i nk}{N/2}} = W_{N/2}^{nk}$$

$$W_N^{nk+\frac{N}{2}} = e^{-\frac{2\pi i(nk+N2)}{N}} = -W_N^{nk}$$

Ces formules sont valables à condition qu'on ait un nombre d'échantillons puissance de 2, c'est-à-dire $N=2^m$ où m=1,2,3,…

Et les indices paires et impaires sont :

$$\begin{cases} x1[n] = x[2n] \\ x2[n] = x[2n+1] \end{cases}$$

Ainsi l'algorithme de **Cooley-Tuckey** est défini comme suit :

Pour $0 \leq k \leq \frac{N}{2}-1$

$$\begin{cases} X[k] = X1[k] + W_N^k . X2[k] \\ X[k+\frac{N}{2}] = X1[k] - W_N^k . X2[k] \end{cases}$$

c. MFCC

Les coefficients Mel Cepstre ou **MFCC**s (Mel-frequency Cepstral Coefficients) servent à déterminer les paramètres de l'analyse spectrale. Ces coefficients font partie des paramètres les plus couramment utilisés en reconnaissance du locuteur.

Ils fournissent une représentation acoustique du signal vocal qui est adaptée à la reconnaissance vocale. Aussi, les **MFCC**s caractérisent bien la forme du spectre et permettent de séparer l'influence de la source du signal vocal de celle du conduit vocal **[11]**.

Les MFCCs sont définis de la manière suivante :

$$\mathbf{MFCC_i} = \sum_{k=0}^{N} X[k] \ .\cos\left[i\left(k-\frac{1}{2}\right)\frac{\pi}{N}\right] \ ; \ i = [1,\ldots, M]$$

Où **M** le nombre de coefficients **MFCC**, typiquement de l'ordre de 10 à 20 pour la voix, **X[k]** les énergies du signal analysé pour **k= [1,…, N]** et **N** est le nombre d'échantillonnage.

3.3.2 Phase de modélisation

La phase de modélisation est la modélisation acoustique du locuteur ou généralement la modélisation générative du locuteur dont le but est d'estimer la distribution qui a pu générer les vecteurs cepstraux du signal d'apprentissage.

C'est dans cette phase qu'on peut bien distinguer les caractéristiques spectrales des locuteurs. Généralement les systèmes de reconnaissance du locuteur utilisent pour la plupart des algorithmes de comparaison de motifs. La technique de modélisation la plus prometteuse est la technique à base du modèle **GMM (**Gaussian Mixture Model**)** ou

Modèles à mélanges de gaussiens maximisant la vraisemblance des données d'apprentissage, car ce modèle est capable de capturer les points communs entre différentes représentations de motifs spectraux issus du même locuteur.

GMM

En reconnaissance du locuteur, on modélise souvent ce dernier comme une source pouvant avoir plusieurs comportements gaussiens. Le mélange de gaussiennes est un modèle basé sur l'algorithme **EM** (Expectation-Maximisation). L'objectif de ce modèle est d'estimer les densités de probabilités des classes acoustiques afin d'avoir une meilleure approximation de la distribution correspondant au locuteur. La densité de probabilité pour une mixture de gaussiennes à **N** composantes pour une variable aléatoire **x** s'exprime sous la forme suivante **[12]**:

$$p(x|\Theta) = \sum_{i=1}^{N} \gamma_i N(x ; \mu_i, \Sigma_i)$$

Où $N(x ; \mu_i, \Sigma_i)$ est la loi gaussienne de moyenne μ et de variance **Σ**, $\Theta = [\mu, \Sigma, \gamma]^T$ est le vecteur de paramètre global du **GMM** et **γ** est le vecteur de poids de la mixture.

Cette densité de probabilité d'une mixture de gaussiennes sert à calculer la vraisemblance qui est une méthode très utile à la reconnaissance des caractères acoustiques de chacun des locuteurs afin de les identifier. La vraisemblance pour un gaussien multidimensionnel est définie de la manière suivante :

$$l(x|\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right]$$

Où **l(x|μ,Σ)=log p(x|μ,Σ)** et **d** la dimension de **x.**

Les paramètres de ce modèle **GMM** obtenus à partir de l'algorithme **EM** sont définis de la manière suivante :

- **Moyenne pondérée des données**

$$\mu_j = \left[\sum_{n=1}^{N} \gamma_j(x_n)\, x_n \right] \left[\sum_{n=1}^{N} \gamma_j(x_n) \right]^{-1}$$

- **Covariances [13]**

$$\Sigma_j = \left[\sum_{n=1}^{N} \gamma_j(x_n)\, (x_n - \mu_j)\, (x_n - \mu_j)^T \right] \left[\sum_{n=1}^{N} \gamma_j(x_n) \right]^{-1}$$

- **Vecteur de poids de la mixture**

$$\gamma_j(x_n) = \pi_j N(x_n|\mu_j, \Sigma_j) \left[\sum_k \Sigma_k \pi_k N(x_n|\mu_k, \Sigma_k) \right]^{-1}$$

, avec

$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(x_n)$$

3.3.3 Phase de décision

La phase de décision est la dernière étape en vérification du locuteur qui désigne ce dernier finalement reconnu. Dans tous les systèmes de reconnaissance du locuteur il faut, à un moment ou à un autre, prendre la décision d'accepter ou de rejeter un segment de parole comme appartenant au client dont on cherche à vérifier l'identité. La stratégie mise en jeu de cette phase dépendra fortement de la phase de modélisation. Cette stratégie est basée sur un test d'hypothèse.

En considérant par exemple un segment de parole **X** et une identité proclamée **S**.

Le test d'hypothèse est défini de la manière suivante :

– H0 : **X** provient du locuteur dont l'identité **S** a été proclamée.

– H1 : **X** ne provient pas du locuteur dont l'identité **S** a été proclamée.

Ensuite, en estimant le rapport de vraisemblance entre ces deux hypothèses, on peut comparer ce rapport à un seuil de décision :

Si $\frac{p(X/H0)}{p(X/H1)}$ < Seuil alors H1 est accepté

Si $\frac{p(X/H0)}{p(X/H1)}$ > Seuil alors H0 est accepté

Ce qui implique finalement que : H1 est accepté lorsque **X** a été généré par un locuteur imposteur et H0 accepté lorsque le segment de parole **X** a été bien généré par un locuteur client.

Chapitre IV : REALISATION DU LOGICIEL « RecSpeaker version 1.0 »

4.1 Introduction

« **RecSpeaker version 1.0** » est un logiciel conçu dans le but démonstratif. Il traite informatiquement la reconnaissance d'une voix humaine ou la reconnaissance d'un locuteur. En effet, il implémente le traitement du signal numérique dont l'objectif est de distinguer les caractéristiques vocales de chaque individu. De plus, ce logiciel intègre une application pour sécuriser vocalement n'importe quel type de fichier, et il sert aussi la visualisation des spectres et le traitement des fichiers audio.

4.2 Description du logiciel « RecSpeaker Version 1.0»

4.2.1 Programmation

Les programmes du logiciel « **RecSpeaker version 1.0** » ont été écrits en langage C avec Visual Studio 2005. Il utilise :

- L'**API** Windows pour l'interface utilisateur.
- La base de données Access pour stocker les locuteurs.
- Fmod et Windows Multimedia pour l'interface avec les périphériques audio et les fichiers audio.
- L'algorithme de Cooley-Tukey pour le calcul de la **FFT**.
- L'algorithme de Gauss pour le calcul de **GMM**.
- Les syntaxes **Sql** pour la connexion à la base de données.

4.2.2 Présentations des interfaces

a. Interface principale

L'interface principale illustrée par la fig4.1 a été réalisée avec l'**API Windows** en langage C. Elle a 6 boutons différents tels que : **A propos**, **Application**, **Visualisation**, **Enregistrer locuteur**, **RecSpeaker** et **Quitter**.

Figure 4.1 : **Interface principale du « RecSpeaker 1.0»**

b. Fonction des boutons

i. Bouton « A propos »

Ce bouton affiche la version du « **RecSpeaker** ».

Figure 4.2 : **Effet du bouton « A propos»**

ii. Bouton « RecSpeaker »

Le bouton **« RecSpeaker »** donne la boite de dialogue montrée par la fig 4.3. Cette boite de dialogue contenant 9 boutons différents, sert à identifier une voix d'un individu au format **wave** seulement**.** Mais, elle peut aussi lire des fichiers audio comme **mp3** et **wma**.

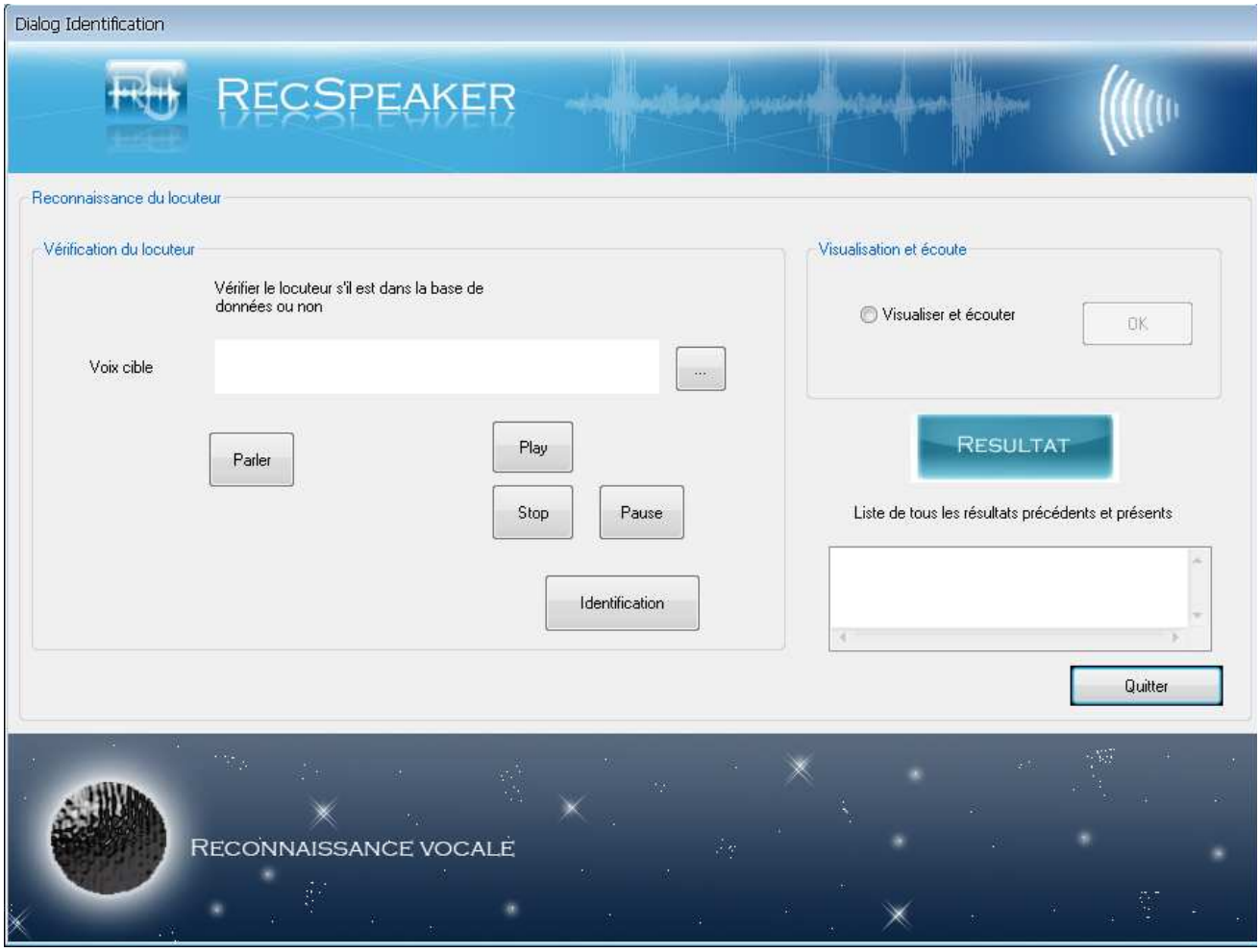

Figure 4.3 : **Boite de dialogue pour l'identification du locuteur**

- ***Bouton « parcourir »*** 

  Apres avoir cliqué sur ce bouton, la boite de dialogue de navigation permettant de choisir le fichier **wave, mp3** et **wma** de la fig 4.4 apparait. Si on veut identifier un locuteur, il faudra choisir un fichier **wave** seulement. Mais pour la lecture on peut choisir un fichier **mp3** ou **wave** ou **wma**.

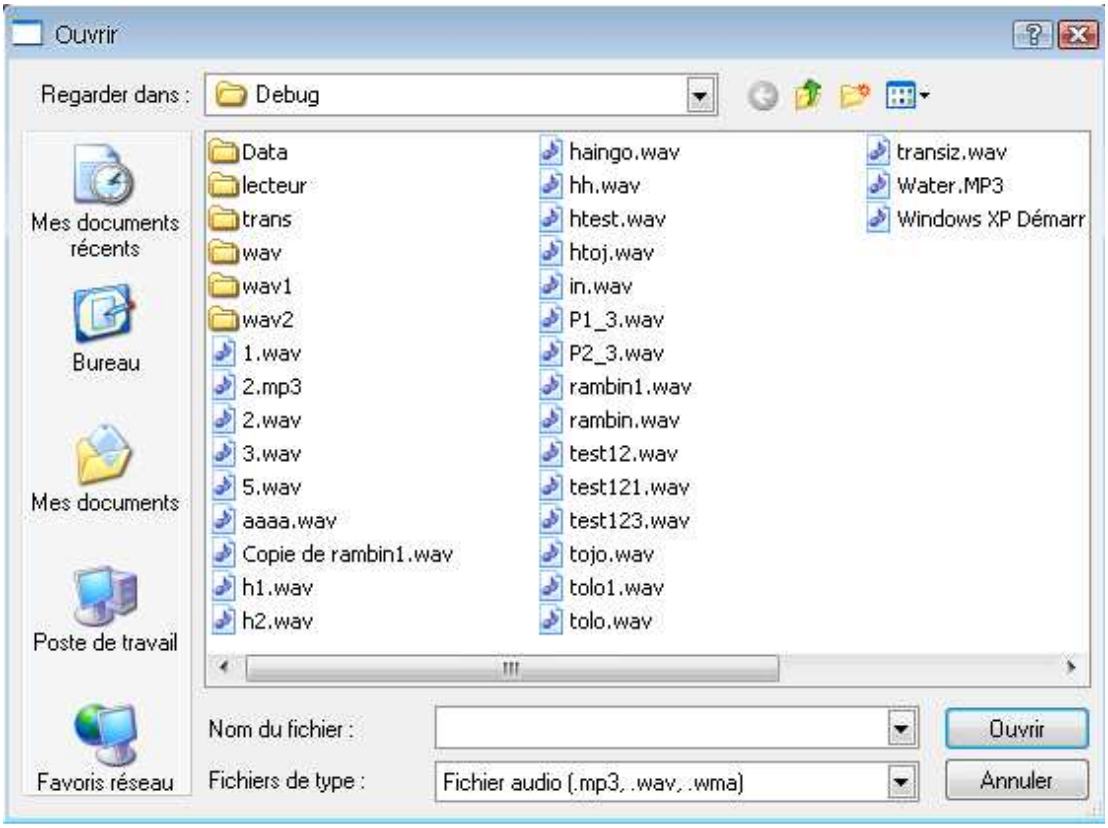

Figure 4.4 : **Boite de dialogue de navigation**

– ***Bouton « Play »*** 

Ce bouton sert pour la lecture des fichiers audio aux formats **wave**, **mp3** et **wma**.

– ***Bouton « Stop »*** 

Si on veut arrêter la lecture, on clique sur ce bouton.

– ***Bouton « Pause »*** 

Ce bouton permet de faire la pause d'une lecture encours.

– ***Bouton « Parler »*** 

Ce bouton ouvre une autre boite dialogue illustrée par la fig 4.5 pour l'enregistreur vocal.

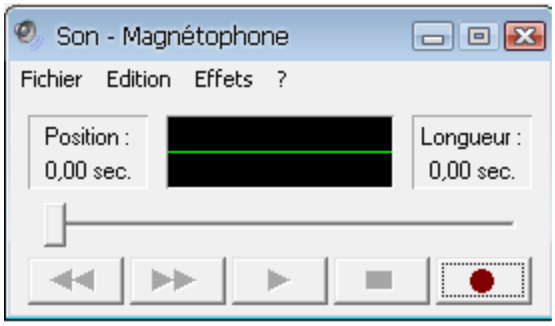

Figure 4.5 : **Magnétophone**

– ***Bouton « Identification »*** 

Après avoir ouvert un fichier au format **wave** contenant les caractéristiques vocales d'un locuteur dans la boite de dialogue de navigation, la zone de texte de la fig 4.3 contient le chemin complet du fichier. Ensuite, il suffit d'appuyer sur le bouton ***« Identification »*** pour vérifier si ces caractéristiques ressemblent à celles qui sont déjà préenregistrées.

– ***Bouton « Quitter »*** 

Pour quitter l'application, on clique sur ce bouton.

– ***Bouton radio « Visualiser et écouter »*** 

Si on veut visualiser des fichiers audio tels que **mp3** et **wav**, il suffit d'appuyer sur ce bouton radio et le bouton ***« ok »*** sera activé.

iii. Bouton « Visualisation »

On peut visualiser à l'aide de la boite de dialogue de la fig 4.6, le spectre d'un signal sonore. Elle est composée de 4 boutons tels que : ***« WaveEditor »***, ***« VisualSpectro »***, ***« WavMaker »*** et ***« WavVisualisation ».***

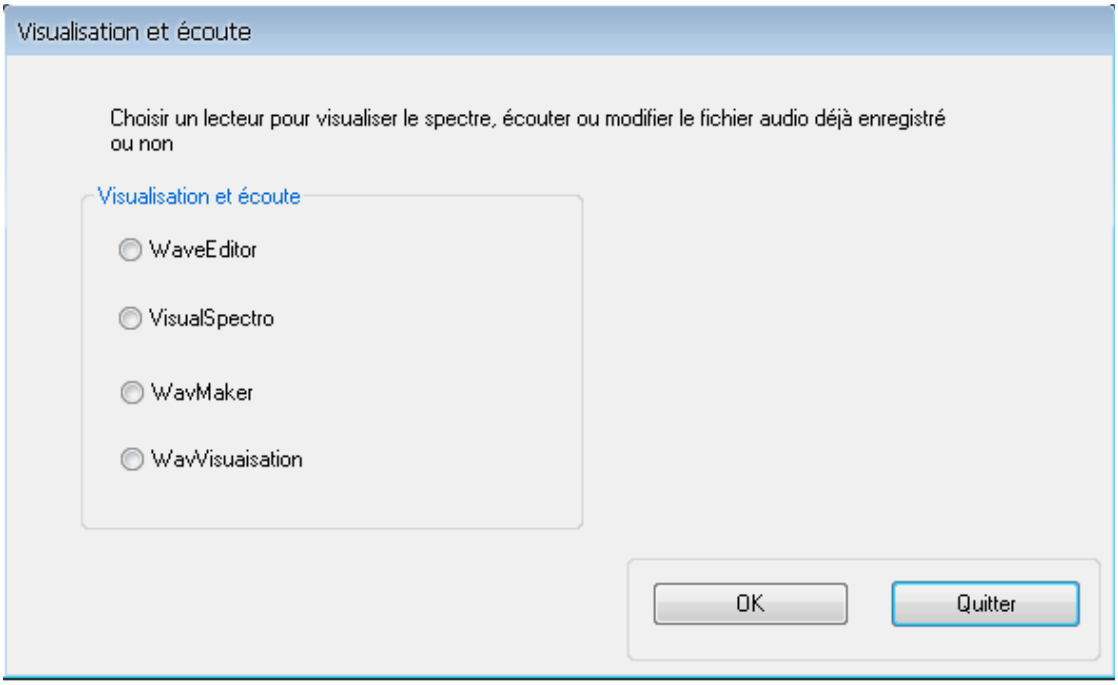

Figure 4.6 : **Boite de dialogue de visualisation spectrale**

- ***Bouton radio « WaveEditor »***

RecSpeaker fait aussi appel à ***WaveEditor1.0*** pour la visualisation d'un signal en fonction du temps. Le fichier doit être au format **wave**.

Sphinx WaveEditor version1.0 est un logiciel à open source.

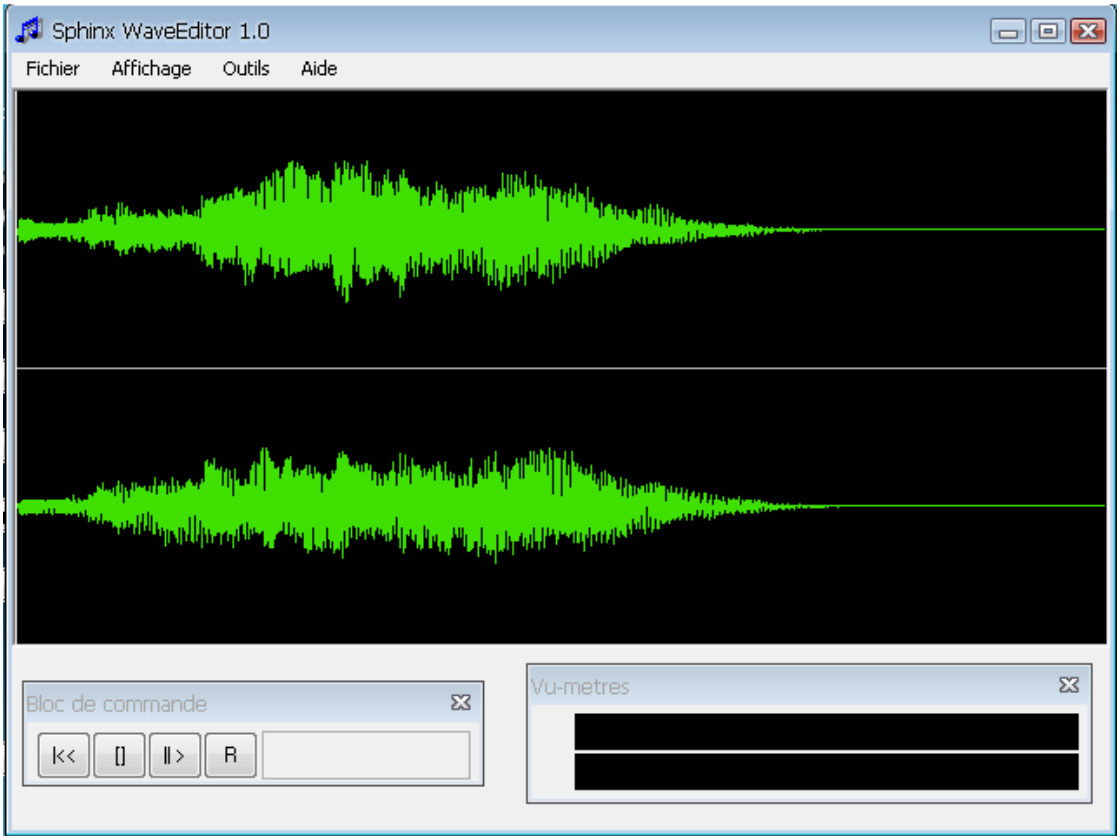

Figure 4.7 : **WaveEditor 1.0**

Après avoir cliqué sur le bouton ***« VisualSpectro »*** et ensuite sur ***« ok »***, l'interface de « ***VisualSpectro »*** de la fig 4.8 apparait. Il est, non seulement un outil de visualisation spectrale des signaux vocaux, mais aussi un lecteur des fichiers audio (**mp3,wma et wave**).

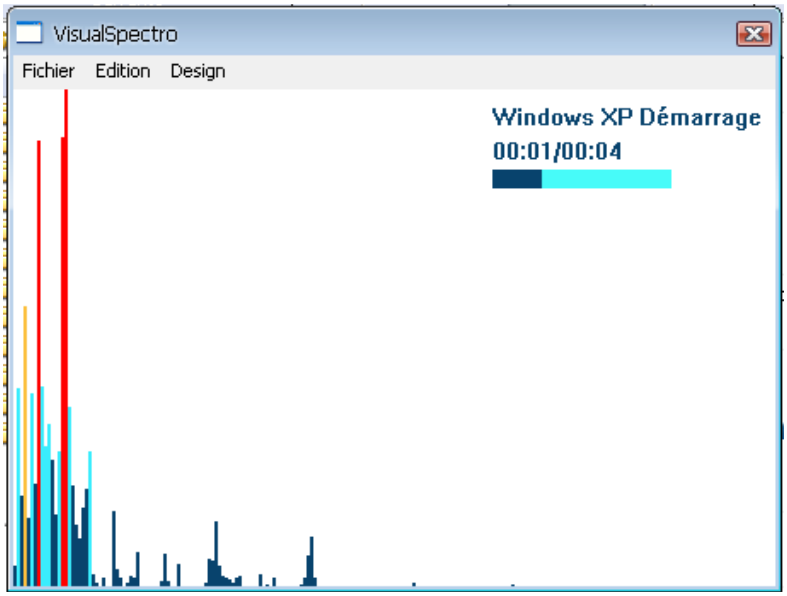

Figure 4.8 : **VisualSpectro**

- ***Bouton « Wavemaker »***

  Ce bouton fait apparaitre l'éditeur de fichier wave : ***« Makewav »*** qui permet de changer les caractéristiques de ce dernier.

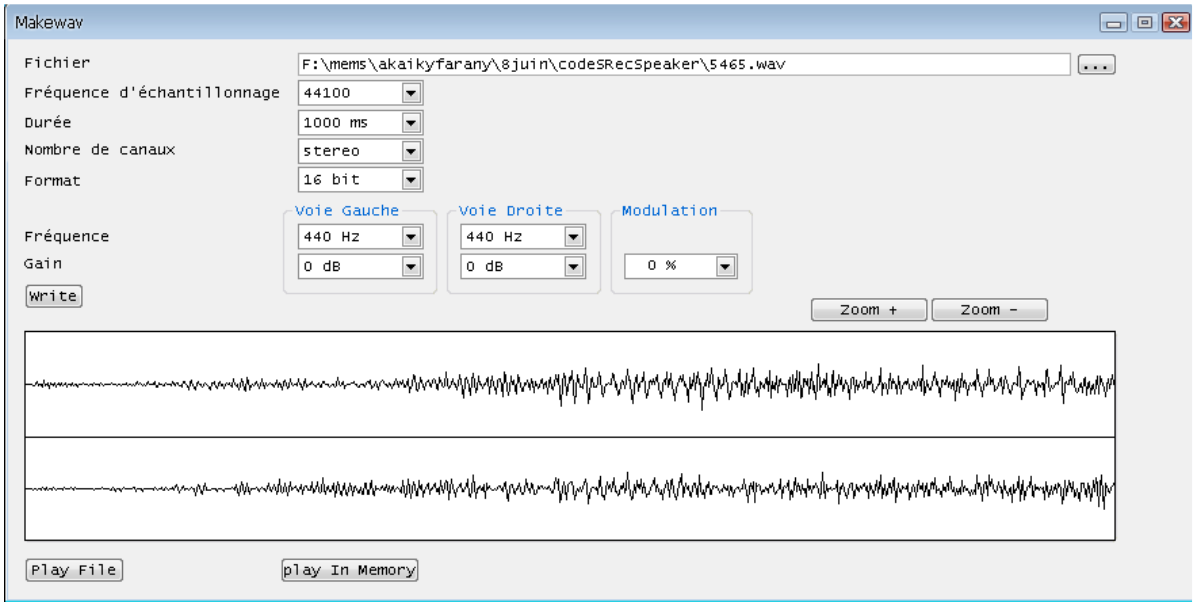

Figure 4.9 : **Makewav**

- ***Bouton « WavVisualisation »***

La boîte de dialogue de visualisation d'un fichier wave de la fig 4.10 apparaîtra si on clique sur ce bouton.

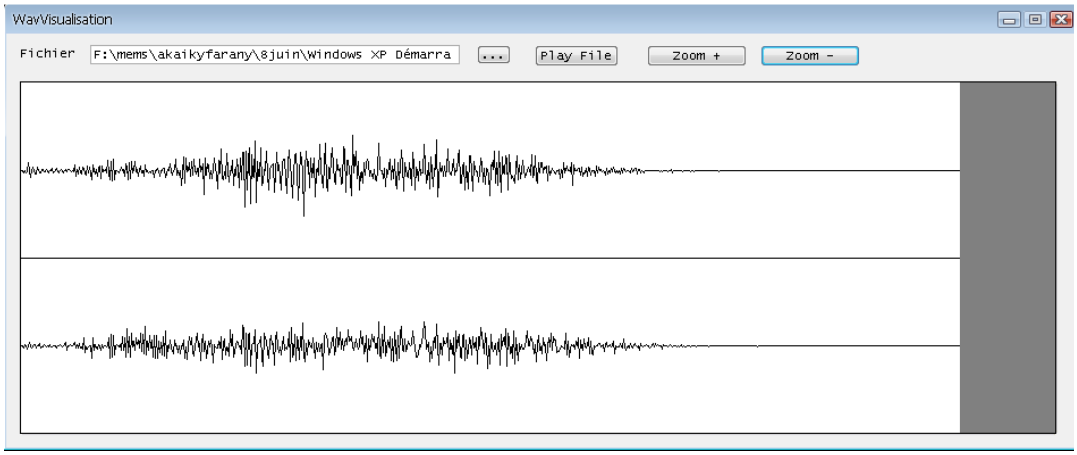

Figure 4.10 : **Visualisation d'un signal audio au format wav**

iv. Bouton « Enregistrer locuteur »

C'est ici qu'on devra faire l'enregistrement, la suppression et la modification des locuteurs. La Fig 4.11 nous montre la boite de dialogue de la mise à jour des données.

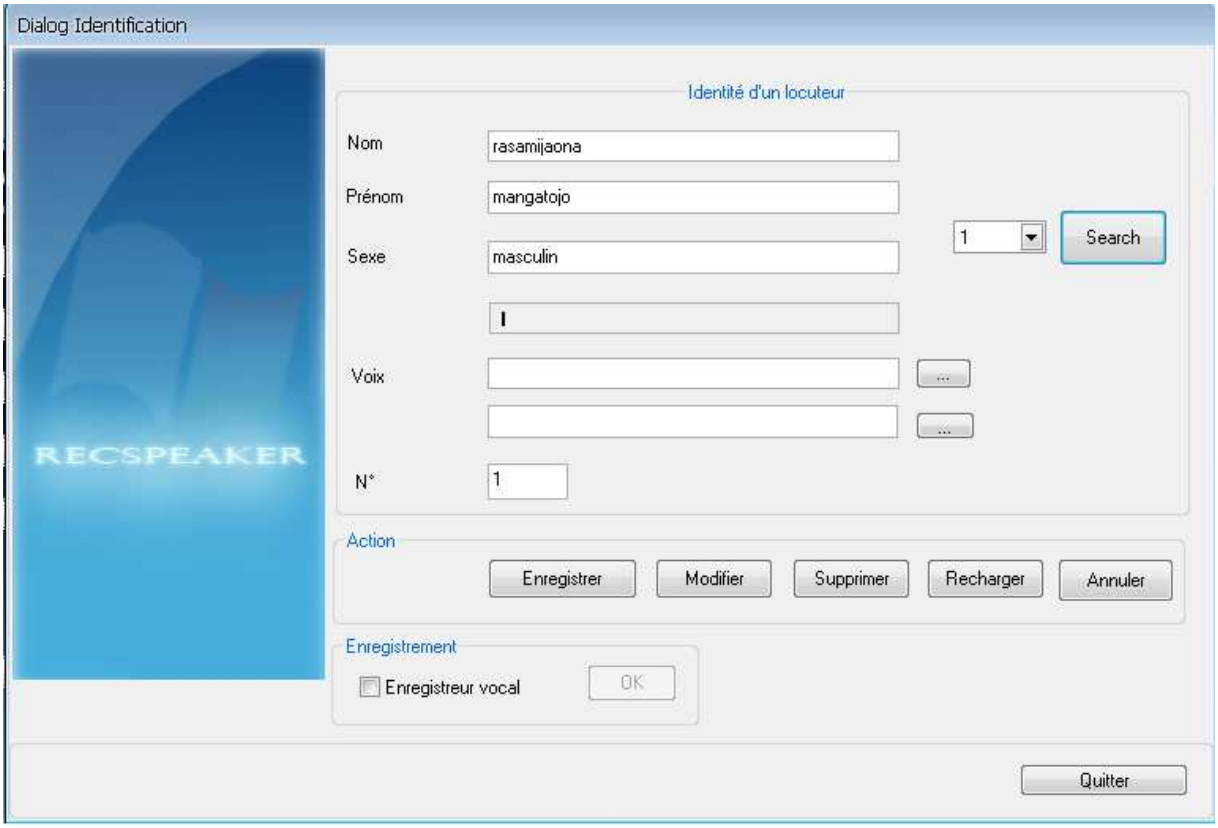

Figure 4.11 : **Mise à jour des données**

v. Bouton « Application »

La boite de dialogue illustrée par la fig 4.12 nous montre une application de la reconnaissance du locuteur. Cette application consiste à crypter un fichier quelconque du locuteur déjà enregistré. Et lors du décryptage, il suffit d'entrer la voix clé du locuteur.

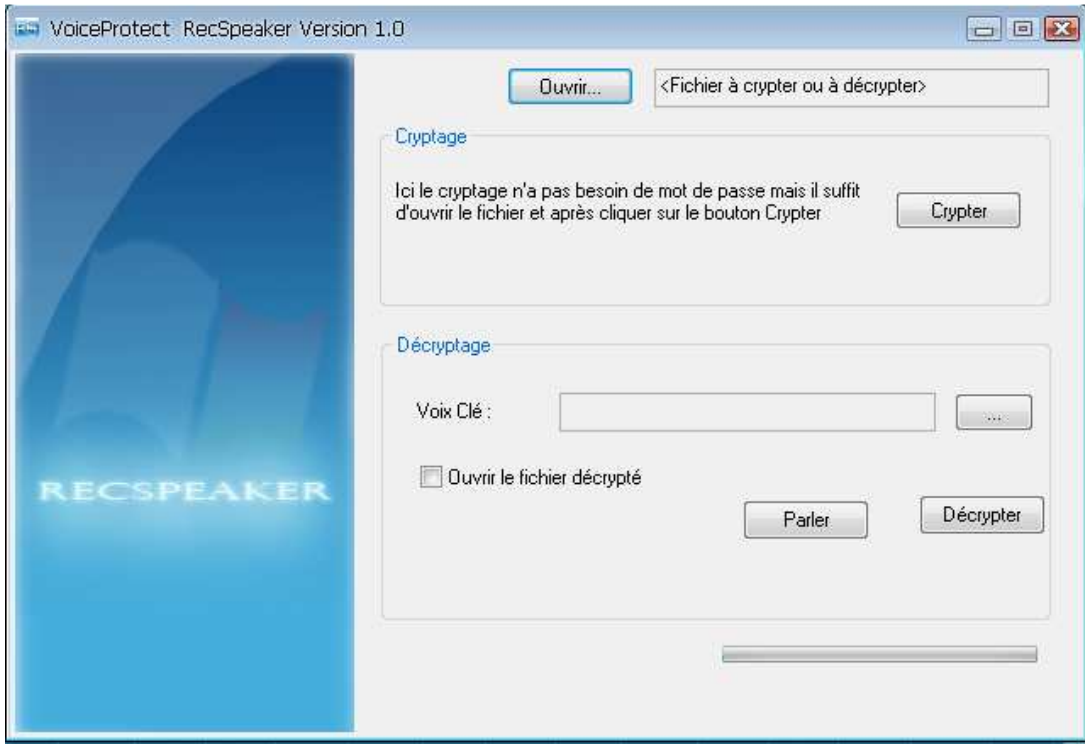

Figure 4.12 : ***VoiceProtect***

CONCLUSION

En matière de sécurité et de surveillance, la biométrie vocale ou la reconnaissance vocale offre beaucoup d'avantages par rapport aux autres biométries. Elle est la seule caractéristique biométrique permettant : de vérifier à distance l'identité d'une personne et de sécuriser tout contrôle d'accès non seulement d'un système informatique, mais aussi dans plusieurs domaines (télécommunication, commerce, etc).

Cet ouvrage explique brièvement les étapes à suivre pour concevoir un système de reconnaissance vocale et un système de vérification d'un locuteur par un ordinateur via le logiciel « ***RecSpeaker version 1.0*** » en utilisant tout simplement un équipement (microphone) pour capter le signal vocal d'un locuteur. Quant à ce logiciel, il identifie quelqu'un par sa voix et sert à protéger un fichier quelconque en cryptant ce dernier et le décryptant vocalement.

Pourtant la performance d'un système de reconnaissance vocale dépend fortement de caractères physiologiques et comportementaux. Apparemment, la qualité d'un signal vocal est en fonction de la variabilité de la voix du locuteur dans le temps comme dans le cas de maladie, des états émotionnels et de l'âge, des conditions d'acquisition de la voix telles que le bruit, de la qualité des équipements comme le microphone, et le fait que différentes personnes peuvent avoir des voix similaires. Voilà pourquoi le système de reconnaissance vocale est vraiment un système très sensible.

Beaucoup de ces conditions précédentes ne sont pas encore faites dans la plupart des logiciels de reconnaissance vocale de notre époque y compris « ***RecSpeaker version 1.0*** ». Ce qui nous oblige encore à terminer les tâches restants pour bien améliorer ce logiciel dans la version suivante.

ANNEXES

Annexe A : LA PROGRAMMATION SOUS WINDOWS

A.1 Fenêtre principale

a. Introduction

Windows est un système d'exploitation proposant une interface graphique. Dans ce premier article, nous allons créer une application composée seulement d'une fenêtre principale, dont le but et de montrer l'architecture d'un programme Windows **[14]**.

Figure A.1 : **Fenêtre principale**

b. Fonction WinMain

Le point d'entrée d'une application Windows est la fonction **WinMain**. C'est l'équivalent de la fonction main des applications classiques. Elle est appelée par le système d'exploitation au lancement du programme. Il lui fournit 4 paramètres.

```
int WINAPI WinMain(HINSTANCE hinstance, HINSTANCE hPrevInstance,
                                                LPSTR lpCmdLine, int nCmdShow);
```

attribue par le système d'exploitation qui lui permet de l'identifier. Le second paramètre est toujours **NULL** pour les applications Win32. Le troisième paramètre est un pointeur sur la ligne de commande.

c. Création de la fenêtre

La fonction qui permet de créer une fenêtre se nomme **CreateWindow** :

```
HWND CreateWindow(
    LPCTSTR lpClassName,  // Pointeur sur une classe de fenêtre.
    LPCTSTR lpWindowName,  // Pointeur sur le texte de la fenêtre.
    DWORD dwStyle,  // Style de la fenêtre.
    int x,  // Position horizontale de la fenêtre.
    int y,  // Position verticale de la fenêtre.
    int nWidth,  // Largeur de la fenêtre.
    int nHeight,  // Hauteur de la fenêtre.
    HWND hWndParent,  // Handle de la fenêtre parent.
    HMENU hMenu,  // Handle de menu ou ID de contrôle.
    HANDLE hInstance,  // Handle d'instance de l'application.
    LPVOID lpParam   // Pointeur sur des données passées à WM_CREATE.
   );
```

Le premier paramètre qu'elle reçoit est un pointeur sur une chaîne de caractères identifiant la classe de fenêtre. Les classes de fenêtre sont des modèles pour construire les fenêtres (le terme de classe n'a rien à voir avec les classes du C++). Si pour les contrôles standard nous avons des classes de fenêtres prédéfinies et globale, nous devrons en créer une pour la fenêtre principale. Nous devons pour cela remplir une structure de type WNDCLASS (définie dans winuser.h).

```
WNDCLASS wc;

wc.style = 0;
wc.lpfnWndProc = MainWndProc;
wc.cbClsExtra = 0;
wc.cbWndExtra = 0;
wc.hInstance = hinstance;
wc.hIcon = LoadIcon(NULL, IDI_APPLICATION);
wc.hCursor = LoadCursor(NULL, IDC_ARROW);
wc.hbrBackground = (HBRUSH)(1 + COLOR_BTNFACE);
wc.lpszMenuName =  NULL;
wc.lpszClassName = "MaWinClass";
```

d. Boucle de message

Pour communiquer avec une application ou ses diverses fenêtres, Windows leurs envoie des messages. Par exemple, si vous cliquez sur le bouton fermeture (en haut à droite de la fenêtre), Windows va créer un message approprié qu'il va envoyer dans la file d'attente de l'application. La file d'attente est un tampon où sont stockés les messages en attente de traitement. C'est à nous de coder l'extraction des messages de la file d'attente :

```
MSG msg;

while (GetMessage(&msg, NULL, 0, 0))
{
    TranslateMessage(&msg);
    DispatchMessage(&msg);
}
```

e. Procédure d'une fenêtre

Voici la procédure de fenêtre de notre fenêtre principale dont nous avions passé un pointeur à la classe de fenêtre au début de ce document (on passe ce pointeur à la classe de fenêtre car c'est nous qui la créons, mais le système d'exploitation qui l'appelle, il doit donc la localiser).

```
LRESULT CALLBACK MainWndProc(HWND, UINT, WPARAM, LPARAM);


LRESULT CALLBACK MainWndProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam)
{
    switch (uMsg)
    {
        case WM_CREATE:

            return 0;

        case WM_DESTROY:
            PostQuitMessage(0);
            return 0;

        default:
            return DefWindowProc(hwnd, uMsg, wParam, lParam);
    }
}
```

plus haut dans ce document, met fin à la boucle de messages et donc à l'application. Les

messages non traités doivent l'être par la fonction **DefWindowProc**. Fonction qui implémente le comportement par défaut d'une fenêtre.

A.2 Boite de dialogue

a. Script de ressources

Figure A.2 : **Boîte de dialogue personnalisée**

Les boîtes de dialogue personnalisées sont créées à partir de ressources :

```
DIALOG1 DIALOG
     60, 60, 160, 80
          STYLE WS_POPUP | WS_VISIBLE | WS_CAPTION | WS_SYSMENU
                                                    CAPTION "A propos"
BEGIN
    DEFPUSHBUTTON "Ok", IDOK, 56, 50, 42, 12
    ICON 2, -1, 20, 15, 32, 32
    LTEXT "Mon beau programme !", -1, 60, 18, 80, 10
END
```

La ressource est composée de son identificateur littéral, suivi de son type (DIALOG), suivie de ses propriétés (position, dimensions, style et titre), suivi de son contenu balisé par les mots BEGIN et END. Il est en général constitué de contrôles. Chaque contrôle est décrit par son type, suivi de son identification visuelle (texte pour le bouton et le contrôle texte, identificateur de ressource pour l'icône), suivi de sa constante numérique d'identification, puis de sa position dans la boîte de dialogue et enfin de ses dimensions (les largeurs et hauteurs des contrôles ne sont pas en pixel, mais dépendent de la police de caractères utilisée). L'icône et le le contrôle texte ont un identificateur à -1 car ils ne sont là que pour la

décoration, mais il faut tout de même leur mettre un identificateur. Celui du bouton est à IDOK que nous n'avons pourtant pas défini. IDOK est défini dans Windows, il est envoyé à la procédure de fenêtre de la boîte de dialogue quand on appuie sur la touche "Entrée". Est défini aussi l'identificateur IDCANCEL qui lui est envoyé quand on appuie sur la touche "Echap" ou que l'on tente de fermer la boîte de dialogue.

b. Appel de la boîte de dialogue

La boîte de dialogue étant maintenant dans les ressources, nous pouvons l'appeler afin de l'ouvrir. C'est le rôle de la fonction **DialogBox**. Son premier paramètre est le handle d'instance de l'application, le second, un pointeur sur la chaîne de caractères identifiant la ressource, le troisième est le handle de la fenêtre parent et le dernier un pointeur sur sa procédure de fenêtre.

```
switch (uMsg)
{
    case WM_COMMAND:
        if(LOWORD(wParam) == IDM_ABOUT)
                DialogBox(hinst, "DIALOG1" , hwnd, (DLGPROC)Dialog1Proc);
```

c. Procédure de fenêtre de la boîte de dialogue

Comme pour la procédure de fenêtre de la fenêtre principale c'est à vous de la définir. C'est Windows qui l'appellera quand elle aura un message.

```
LRESULT CALLBACK Dialog1Proc(HWND, UINT, WPARAM, LPARAM);

BOOL APIENTRY Dialog1Proc(HWND hDlg,UINT uMsg,WPARAM wParam,LPARAM lParam)
{
    switch (uMsg)
    {
      case WM_INITDIALOG:

         return TRUE;

      case WM_COMMAND:
         if (LOWORD(wParam) == IDCANCEL || LOWORD(wParam) == IDOK)
                {
                   EndDialog(hDlg,0);
                   return TRUE;
                }

      default:
         return FALSE;
    }
}
```

Elle est fort semblable à celle de la fenêtre principale. Si le message est traité, elle doit renvoyer TRUE sinon elle doit renvoyer FALSE. WM_INITDIALOG doit être intercepté et renvoyer TRUE afin que les commandes IDOK et IDCANCEL soit envoyées lors des appuis sur les touches correspondantes. Le message WM_INITDIALOG est envoyé par Windows après la création de la boîte de dialogue, mais avant qu'elle soit visible.

Annexe B : LA CRYPTOGRAPHIE

B.1 Objectifs

Traditionnellement, le but de la cryptographie est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle. Mais à présent, la cryptographie moderne s'attaque plus généralement aux problèmes de sécurité des communications. Ce but est d'offrir un certain nombre de services de sécurité comme la confidentialité, l'intégrité, l'authentification des données transmises et l'authentification d'un tiers.

B.2 Définition de quelques termes

- **Cryptologie**

La cryptologie est une science mathématique qui comporte deux branches qui sont : la cryptographie et la cryptanalyse.

- **Cryptographie**

La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle.

- **Chiffrement**

Le chiffrement est l'opération qui transforme un message compréhensible (texte en clair) en un message incompréhensible ou texte chiffré ou cryptogramme afin de le protéger.

- **Déchiffrement**

Le déchiffrement est l'inverse du chiffrement, qui est l'action permettant de reconstruire le texte en clair à partir du texte chiffré.

- **Cryptanalyse**

La cryptanalyse, à l'inverse, est l'étude des procédés cryptographiques dans le but de trouver des faiblesses et, en particulier, de pouvoir décrypter des textes chiffrés.

- **Décryptement**

Le décryptement et l'action consistant à retrouver le texte en clair sans connaître la clef de chiffrement **[15]**.

B.3 Algorithmes de chiffrement

Il existe deux différents types d'algorithmes :

- Algorithmes symétriques ou à clef secrète
- Algorithmes asymétriques ou à clef publique

a. Chiffrement symétrique

La cryptographie symétrique est la plus ancienne forme de chiffrement.

i. Principe

La clef de chiffrement doit être égale à la clef de déchiffrement, mais elle doit rester secrète et être connue par des tiers et d'eux seuls.

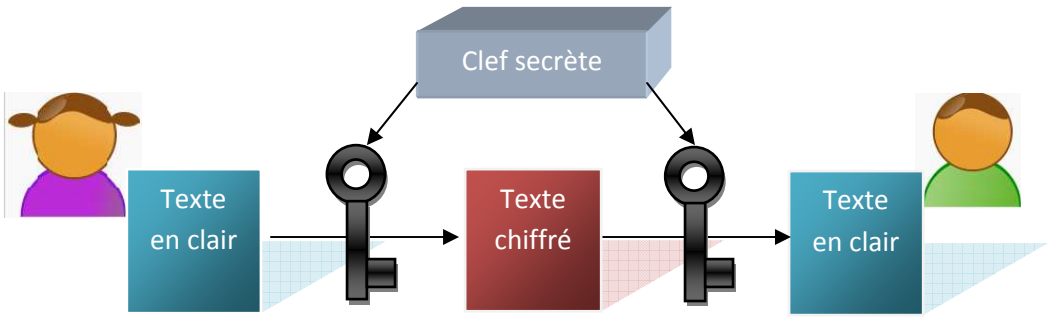

Figure B.1 : **Cryptographie symétrique**

ii. Algorithme de chiffrement en continu

- **RC4**

Historique

RC4 a été conçu par Ronald Rivest de RSA Security en 1987. Il est officiellement nommé Rivest Cipher 4, mais l'acronyme RC est aussi surnommé Ron's Code comme dans le cas de RC2, RC5 et RC6.

Les détails de RC4 furent initialement tenus secrets mais en septembre 1994, une description du chiffrement fut postée de manière anonyme sur la liste de diffusion Cypherpunks. Le message apparut ensuite sur le forum sci.crypt puis sur divers sites. L'algorithme avait vraisemblablement fait l'objet d'une rétro-ingénierie. Sur le plan légal, RC4 est une marque déposée dont les implémentations non officielles sont autorisées sous un autre nom que RC4, car l'algorithme n'a pas été breveté. La version non officielle de RC4 est aussi connue sous le nom de « ARCFOUR », « ARC4 » ou « Alleged RC4 » (signifiant « RC4 supposé » puisque RSA Security n'a jamais officiellement publié les spécifications de l'algorithme) **[15]**.

Principe général

RC4 est un algorithme de chiffrement à flot. Il fonctionne de la façon suivante : la clef **RC4** permet d'initialiser un tableau de 256 octets en répétant la clef autant de fois que nécessaire pour remplir le tableau. Par la suite, des opérations très simples sont effectuées : les octets sont déplacés dans le tableau, des additions sont effectuées, etc. Le but est de mélanger autant que possible le tableau. Finalement on obtient une suite de bits pseudo-aléatoires qui peuvent être utilisés pour chiffrer les données via un XOR.

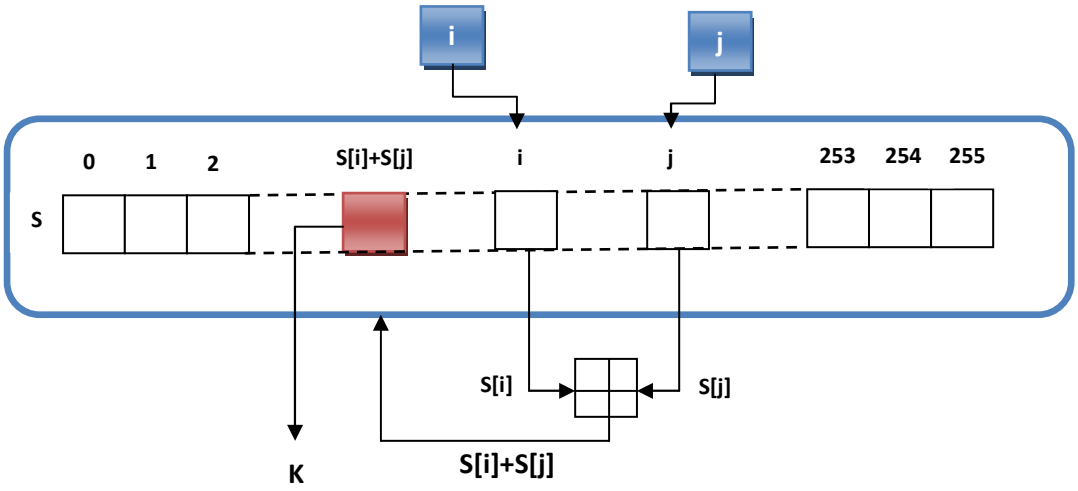

Figure B.2 : **Tour de RC4**

RC4 est un générateur de bits pseudo-aléatoires dont le résultat est combiné avec le texte en clair via une opération XOR, le déchiffrement se fait de la même manière.
Pour générer le flot de bits, l'algorithme dispose d'un état interne, tenu secret, qui comprend deux parties :

- une permutation **S** de tous les 256 octets possibles
- deux pointeurs **i** et **j** de 8 bits qui servent d'index dans un tableau

La permutation est initialisée grâce à la clé de taille variable, typiquement entre 40 et 256 bits, grâce au **key schedule** de RC4.

Génération de la permutation

La permutation **S** est initialisée grâce à la clé **K**. La longueur de la clé varie de 1 à 256 bits. En pratique, elle est souvent choisie de taille égale à 5 octets (pour 40 bits) ou 16 octets (pour 128 bits). La permutation se présente sous la forme d'un tableau de 256 entrées. Ses valeurs initiales correspondent à l'identité au sens mathématique (le premier octet est permuté avec le premier octet, etc).

iii. Algorithmes de chiffrement par blocs

- **DES**

Introduction

L'algorithme **DES** (Data Encryption Standard) a été créé dans les laboratoires de la firme IBM Corp. Il est devenu le standard du NIST en 1976 et a été adopté par le gouvernement en 1977. C'est un chiffrement qui transforme des blocs de **64 bits** avec une clé secrète de **56 bits** au moyen de permutations et de substitutions.
Le DES est considéré comme étant raisonnablement sécuritaire.
Il est officiellement défini dans la publication FIPS 46-3 et est public.
La clé est en fait constituée de 64 bits, dont 56 bits sont générés aléatoirement et utilisés dans l'algorithme. Les huit autres bits peuvent être utilisés pour la détection d'erreurs (dans une transmission par exemple).
Chacun des huit bits est utilisé comme bit de parité des sept groupes de 8 bits. Comme blowfish, le DES est un chiffrement Feistel. Il utilise les transformations de substitution et de transposition (chiffrement par produit). Il est aussi appelé Data Encryption Algorithm (DEA) **[17]**.

Algorithme

Pour être chiffré, un bloc subit tout d'abord une **permutation initiale**, puis un algorithme complexe est appliqué en fonction de la clé (**calcul médian**), et enfin le bloc subit une **permutation finale**. Cette dernière permutation est l'inverse de la permutation initiale. De cette façon, l'algorithme de chiffrement et de déchiffrement est le même. Le calcul médian dépendant de la clé peut être défini comme étant deux fonctions : une première appelée la fonction de chiffrement et une fonction de programmation de la clé.

Permutation initiale

Les 64 bits du bloc en entrée dans l'algorithme DES subissent la permutation initiale.

Bloc en entrée										Bloc en sortie							
1	2	3	4	5	6	7	8	P.I.		58	50	42	34	26	18	10	2
9	10	11	12	13	14	15	16			60	52	44	36	28	20	12	4
17	18	19	20	21	22	23	24			62	54	46	38	30	22	14	6
25	26	27	28	29	30	31	32			64	56	48	40	32	24	16	8
33	34	35	36	37	38	39	40			57	49	41	33	25	17	9	1
41	42	43	44	45	46	47	48			59	51	43	35	27	19	11	3
49	50	51	52	53	54	55	56			61	53	45	37	29	21	13	5
57	58	59	60	61	62	63	64			63	55	47	39	31	23	15	7

Ainsi, le premier bit du bloc résultant de la permutation initiale est le 58ème bit du bloc en entrée, le deuxième bit est le 50ème bit et ainsi de suite.

Calcul médian

Le calcul médian peut se résumer comme étant une fonction contenant 16 itérations identiques. Cette fonction traite deux blocs à la fois : un bloc de 32 bits, les données, et l'autre de 48 bits, la clé. Le résultat donne un bloc de 32 bits.
Le bloc de donnée de 64 bits est préalablement divisé en deux blocs de 32 bits, "L" et "R" (pour "Left" et "Right"), après être passé dans la permutation initiale. Ainsi, "L" contient les bits pairs et "R" contient les bits impairs. Les 48 bits du bloc "K" (pour "Key") sont choisis à partir de la clé initiale de 64 bits.
La sortie de L'R' après l'itération est définie par :

L' = R

R' = L XOR f(R,K)

L'R' = [R][L + f(R,K)]

L'entrée à la première itération du chiffrement est le bloc ayant subit la permutation initiale. À la fin, le bloc L'R' restant après la seizième itération devient le bloc de pré-sortie (avant la permutation finale). À chaque itération, un bloc "K" différent de 48 bits est choisi à partir de la clé de 64 bits.

Pour "n" variant de 1 à 16, on a:

$\mathbf{L_n = R_{n-1}}$

$\mathbf{R_n = L_{n-1}\ XOR\ f(R_{n-1},K_n)}$

b. Chiffrement asymétrique

i. Historique

Le concept de cryptographie à clef publique a été inventé par Whitfield Diffie et Martin Hellman en 1976, dans le but de résoudre le problème de distribution des clefs posé par la cryptographie à clef secrète. De nombreux algorithmes permettant de réaliser un cryptosystème à clef publique ont été proposés. Ils sont le plus souvent basés sur des problèmes mathématiques difficiles à résoudre, donc de leur sécurité est conditionnée par ces problèmes, sur lesquels on a maintenant une vaste expertise **[15]**.

ii. Principe

La clef publique utilisée pour le chiffrement est connue seulement par le détenteur de la clef privée.

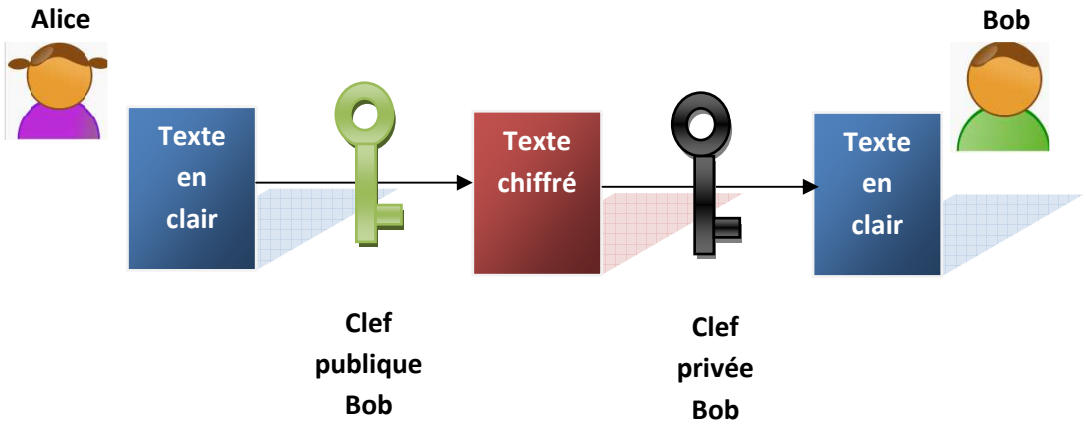

Figure B.3 : **Chiffrement**

La clef privée utilisée pour le chiffrement est connue seulement par son détenteur, mais tout le monde peut déchiffrer.

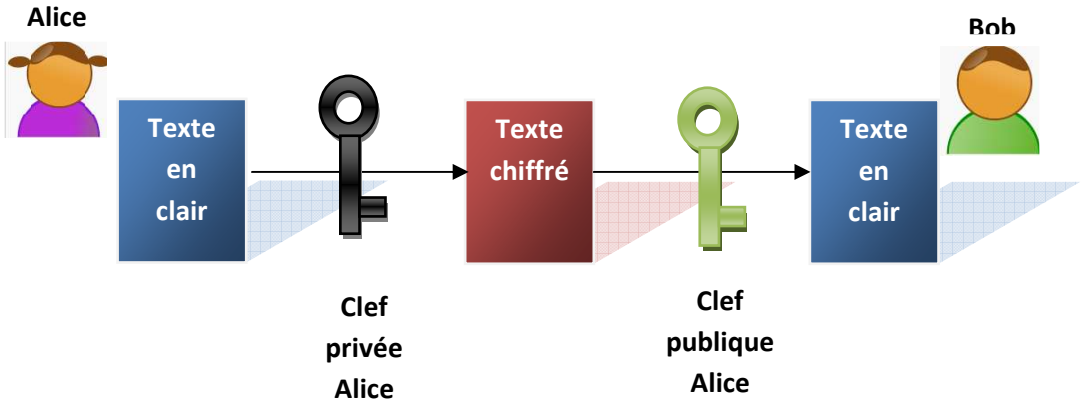

Figure B.4 : **Signature**

iii. RSA

RSA est un algorithme asymétrique de cryptographie à clé publique, utilisé dans le commerce électronique, et pour échanger des données confidentielles. Il a été décrit en 1977 par Ron Rivest, Adi Shamir et Len Adleman. RSA était breveté par le MIT en 1983 aux Etats Unis. En 2008, c'est le système à clef publique le plus répandu et plus utilisé.

B.4 Combinaison clefs publiques et clefs secrètes

Parfois on peut combiner la cryptographie à clef publique avec la cryptographie à clef secrète.

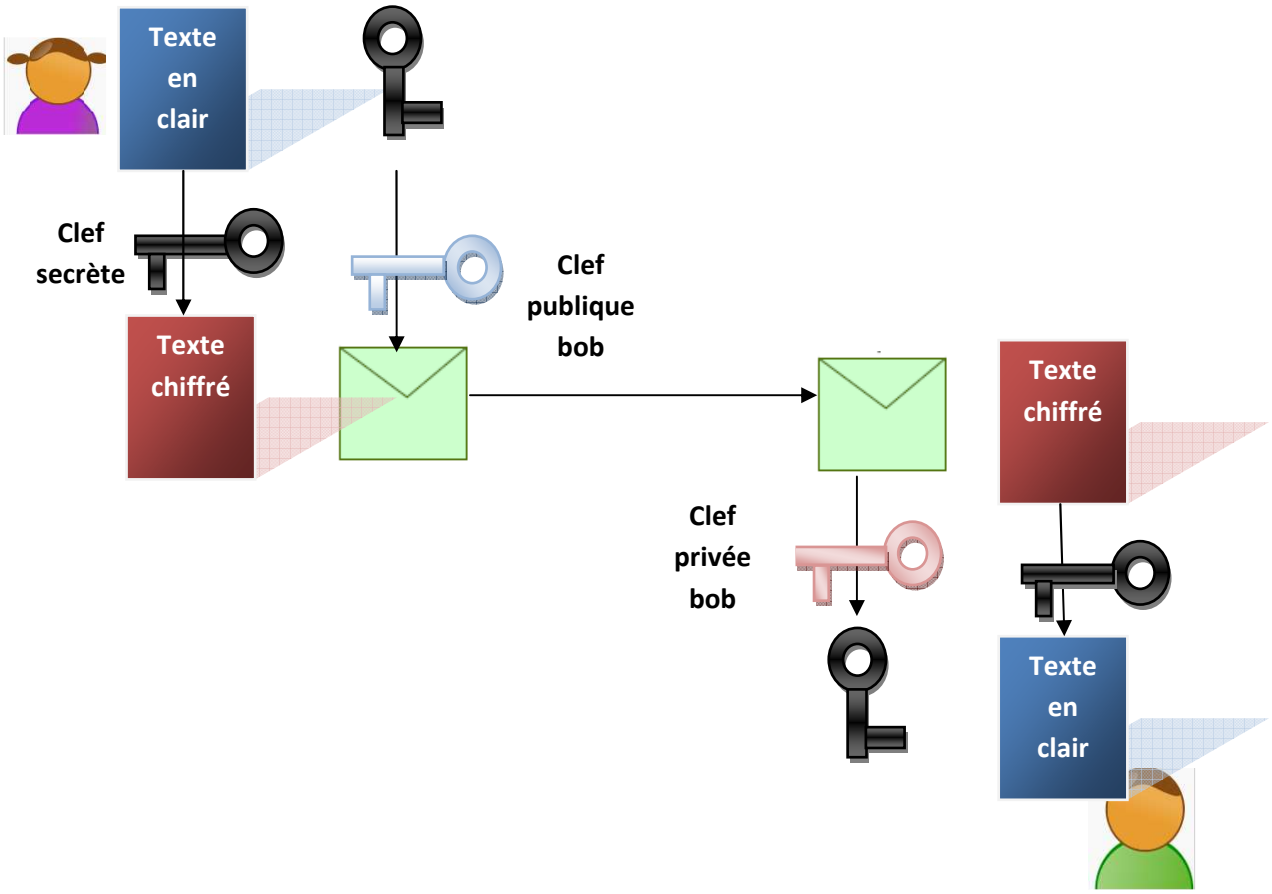

Figure B.5 : **Combinaison de clef publique/clef secrète**

REFERENCES

[1] RAZAFIARISON Zo Manankasina, Livre de mémoire de fin d'étude intitulé : « Réalisation d'une suite complète d'acquisition, de génération et de traitement du son », 2009.

[2] http://www.laboratoiredelavoix.com/wp-content/uploads/lavoix.pdf.

[3] http://tcts.fpms.ac.be/cours/1005-08/speech.

[4] Vu Minh Quang, Thèse « Exploitation de la Prosodie pour la Segmentation et l'Analyse Automatique de Signaux de Parole », soutenue le 20 septembre 2007.

[5] Anthony LARCHER, Thèse de Doctorat « Modèles acoustiques à structure temporelle renforcée pour la vérification du locuteur embarquée, 24 septembre 2009.

[6] http://scgwww.epfl.ch/courses/Traitement_de_la_parole-2005-2006-pdf/Drygajlo-Reconnaissance-du-locuteur.pdf.

[7] http://www.geea.org/IMG/pdf/Cours_TS.pdf

[8] Cours Logique combinatoire, 2ème Année, Département Electronique, ESPA 2007.

[9] E531 Cours Traitement Numérique du signal, 5ème Année, Département Electronique, ESPA, 2009-2010.

[10] http://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm.

[11] Mohamed CHETOUANI, Thèse de DOCTORAT « Codage neuro-prédictif pour l'extraction de caractéristiques de signaux de parole ».

[12] http://w3.u-grenoble3.fr/idl/IMG/protege/form17/ConfBONASTRE.pdf.

[13] http://www-clips.imag.fr/geod/User/laurent.besacier/M2R-ILP/1.a.Parole.pdf

[14] http://www.developpeez.com

[15] http://www.hsc.fr

[16] http://fr.wikipedia.org/wiki/RC4

[17] http://www.uqtr.ca/~delisle/Crypto/prives/blocs_des.php

Auteur: **RASAMIJAONA TOJOMANGA ALAIN MICHEL**

Titre : **VERIFICATION DU LOCUTEUR PAR RECONNAISSANCE VOCALE**

Nombre de pages : **58**

Nombres de figures : **31**

Nombre de tableau : **01**

RESUME

L'évolution du système informatique actuel entraine une progression sans cesse de la technologie vocale. Et grâce à cette dernière que la biométrie vocale est née et qui devient un outil de sécurité plus répandu aujourd'hui. « ***RecSpeaker version 1.0*** » est un logiciel de reconnaissance vocale, possédant trois fonctionnalités principales : un outil de vérification d'un locuteur qui détermine vocalement l'identité d'un individu, une fonction de visualisation de signal vocal et de spectre et une fonction de protection servant la protection d'un fichier quelconque dans le disque dur.

Mots clés : signal vocal, reconnaissance vocale, vérification du locuteur, identité, parole, audio,

FFT, MFCC, GMM.

Directeur de mémoire : Monsieur **ANDRIAMANANTSOA Guy Danielson**

Adresse de l'auteur :

Lot 191 Ambohidratrimo

mangatojoboy@yahoo.fr