# Cryptosystemes biometriques

## 2.1 Introduction

Dans ce chapitre, nous allons présenter le concept des cryptosystèmes biométriques, leur classifications et leurs différentes méthodes,ensuite nous détaillons les différents principes et étapes de la méthode proposée.

## 2.2 Cryptosystème biométrique

Les cryptosystèmes biométriques combinent la cryptographie et la biométrie pour bénéficier des atouts des deux domaines. Dans de tels systèmes, alors que la cryptographie fournit des niveaux de sécurité élevés et ajustables, la biométrie apporte la non-répudiation et élimine le besoin de se souvenir des mots de passe ou de porter des jetons, etc.[20]
Les cryptosystèmes biométriques sont conçus pour lier en toute sécurité une clé numérique à une clé biométrique ou pour générer une clé numérique à partir d'une clé biométrique.[21]

### 2.2.1 Protection du modèle biométrique

Les cryptosystèmes biométriques libèrent des clés cryptographiques qui sont associées aux traits biométriques des utilisateurs enregistrés. Par conséquent, les cryptosystèmes biométriques offrent des solutions pour sécuriser la gestion des clés basée sur la biométrie ainsi que la protection des modèles biométriques. Puisque l'authentification est effectuée indirectement en vérifiant les validités de clé, le système n'a pas besoin de stocker les modèles biométriques d'origine. De plus, la plupart des cryptosystèmes biométriques fournissent des mécanismes pour mettre à jour ces clés à tout moment afin que les utilisateurs puissent appliquer différentes clés à différentes applications.

### 2.2.2 Classification des cryptosystèmes biométriques

Les cryptosystèmes biométriques sont classés comme systèmes de liaison de clé et de génération de clé en fonction de la façon dont le croquis sécurisé est obtenu. Le croquis sécurisé est une information publique sur les caractéristiques biométriques stockées dans les bases de données lors de l'inscription.[20]

#### 2.2.2.1 Cryptosystèmes biométriques de liaison de clés

Les données auxiliaires sont obtenues en liant une clé choisie à un modèle biométrique. À la suite du processus de liaison, une fusion de la clé secrète et du modèle biométrique est stockée en tant que données auxiliaire. En appliquant un algorithme de récupération de clé approprié, les clés sont obtenues à partir des données auxiliaires lors de l'authentification[22]. Étant donné que les clés cryptographiques sont indépendantes des fonctionnalités biométriques, elles sont révocables tandis qu'une mise à jour de la clé nécessite généralement une réinscription afin de générer de nouvelles données auxiliaires. Le mode de fonctionnement général d'un schéma de liaison de clé est illustré sur la **figure 2.1** :

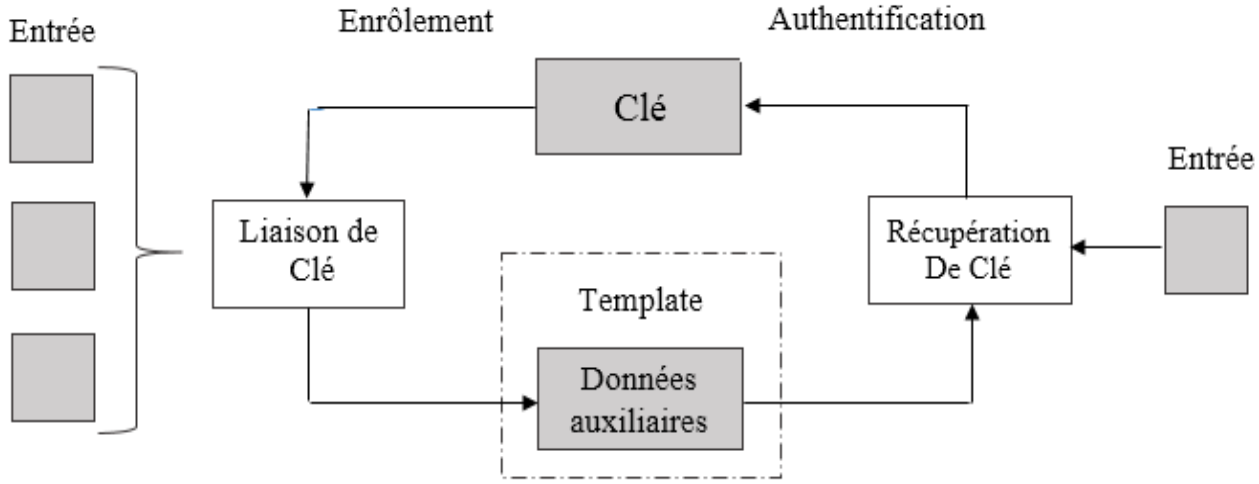



FIGURE 2.1 – Mode de fonctionnement général d'un schéma de liaison de clé

#### 2.2.2.2 Cryptosystèmes biométriques de génération de clés

Si l'esquisse sécurisée est dérivée uniquement du modèle biométrique et que la clé cryptographique est directement générée à partir des données d'aide et des fonctionnalités biométriques de requête, alors on parle de cryptosystèmes biométriques de génération de clé. Les données auxiliaires sont appliquées pour récupérer le modèle biométrique d'origine[21]. Le mode de fonctionnement général d'un schéma de génération de clé est illustré sur la **figure 2.2** :

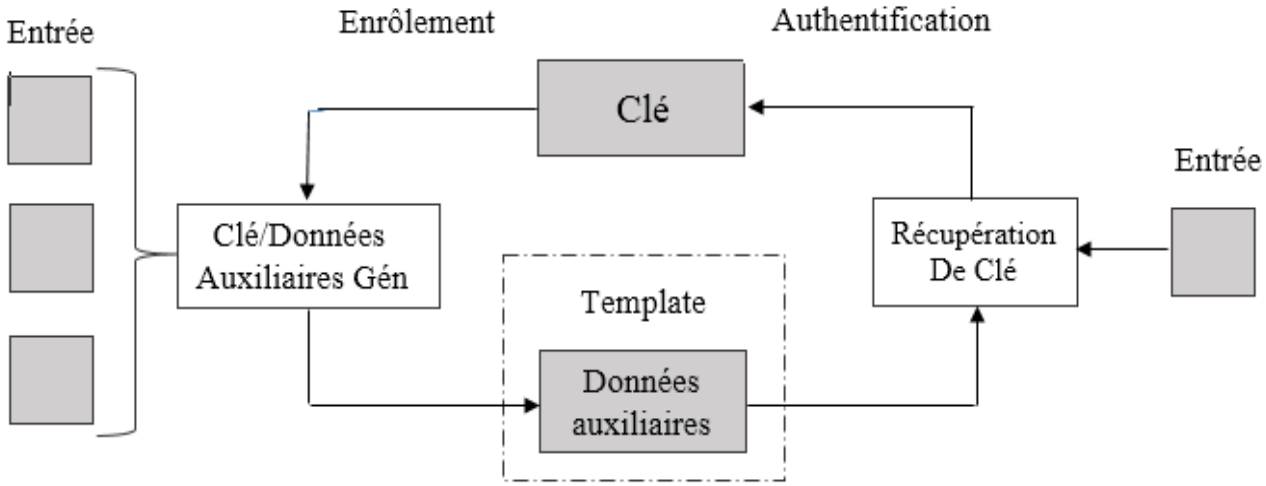



FIGURE 2.2 – mode de fonctionnement général d'un schéma de génération de clé

### 2.2.3 les méthodes de cryptosystéme biométrique

#### 2.2.3.1 Fuzzy commitment

Le principe de cette méthode est d'utiliser la donnée biométrique courante pour recalculer une valeur qui servira ensuite pour l'authentification de l'utilisateur[25]. Plus formellement, la méthode dite du fuzzy commitment se décompose en deux étapes : enrôlement puis authentification. Traditionnellement en biométrie, l'étape d'enrôlement consiste à relever plusieurs fois la donnée biométrique de l'utilisateur (son empreinte par exemple) afin de constituer une valeur de référence. L'étape d'authentification consiste ensuite à comparer la valeur courante à la valeur de référence pour déterminer si l'utilisateur est bien celui attendu. La méthode dite du fuzzy commitment suit ces deux étapes sauf que la valeur de référence stockée n'est pas une donnée biométrique et ne permet pas de retrouver la donnée biométrique utilisée pour la générer. Pour cela, le fuzzy commitment utilise un ensemble de mots de code dans $\{0,1\}^n$ et une fonction de hachage $H$ (**figure 2.3**).

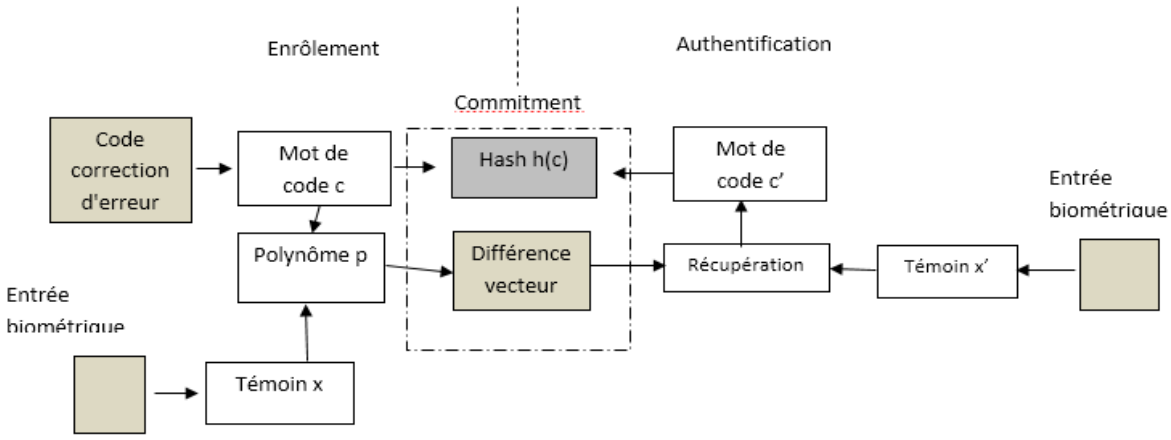



FIGURE 2.3 – Schéma de fuzzy commitment.
[31]

— **Enrôlement :** Un utilisateur **U** présente son empreinte biométrique représentée sous la forme d'une suite **x** de **n** bits. Le système choisit aléatoirement un mot de code $c \in \{0,1\}^n$ et calcule $(c - x, H(c))$. Le fuzzy commmitment de l'utilisateur **U** est le couple $(c - x, H(c))$. Cette donnée est stockée, le reste est effacé. A la fin de cette étape, le système authentifiera comme étant **U** tout utilisateur capable de produire une empreinte biométrique permettant de retrouver le mot de code **c**.

— **Authentification :** Supposons qu'un utilisateur se présente sous l'identité de **U** et que son signal biométrique courant soit **x'**. Le système utilise **x'** pour vérifier l'engagement $(c - x, H(c))$. Pour cela,il faut tout d'abord calculer $(c - x) + x'$. Si l'utilisateur était bien celui qu'il prétend être, alors son empreinte biométrique courante **x'** devrait être proche de son empreinte biométrique de référence **x** et par voie de conséquence $(c-x)+x'$ devrait être proche de **c**, au sens de la métrique de Hamming.Si la distance entre $(c - x) + x'$ et **c** est inférieure à la distance du code, alors le mot de code **c'**, le plus proche de $(c-x)+x'$ est égal à **c**. Pour cela, il suffit de vérifier que $H(c') = H(c)$ où **c'** est le mot de code le plus proche de $(c - x) + x'$.

#### 2.2.3.2 Fuzz vault

Les auteurs proposent une méthode appelée fuzzy vault[26]. L'authentification d'un utilisateur repose, là encore, sur sa capacité à présenter une donnée biométrique permettant de retrouver un secret, le secret en question est un polynôme **P** de degré **n**. Lors de l'étape d'enrôlement, le système calcule un fuzzy vault **V** , littéralement coffre fort flou, à partir de **P** et de l'ensemble des minuties de l'empreinte de l'utilisateur. L'utilisateur est authentifié lorsqu'il est possible de retrouver **P** à partir de **V** et des minuties apparaissant dans son empreinte courante(**Figure 2.4**).

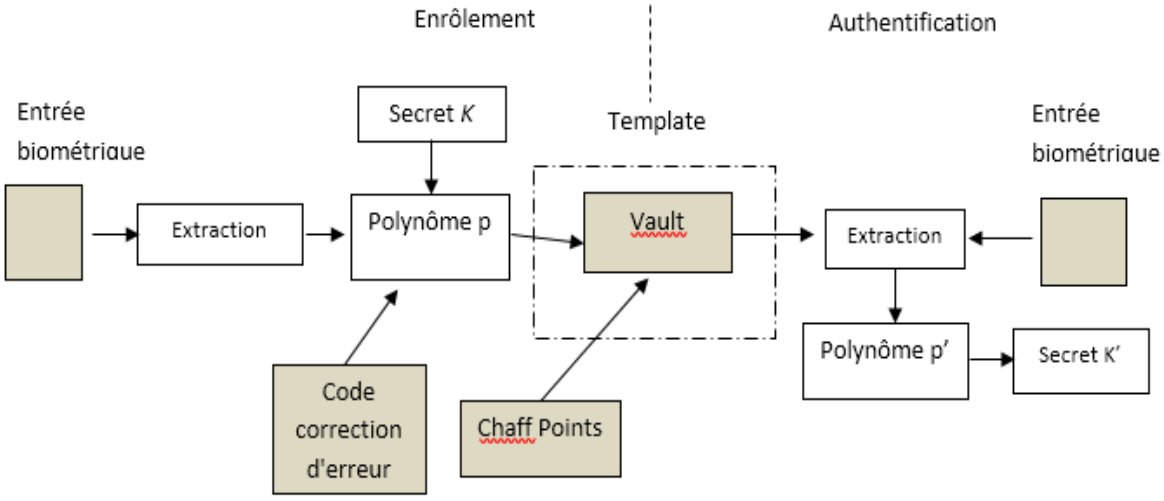



FIGURE 2.4 – Schéma de fuzzy vault.
[31]

Plus précisément, l'algorithme de Juels et Sudan peut se présenter comme suit :

— **Enrôlement :** Considérons un utilisateur **U**, l'ensemble des minuties $\{a_i\}_{1\leqslant i\leqslant t}$ apparaissant dans l'image de son empreinte digitale et un polynôme **P** de degré **k** où $t > k$. Nous calculons tout d'abord l'ensemble des points $R_1 = \{(a_i, P(a_i)\}_{1\leq i\leq t}$. Nous générons ensuite un second ensemble de points $R2 = \{(x_j, y_j\}_{1\leq j\leq n}$ tels que $x_j \neq a_i$ et $y_j \neq P(a_i)$ pour tout $1 \preceq i \preceq t$ et tout $1 \preceq j \preceq n$. L'ensemble $R_2$ est destiné à brouiller l'information venant de $R_1$. Le coffre-fort flou de l'utilisateur **U** est l'ensemble $V_U = R_1 \cup R_2$. Cet ensemble est rendu publique, le reste est effacé.

— **Authentification :** Lorsqu'un utilisateur souhaite s'authentifier en temps que **U**, il présente son empreinte digitale et nous extrayons l'ensemble des minuties $\{b_i\}_{1\preceq i\preceq t'}$ de l'image acquise. Nous sélectionnons ensuite dans $V_U$ l'ensemble des points de la forme $(x, y)$ où x est proche d'une valeur $b_i$. Si l'empreinte proposée est suffisamment proche de l'empreinte de l'utilisateur **U** alors ce procédé, un cryptosystème biométrique pour l'authentification permet de retrouver suffisamment de points de $R_1$ pour pouvoir reconstruire **P** par interpolation de Lagrange. Si c'est le cas, l'utilisateur est authentifié en tant que **U**.

#### 2.2.3.3 Schémas de Quantization

Au sein de ce groupe de schémas, les données auxiliaires sont construites de manière à aider à quantifier les caractéristiques biométriques afin d'obtenir des clés stables( **Figure 2.5**).

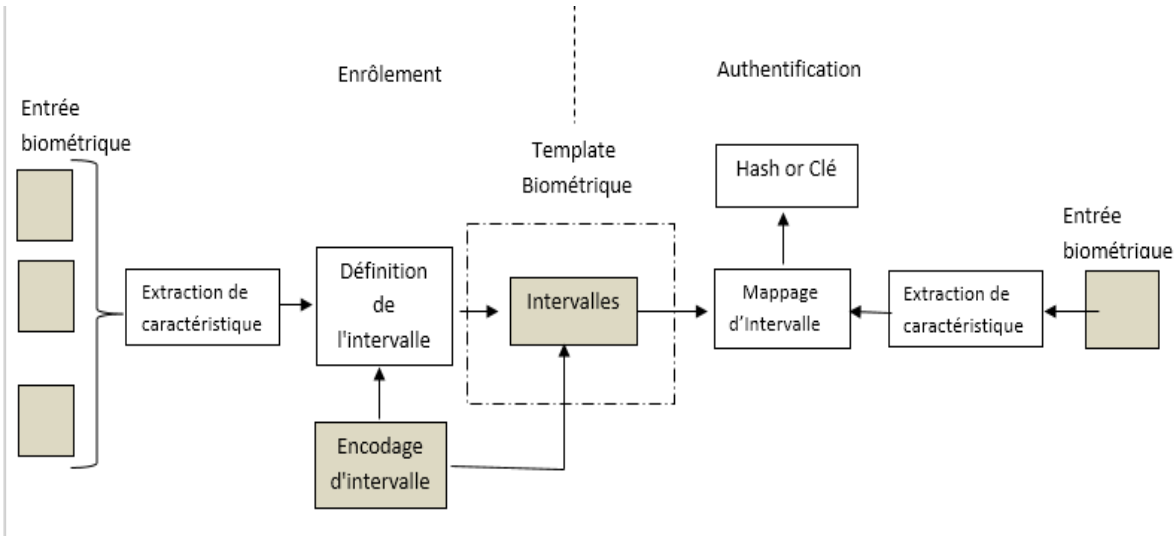



FIGURE 2.5 – Schéma de Quantization proposée.

**Mode de fonctionnement :**
dans le schéma[27] de quantification généraux, qui ont été appliqués aux caractéristiques biométriques physiologiques et comportementales, traiter les vecteurs de caractéristiques à partir de plusieurs échantillons de recrutement et dérivez des intervalles appropriés pour chaque élément de caractéristique (des vecteurs de caractéristiques à valeur réelle sont nécessaires ). Ces intervalles sont codés et stockés en tant que données d'assistance. Au moment de l'authentification, là encore, les caractéristiques biométriques d'un sujet sont mesurées et mappées dans les intervalles précédemment définis, générant un hachage ou une clé. Afin de fournir des clés ou des hachages pouvant être mis à jour, la plupart des schémas fournissent un codage paramétré des intervalles.

#### 2.2.3.4 Méthode cryptosystème biométrique basé sur les minuties voisinages

est une technique qui utilise les informations de localisation des points de minutie pour construire un modèle hautement sécurisé pour un utilisateur. Pour chaque point de minutie, un emplacement modifié sécurisé est généré en utilisant les informations de ses minuties voisines et un jeu de clés. Nous allons détaillées les étapes de cette méthode dans la section suivants.

### 2.2.4 Avantage des cryptosystèmes biométriques

Les cryptosystèmes biométriques offrent plusieurs avantages par rapport aux systèmes biométriques[19] conventionnels. Les principaux avantages peuvent être résumés comme suit :

— **Protection du gabarit :** dans les systèmes cryptographiques biométriques, le gabarit biométrique d'origine est masqué de sorte qu'une reconstruction est difficilement réalisable.

— **Libération de clé dépendante de la biométrie :** les cryptosystèmes biométriques fournissent des mécanismes de libération de clé basés sur la présentation de données biométriques.
— **Révocabilité des modèles biométriques :** plusieurs instances de modèles sécurisés peuvent être générées en liant ou en générant différentes clés.
— **Sécurité accrue :** les cryptosystèmes biométriques empêchent plusieurs types traditionnels d'attaques contre les systèmes biométriques (par exemple, les attaques de substitution).
— **Meilleure acceptation sociale :** en raison des avantages de sécurité mentionnés ci-dessus, l 'acceptation sociale des applications biométriques devrait augmenter.

## 2.3 Méthode de cryptosystème biométrique basé sur les minuties voisinages

Pour disposer un système d'authentification d'empreintes digitales hautement sécurisé, nous besoin de générer un modèle sécurisé pour un utilisateur. En utilisant les données extrait de l'empreinte digitale de l'utilisateur, un modèle sécurisé est généré et stocké dans la base de données, qui est également utilisée pour l'authentification. Parmi diverses fonctionnalités d'empreintes digitales, la technique utilise des points singuliers et des informations de points de minuties pour construire un modèle utilisateur sécurisé, à l'aide d'un jeu de clés utilisateur $p_0$, $q_0$, $r_0$.[23]
Comme représenter à la **La figure 2.6**, notre Schéma est composé de quatre étapes :

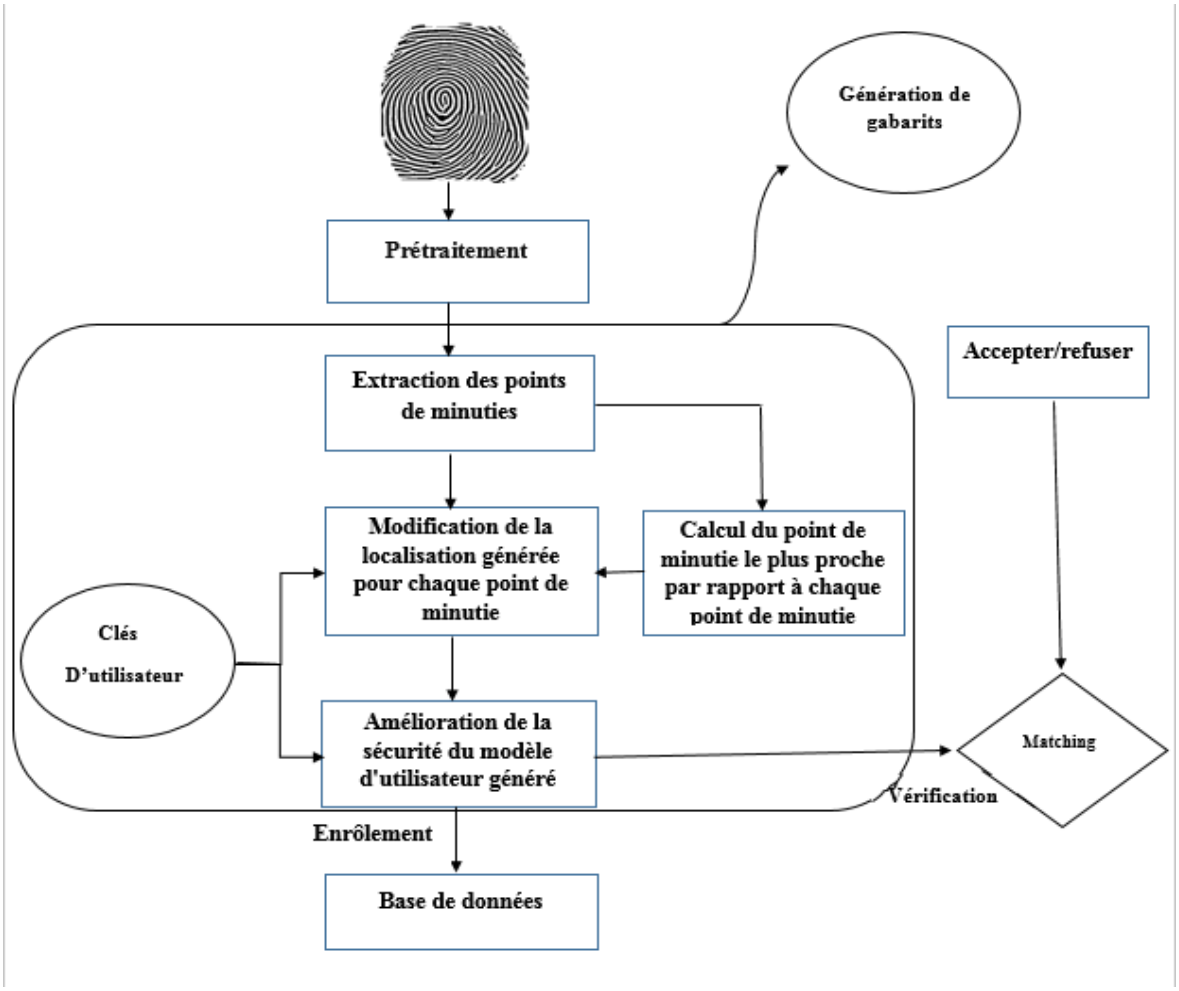



FIGURE 2.6 – Datagrame des étapes de la technique proposée.

### 2.3.1 Prétraitement

Initialement, l'empreinte digitale est capturée à l'aide de capteurs. Pour améliorer la qualité d'image nous utilisons le filtre de Gabor. Ce dernier est une opération de traitement d'images numériques qui consiste à appliquer des opérateurs afin de faire des transformations sur toute l'image ou à une partie d'elle pour améliorer la qualité visuelle de cette dernière.[24](**Figure 2.7**)

$G(x, y, \theta, f) = e^{-\frac{1}{2}(\frac{x_\theta^2}{\sigma_x^2}) + (\frac{y_\theta^2}{\sigma_y^2})\cos(2\pi f x_\theta)}$

avec : $x_\theta = \mathrm{x}\ \cos\theta + \mathrm{y}\ \sin\theta$

et : $y_\theta = \mathrm{y}\cos\theta + \mathrm{x}\sin\theta$

où $\theta$ est l'orientation de la sinusoïde, f sa fréquence et $\sigma_x$ (respectivement $\sigma_y$ ) l'écart type de la gaussienne selon l'axe des abscisses.

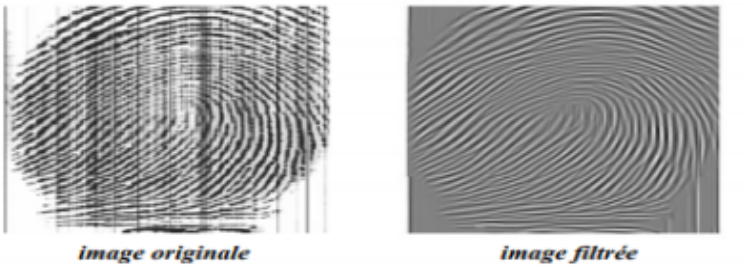



FIGURE 2.7 – Le résultat obtenu après le pré-traitement.

### 2.3.2 Extraction de points des minuties

A partir de l'image filtrée, sont extraites de diverses informations présentes dans l'empreinte digitale, nous avons vu dans le chapitre 1, l'extraction des minuties.(**figure 2.8**).

FIGURE 2.8 – Résultat de l'extraction de points des minuties.

### 2.3.3 Génération de modèles sécurisés(gabarit)

Les emplacements des points de minutie et leur orientation sont aléatoires. Habituellement, le modèle utilisateur est directement stocké dans la base de données dans la forme des points de minutie avec leurs attributs (tels que l'emplacement, type et orientation). L'empreinte digitale originale d'un utilisateur peut être reconstruite. Donc pour éviter cela au lieu de stocker directement les points de minutie avec leurs attributs, la technique proposée stocke les emplacements modifiés de points de minuties. Ces emplacements modifiés sont obtenus par modifier l'emplacement d'origine des points de minuties à l'aide d'un jeu de clés utilisateur ainsi qu'en utilisant l'emplacement des minuties voisines points (Ce processus est décrit dans le cadre de l'algorithme 1 (**La figure 2.9**) de l'étape 5 à l'étape 22.).

1: **Input:** *Minutiae points locations* $(x_1, y_1)$, $(x_2, y_2)$, ..., $(x_n, y_n)$ *and key-set* $\{p_0, q_0, r_0\}$
2: **Output:** *secured user template (UT)*
3: */* Computation of key* $s_0$ *for user template security enhancement */*
4: $s_0 = \lfloor p_0 \rfloor + \lfloor q_0 \rfloor \times (2^{16}) + \lfloor r_0 \rfloor \times (2^{32})$
5: */* Secured template generation, from step 6 to step 22 */*
6: **for** $i = 1$ to n **do**
7: $dis_i = \infty$
8: **for** $k = 1$ to $n$ **do**
9: **if** $i \neq k$ **then**
10: $dis = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}$
11: **if** $dis_i > dis$ **then**
12: $dis_i = dis$
13: $j = k$
14: **end if**
15: **end if**
16: **end for**
17: */* Computation of modified locations */*
18: $x'_i = x_i + (p_0 \times \cos(q_0 + tan^{-1}(\frac{y_i - y_j}{x_i - x_j})))$
19: $y'_i = y_i + (p_0 \times \sin(q_0 + tan^{-1}(\frac{y_i - y_j}{x_i - x_j})))$
20: */* Reducing translation due to intra-subject variance */*
21: $x'_i = x'_i - x_{sing}$
22: $y'_i = y'_i - y_{sing}$
23: */* Security enhancement of the user template */*
24: $\begin{bmatrix} x'_i \\ y'_i \end{bmatrix} = \begin{bmatrix} cos(r_0) & -sin(r_0) \\ sin(r_0) & cos(r_0) \end{bmatrix} \times \begin{bmatrix} x'_i \\ y'_i \end{bmatrix}$
25: $x'_i = x'_i + (s_0 \times \cos(r_0))$
26: $y'_i = y'_i + (s_0 \times \sin(r_0))$
27: **end for**
28: */* abscissae of the secured user template */*
29: $UT(1, :) = \{x'_1, x'_2, ..., x'_n\}$
30: */* ordinates of the secured user template */*
31: $UT(2, :) = \{y'_1, y'_2, ..., y'_n\}$

FIGURE 2.9 – Algorithme 1 : Calcul du modèle utilisateur sécurisé

Supposons qu'il y ait n points de minutie obtenus à partir d'une empreinte digitale, $m_i = \{x_i, y_i, \theta_i, type_i\}$, où $x_i$ est la valeur en abscisse, $y_i$ est la valeur ordonnée, $\theta_i$ est la valeur d'orientation, et $type_i$ représente le type de minutie (bifurcation de crête ou début / fin de crête). Pour générer les emplacements modifiés d'un point de minutie $m_i$, la technique proposée utilise uniquement l'emplacement d'un point de minutie (valeur d'abscisse $x_i$ et ordonnée $y_i$).

Pour un point de minutie $m_i$, soit $m_j$ (valeur d'abscisse $x_j$ et valeur d'ordonnée $y_j$) le point de minutie le plus proche au point de minutie $m_i$, maintenant en utilisant les clé utilisateur $q_0$ et $p_0$ et le point de minutie $m_j$, pour modifier l'emplacement $m_i$. Comme montre la **figure 2.10**, un emplacement modifié correspondant à un point de minutie $m_i$ est généré en calculant $x'_i$ et $y'_i$. Les valeurs de $x'_i$ et $y'_i$ sont obtenues à partir de l'emplacement d'origine du point de minutie $m_i$ ($x_i$, $y_i$) et $m_j$ ($x_j$, $y_j$) à l'aide de $p_0$ et $q_0$. les valeurs $x'_i$ et les valeurs $y'_i$ sont

calculées comme suit :

$$x'_i = x_i + (P_0 \times \cos(q_0 + \tan^{-}1\frac{y_i - y_j}{x_i - x_j})) \tag{2.1}$$

$$y'_i = y'_i + (P_0 \times \sin(q_0 + \tan^{-}1\frac{y_i - y_j}{x_i - x_j})) \tag{2.2}$$

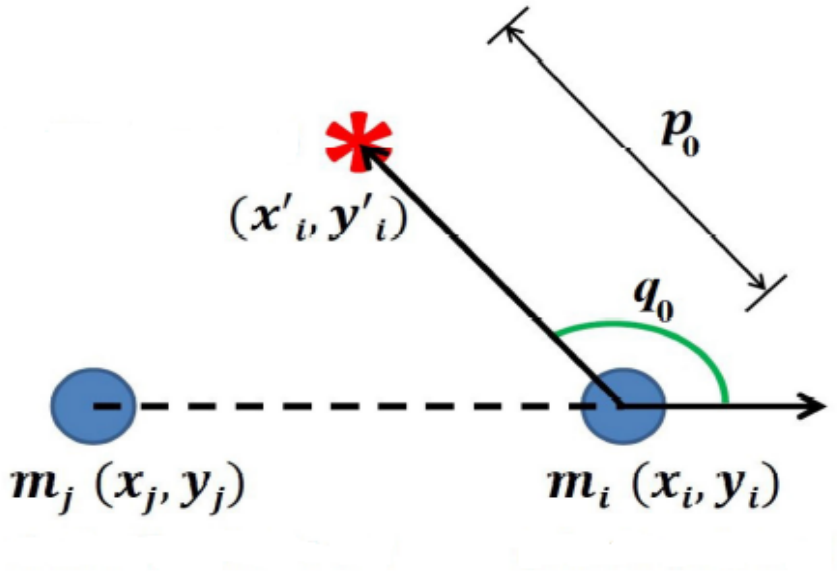



FIGURE 2.10 – Génération d'un emplacement modifié $m_i$ avec $m_j$ le plus proche à l'aide des touches utilisateur $p_0$ et $q_0$.

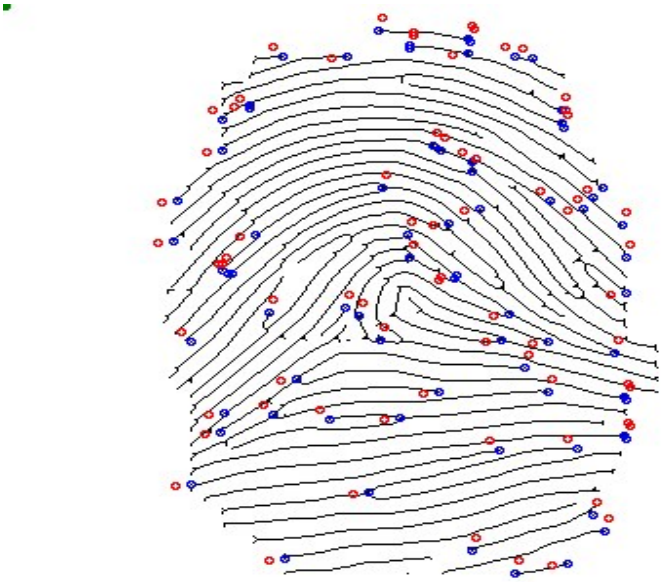


FIGURE 2.11 – les emplacements modifiés pour les points de minutie d'un empreinte digitale.

En raison de la variance intra-sujet, il existe un problème de traduction. Pour réduire les effets de la traduction, on utilise l'emplacement du point singulier ($x_{sing}$ étant la valeur

d'abscisse et $y_{sing}$ la valeur d'ordonnée d'un point singulier). Tous les emplacements modifiés des points de minuties sont traduits de telle sorte que l'emplacement du point singulier devienne l'origine (**La figure 2.12** a et b).
Cela se fait de la manière suivante :

$$x'_i = x'_i - x_{sing} \tag{2.3}$$

$$y'_i = y'_i - y_{sing} \tag{2.4}$$

Pour chaque point singulier présent dans une empreinte digitale, un ensemble des emplacements modifiés correspondants des points de minuties sont générés. En d'autres termes, on peut dire que pour chaque point singulier d'une empreinte digitale, un modèle d'utilisateur est généré et enregistré dans une base de données lors du processus d'enregistrement. Au moment de la vérification, tous ces modèles sont comparés avec le modèle de requête.

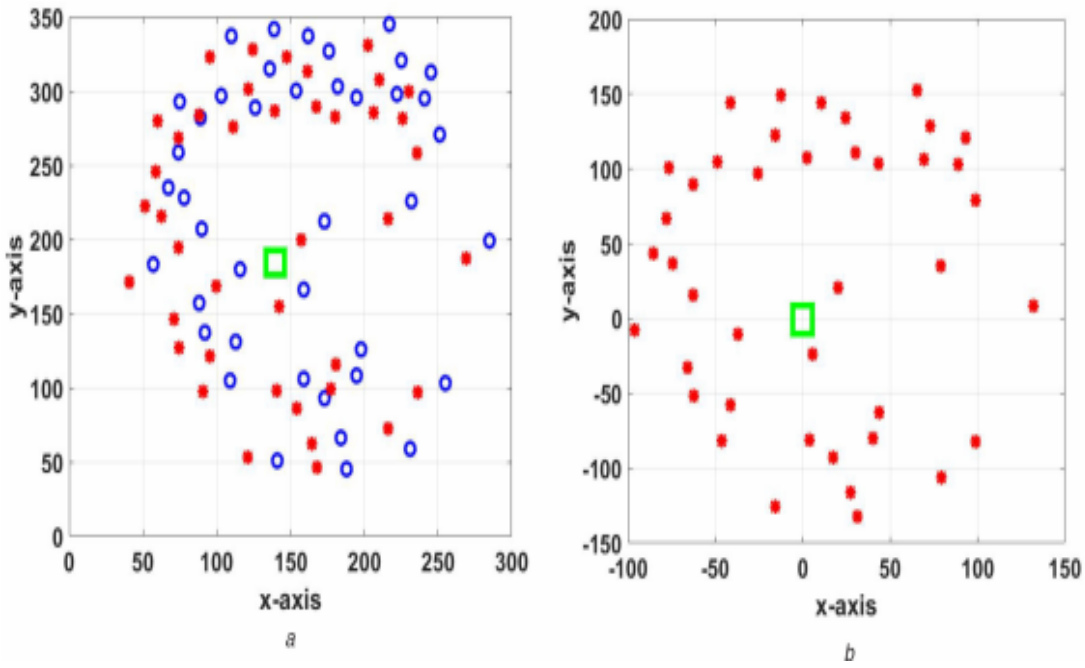



FIGURE 2.12 – les emplacements modifiés pour les points de minutie d'un empreinte digitale.

Amélioration de la sécurité du modèle utilisateur généré Le modèle d'utilisateur sécurisé généré à la dernière étape est encore amélioré en utilisant $r_0$ et $s_0$.
$s_0$ est un entier de 48 bits généré à l'aide de valeurs intégrales de $p_0$, $q_0$ et $r_0$ :

- les 16 premiers bits (à partir du bit le moins significatif) sont des valeurs intégrales de $p_0$.
- du 17 au 32 bits (à partir des bits les moins significatifs) sont des valeurs intégrales de $q_0$ .
- les 16 derniers bits (du bit le moins significatif) sont des valeurs intégrales de $r_0$.

comme le montre **la figure 2.13**. Tous les emplacements modifiés sont tournés de l'angle de $r_0$ par rapport à l'origine, puis traduits par des unités $s_0$ à partir de l'origine, ce qui fait un angle $r_0$ par rapport à l'axe des abscisses.

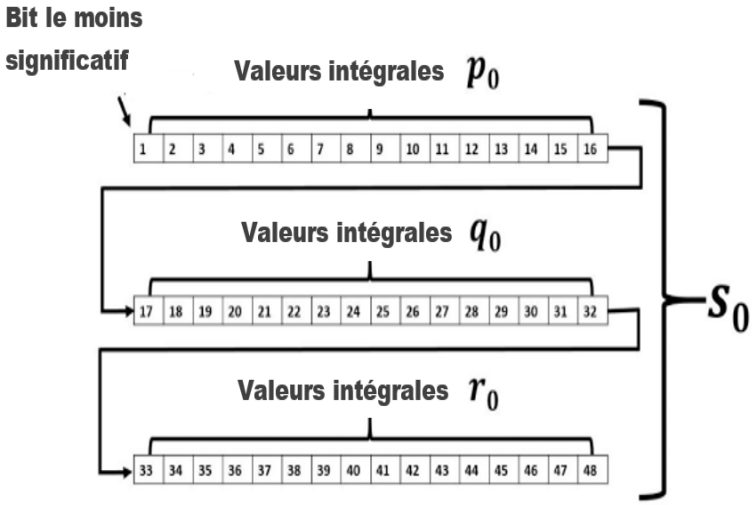



FIGURE 2.13 – Calcul de la clé $s_0$ de 48 bits en utilisant $p_0$, $q_0$ et $r_0$.

les valeurs des abscisses des emplacements modifiés sont augmentées de $s_0 \times cos(r_0)$ et la valeur des ordonnées de $s_0 \times sin(r_0)$.les emplacements sont les suivants :

$$S_0 = \lfloor p_0 \rfloor + \lfloor q_0 \rfloor \times (2^{16}) + \lfloor r_0 \rfloor \times (2^{32}) \tag{2.5}$$

$$x'_i = x'_i + (s_0 \times \cos(r_0)) \tag{2.6}$$

$$y'_i = y'_i + (s_0 \times \sin(r_0)) \tag{2.7}$$

Enfin, ces emplacements modifiés et améliorés de sécurité généré à l'aide d'attributs de points de minutie, de point singulier et le jeu de clés d'utilisateurs sont stockés dans la base de données en tant que modèle utilisateur pour la vérification d'un utilisateur à l'avenir.

### 2.3.4 Comparaison

Le point singulier qui est présent à l'endroit le plus proche par rapport au centre d'une image d'empreinte digitale de requête est sélectionné, et en l'utilisant, un seul modèle d'utilisateur est construit au moment de la vérification.
Ce gabarit est comparé à tous les gabarits qui sont présents dans la base de données par la distance Hausdorff
tel que : $s^{(1)} = s_1^{(1)}, s_2^{(1)}, .., s_n^{(1)}$ et $s^{(2)} = s_1^{(2)}, s_2^{(2)}, .., s_m^{(2)}$ être deux signatures, la distance de Hausdorff est défini comme :

$H_d(s^{(1)}, s^{(2)}) = max\{max\{d(n, s^{(2)}) : n \in s^{(1)}\}, max\{d(m, s^{(1)}) : m \in s^{(2)}\}\}$
où $d(n, s^{(1)}) = min\{d(n, m) : m \in s^{(1)}\}$

## 2.4 Conclusion

Dans ce chapitre nous avons vue les cryptosystèmes biométriques et ses méthodes, ainsi ses avantages ensuite nous avons détaillé les différents principes et étapes de la méthode proposée, qui utilise les informations de localisation des points de minuties pour construire un modèle hautement sécurisé. La performance de la méthode proposée est évaluée en l'appliquant sur une base de donnée, où les résultats obtenus seront analyser, chose qui sera abordé dans le chapitre suivant.

Chapitre 3

# Tests et résultats expérimentaux

## 3.1 Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes de la conception de la méthode proposée "Cryptosystème biométrique basé sur les minuties voisinages", nous présentons dans ce chapitre un aperçu général sur la phase pratique de notre travail.
Le but de ce projet est de construire pour la sécurisation de notre base de données d'empreinte digitale en appliquant certaine manipulation acceptable comme la rotation, le bruit, nous mettons en évidence les raisons de nos choix technique, les testes sur l'application et les résultats obtenue
Nous résumons cette mise en ouvre en trois parties :

- Environnement de développement
- Présentation l'application
- Tests et évaluation

## 3.2 Environnement de développement

### 3.2.1 Matériel

Nous avons utilisés deux machines avec les caractéristiques suivantes :

| **Caractéristiques** | **Machine1** | **Machine2** |
|---|---|---|
| **Processeur** | 1.80 GHz Intel core i3-CPU | 2.40 GHz Intel core i5-6200U |
| **RAM** | 3.00 GO | 400 GO |
| **Carte graphique** | Intel(R) HD Graphics 4000 | Intel(R) HD Graphics 4000 |
| **Système d'exploitation** | Windows 7 64 Bits | Windows 7 64 Bits |

Table 3.1 – Caractéristiques des machines utilisés

### 3.2.2 Software

L'environnement logiciel utilisé pour la réalisation de notre application est : Python


FIGURE 3.1 – Logo Python

Nous avons utilisés la version Python 2.7 comme langage de programmation, parmi les raisons de cette utilisation :

- Python fonctionne sur différentes plateformes (Windows, Mac, Linux).
- Il a une syntaxe simple claire, respecte les standards du domaine. Similaire a la langue anglaise.
- langage peut être traite de manière procédurale, de manière orientée objet ou de manière fonctionnelle.

**Modules et bibliothèques** Une bibliothèque est une ensemble de fonctions. Elles sont regroupées et mises à disposition afin de pouvoir être utilisées sans avoir à les réécrire.
Celles-ci permettent de faire : du calcul numérique, du graphisme, de la programmation internet ou réseau, du formatage de texte, de la génération de documents...
Parmi les différentes bibliothèques utilisés dans notre application :

**Module PIL :**
La bibliothèque PIL (Python Imaging Librairie) permet la manipulation de tout type d'images et fournit quelques fonctions de traitement d'images de base.

**Numpy :**
Numpy est une bibliothèque numérique apportant le support efficace de larges tableaux multidimensionnels, et de routines mathématiques de haut niveau.

**OpenCV :**
Cette bibliothèque permet de manipuler les structures de base, réaliser des opérations sur des matrices, dessiner sur des images, sauvegarder et charger des données.

**Matplotlib :**
Matplotlib est une bibliothèque destinée à tracer et visualiser des données sous formes de graphiques.

**TKinter :**
Tkinter (de l'anglais Tool kit interface) est la bibliothèque graphique libre d'origine pour le

langage Python, permettant la création d'interfaces graphiques. Elle vient d'une adaptation de la bibliothèque graphique Tk écrite pour Tcl.

**SciPy :**

La librairie SciPy contient de nombreuses boites à outils consacrées aux méthodes de calcul scientifique. Ses différents sous-modules correspondent à différentes applications scientifiques, comme les méthodes d'interpolation, d'intégration, d'optimisation, de traitement d'image, de statistiques, de fonctions mathématiques spéciales, etc.

**os :**

fonctions permettant d'interagir avec le système d'exploitation.

**math**

toutes les fonctions utiles pour les opérations mathématiques (cosinus,sinus,exp, etc.).

**random :**

Des fonctions permettant de travailler avec des valeurs aléatoires.

## 3.3 Présentation de l'application

Le but de notre travail est de tester la performance de notre algorithme (cryptosystème biométrique basé sur les minuties voisinages) contre quelques attaques acceptables, sur une base d'image d'empreinte digitale, nous pouvons mesurer la performance de cet algorithme.

### 3.3.1 Base de donnée utilisée

Pour évaluer la méthode proposée dans ce mémoire nous l'avons appliqué sur certain images (80 images) de la base FVC 2002, la résolution de ces images est ($388 \times 374$) ainsi que son format est TIFF. Il existe dans cette base de données huit échantillons par doigt ce qui permet aux algorithmes de comprendre les variations entre les doigts correspondants. La **figure 3.2** suivante montre une partie de cette base de données :

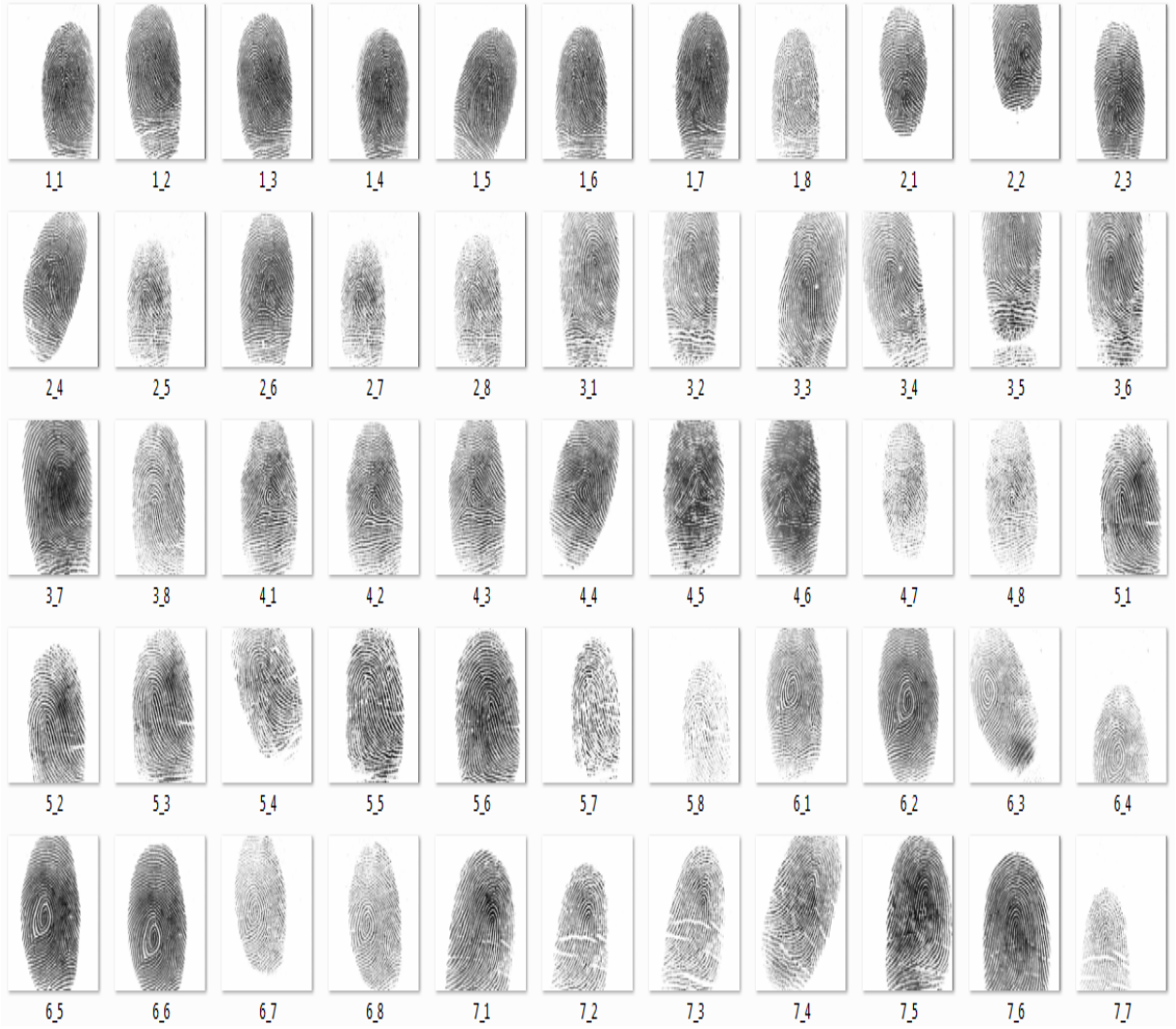



FIGURE 3.2 – Une partie de la base de données utilisée

### 3.3.2 Les attaques acceptables utilisés

Nous appliquons quelques manipulations acceptables tel que la rotation, shearing, la compression, bruits gaussien et scaling. Le **tableau 3.2** montre les paramètres de chaque attaque :

| **Attaque** | **Paramètre** |
|---|---|
| **Scaling** | **0.5** 0.7 1 1.2 1.5 |
| **Rotation** | **5** ±10 ±15 ±30 ±45 ±90 |
| **Shearing** | 0.01 0.03 **0.05** 0.07 0.09 |
| **Compression JPEG** | **10** 20 30 40 50 60 70 80 90 |
| **Bruit Gaussien** | 0.001 **0.005** 0.010 0.015 0.020 0.025 0.030 0.035 |

TABLE 3.2 – Paramètre utilisé pour chaque manipulation

— Pour calculer le TPR et FPR, nous appliquons les attaques présentées dans le **tableau 3.2** sur la base d'image (80 empreintes), en utilisant le paramètre en gras pour chaque

attaque.

### 3.3.3 Interface graphique

Dans cette partie, nous allons présenter les différentes phases de la réalisation de notre projet, Après le lancement de l'application la fenêtre d'accueil s'affiche :


FIGURE 3.3 – La représentation de page d'accueil

l'interface graphique est composée de deux bouton comme illustre **la figure 3.3** Ci-dessus :

— **Le bouton Exit** :pour quitter l'application.

— **le bouton Next :** pour accéder à la page principale.

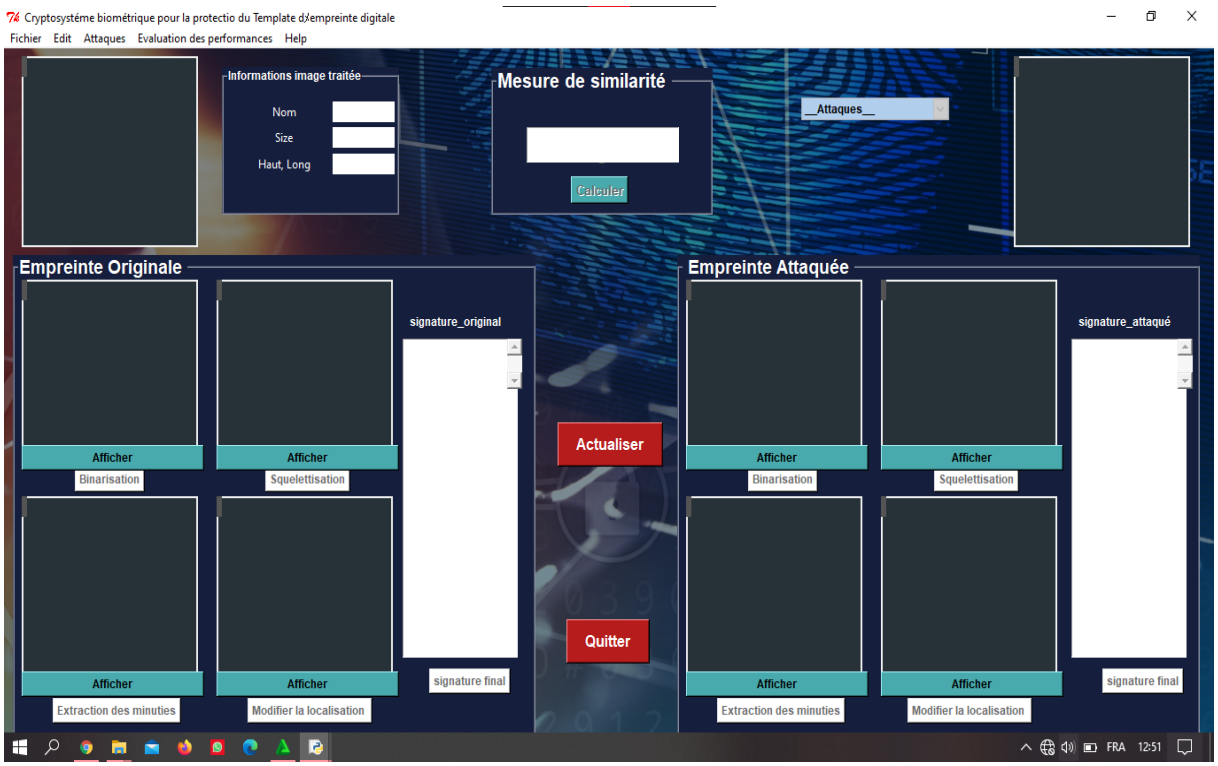



FIGURE 3.4 – La représentation de la page principale

La barre des menus de l'interface graphique (**Figure 3.4**) est contient :

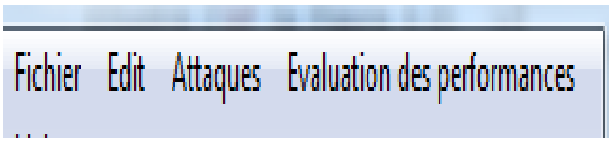



FIGURE 3.5 – La barre de menus de la page principale

- **Ficher :** contient Ouvrir image originale ( permet d'ouvrir une image pour la traité), Réinitialiser et Fermer.
- **Edit :** le menu Edit contient Actualiser et Supprimer.
- **Attaques :** le menu Attaques permet de choisir une attaque parmi les cinq attaques ( Rotation,Shearing, Compression, Bruit gaussien, Scaling).
- Evaluation des performance :permet d'afficher une fenêtre, elle est décomposée sur deux pages pour la performance et le Taux.


FIGURE 3.6 – Fenêtre pour l'evaluation des performances

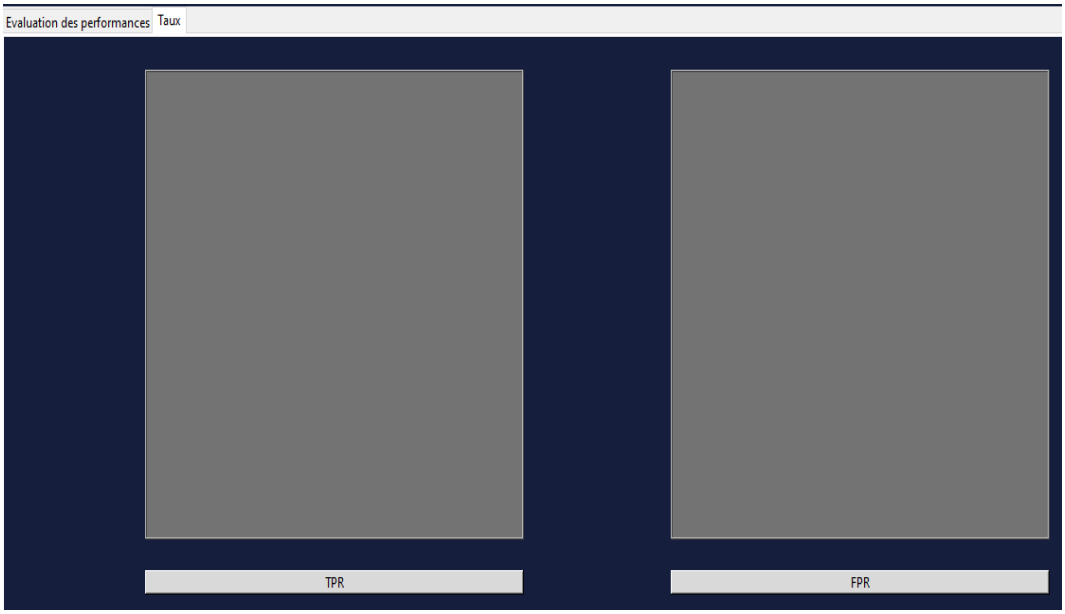



FIGURE 3.7 – Fenêtre pour l'evaluation des performances (Taux)

La **figure 3.8** montre les résultat obtenu de l'evaluation de performance :

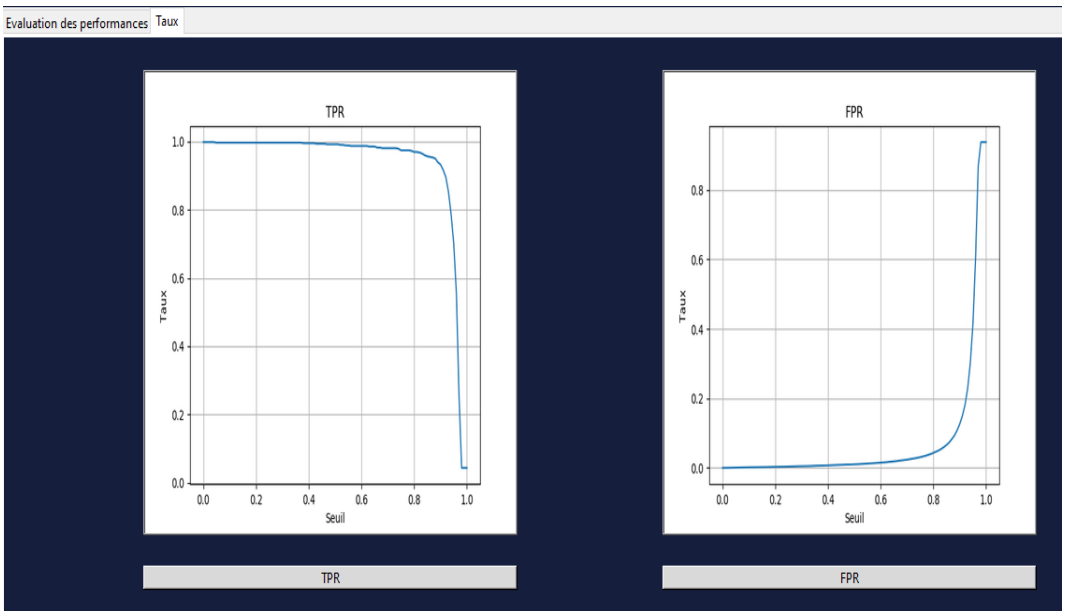



FIGURE 3.8 – Résultat de l'evaluation des performances

L'interface graphique (**figure 3.4**) est composée de trois parties (Empreinte originale, Empreinte attaquée, et mesure de similarité). Pour les deux premières étapes nous avons les boutons suivants :

- **Bouton Binarisation :** il sera actif que lorsque l'image originale est affichée pour la partie d'image originale, et lorsque l'attaque est choisi pour la partie d'image attaquée, il suffit de cliquer sur ce bouton pour obtenir l'image binarisée.
- **Bouton Squelettisation :** il sera actif que lorsque l'image binarisée est affichée, il suffit de cliquer sur ce bouton pour obtenir l'image squelattisée.
- **Bouton Extraction des minuties :** il suffit de cliquer sur ce bouton pour obtenir l'ensemble des minuties extraits.
- **Bouton décomposition :** il sera actif que lorsque les minuties sont extraits, il suffit de cliquer sur ce bouton pour obtenir la modification des minuties
- **Bouton Signature** : il sera actif que lorsque la est faite, il suffit de cliquer sur ce bouton pour obtenir la signature.
- **Bouton Afficher :** permet d'afficher les chaque image traité.

la **Figure3.8** montre les résultats obtenu les étapes de l'empreint original et attaqué.

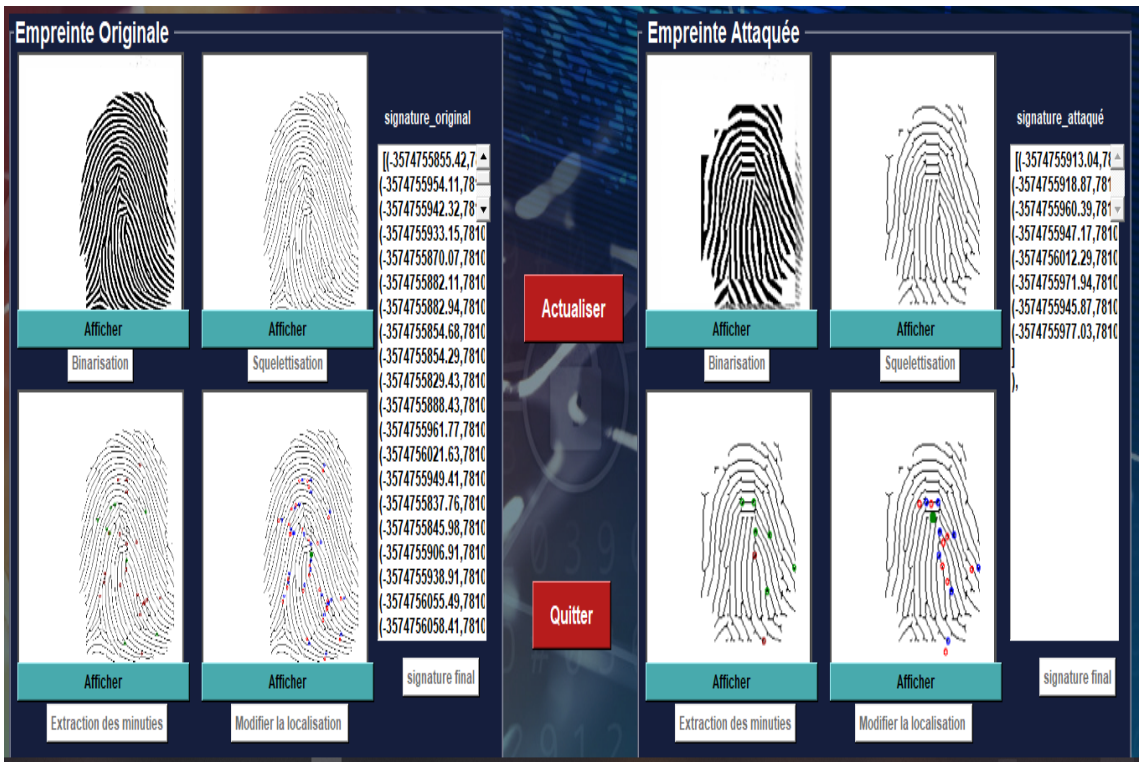



FIGURE 3.9 – le résultat obtenu de la la partie d'image originale et la partie d'image attaquée.

la partie trois de l'interface est la mesure de similarité pour calculer la distance entre la signature d'image originale et la signature attaqué, il suffit de cliquer sur le bouton calculer.

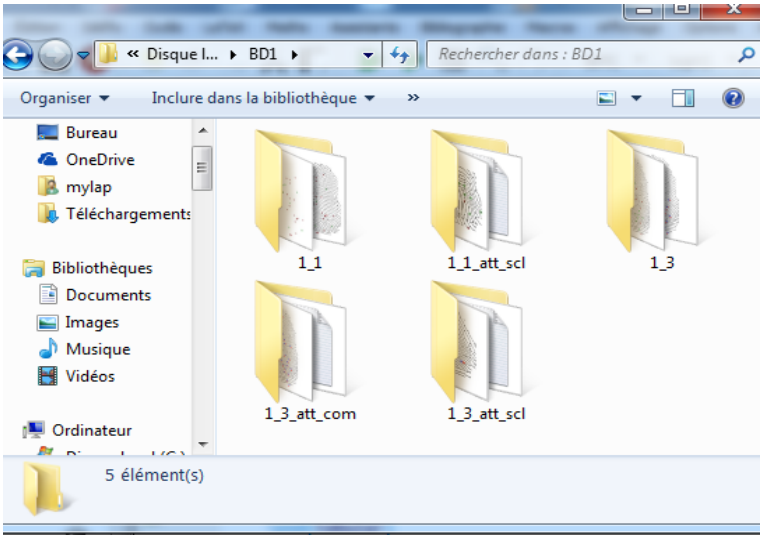



FIGURE 3.10 – Les résultats sauvegardés

## 3.4 Tests et évaluations

### 3.4.1 Evaluation des performances

Pour évaluer la performance de notre méthode contre les différentes manipulations acceptables et qui sont : la rotation, shearing, compression, bruit gaussien et scaling, nous calculons la mesure de similarité entre la signature de chaque image d'empreinte originale et sa signature attaqué, pour chaque paramètre d'attaque, ensuite nous calculons la moyenne de cette mesure. Notre teste est appliqué sur 80 images choisies de la base FVC2002, puis nous comparons les résultats avec notre seuil choisi (0.5).
Les résultat obtenus pour chaque attaque sont illustré dans les tableaux suivants :
**Rotation :**

| **Attaque** | **Paramètre** | **Moyenne de Mesure** |
|---|---|---|
| **Rotation** | 5 | 0.623 |
| | 10 | 0.589 |
| | 15 | 0.572 |
| | 30 | 0.5 |
| | 45 | 0.299 |
| | 90 | 0.501 |

TABLE 3.3 – La moyenne de similarité pour chaque paramètre de Rotation

**Shearing :**

| **Attaque** | **Paramètre** | **Moyenne de Mesure** |
|---|---|---|
| **Shearing** | 0.01 | 1.608 |
| | 0.03 | 1.042 |
| | 0.05 | 1.121 |
| | 0.07 | 1.197 |
| | 0.09 | 1.0764 |

TABLE 3.4 – La moyenne de similarité pour chaque paramètre de Shearing

**Compression :**

| Attaque | Paramètre | Moyenne de Mesure |
|---|---|---|
| **Compression** | 10 | 0.635 |
| | 20 | 0.642 |
| | 30 | 0.697 |
| | 40 | 0.671 |
| | 50 | 0.648 |
| | 60 | 0.671 |
| | 70 | 0.698 |
| | 80 | 0.697 |
| | 90 | 0.719 |

TABLE 3.5 – La moyenne de similarité pour chaque paramètre de Compression

**Bruit Gaussien :**

| Attaque | Paramètre | Moyenne de Mesure |
|---|---|---|
| **Bruit Gaussien** | 0.001 | 1.222 |
| | 0.005 | 0.972 |
| | 0.010 | 1.248 |
| | 0.015 | 1.139 |
| | 0.020 | 1.230 |
| | 0.025 | 1.172 |
| | 0.030 | 0.831 |
| | 0.035 | 1.260 |

TABLE 3.6 – La moyenne de similarité pour chaque paramètre de Bruit Gaussien

**Scaling :**

| Attaque | Paramètre | Moyenne de Mesure |
|---|---|---|
| **Scaling** | 0.5 | 0.351 |
| | 0.7 | 0.545 |
| | 1 | 0.598 |
| | 1.2 | 0.598 |
| | 1.5 | 0.421 |

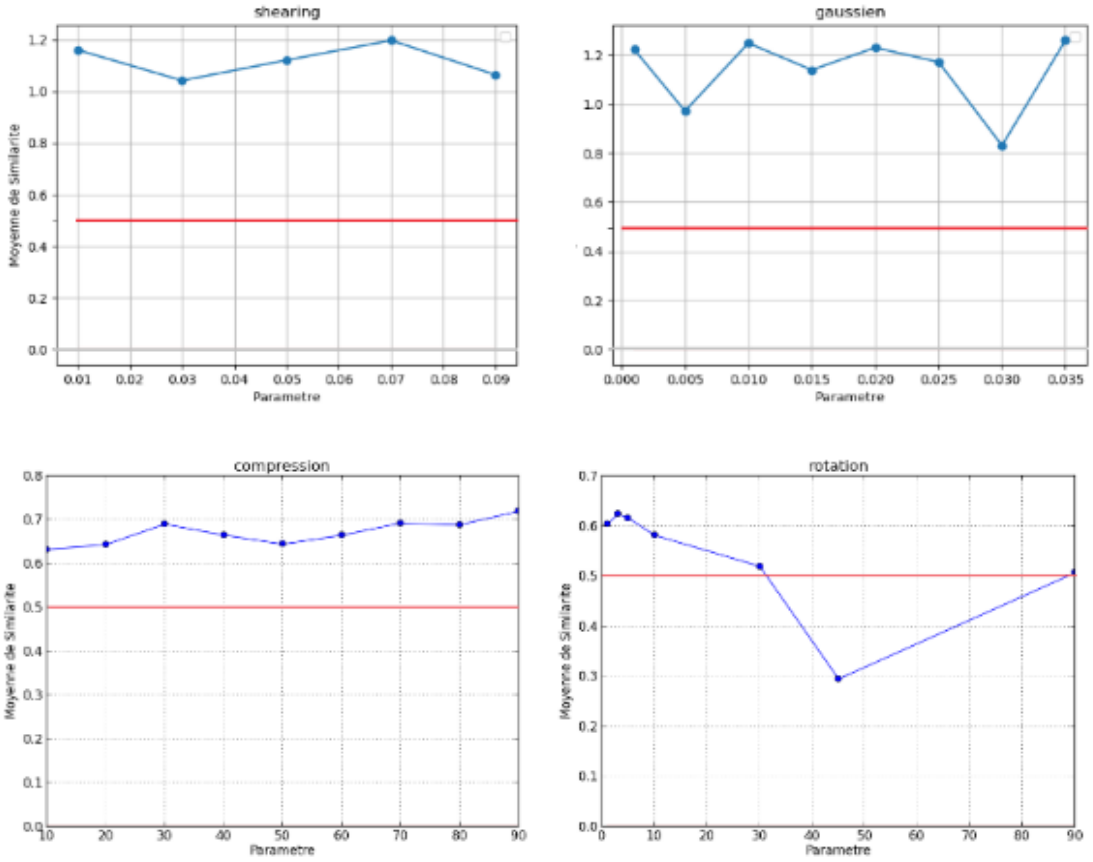
TABLE 3.7 – La moyenne de similarité pour chaque paramètre de Scaling

La moyenne de mesure de similarité générale pour toutes les attaques sont illustré dans le tableau suivant :

| **Attaque** | **Moyenne global de similarité** |
|---|---|
| Rotation | 0.505 |
| Shearing | 1.116 |
| Compression | 0.675 |
| Bruit Gaussien | 1.287 |
| Scaling | 0.502 |

TABLE 3.8 – La moyenne générale de la mesure de similarité pour chaque attaque.

Nous avons représentés les résultats sous forme des courbes. Les figures suivantes montrent les différentes courbes représentatives de la performance de notre méthode :



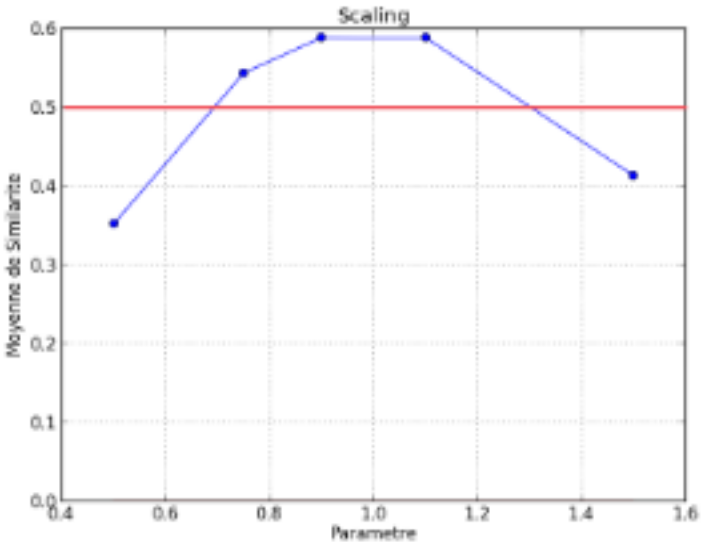



FIGURE 3.11 – Évaluation de performance par les différentes attaques.

Les résultats expérimentaux montrent que notre méthode est robuste, la moyenne de mesure de similarité calculée entre la signature des images originaux et la signature des images attaquées est supèrieure à 0.5 pour la totalité des manipulations, sauf dans le cas de la rotation 45, de scaling 0.5 et 1.5 ne donne pas des bons résultats (Moyenne de similarité 0.5), aussi la moyenne globale de mesure pour toutes les attaques est supérieure à 0.5. Donc la méthode proposée est robuste contre les attaques acceptables.

### 3.4.2 Sécurité

— **Le TPR (True positifs rate) :** pour calculer le TPR nous testons la mesure de similarité entre la signature de chaque image d'empreinte originale et les signatures de cette image avec un paramètre standard pour chaque attaque, donc nous calculons la distance entre la signature d'une image et les cinq signature manipulés de cette dernière, nous comparons le résultats avec certains seuils, dans ce cas nous obtenons le nombre de paire d'image visuellement identique considérées comme des images similaire (N similaire). Le nombre total de paires d'image visuellement identique est 400 (n identique).

— **Le FPR (False positifs rate) :** Pour calculer le FPR . Nous calculons la mesure de similarité entre la signature d'un échantillons d'empreinte original est les 72 signatures attaqués (nous ne comparons pas avec les échantillons de la même personne), les résultats obtenus nous lui comparons avec les différents seuils, nous obtenons le nombre de paires d'images différentes considérées comme des images similaires(N distinct) pour chaque seuils. Le nombre total de paires d'images différentes est 28800 (n différent).

Le **tableau 3.9** montre les résultats obtenu pour le $P_{TPR}$ et $P_{FPR}$ pour des différents seuils :

| **Seuils** | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 | 0.9 | 1.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| $P_FPR$ | 0.00 | 0.00 | 0.005 | 0.013 | 0.021 | 0.037 | 0.045 | 0.051 | 0.172 | 0.97 |
| $P_TPR$ | 1.00 | 1.00 | 1.00 | 0.995 | 0.993 | 0.991 | 0.982 | 0.975 | 0938 | 0.042 |

TABLE 3.9 – Les résultats de $P_{TPR}$ et $P_{FPR}$ par rapport au différents seuils

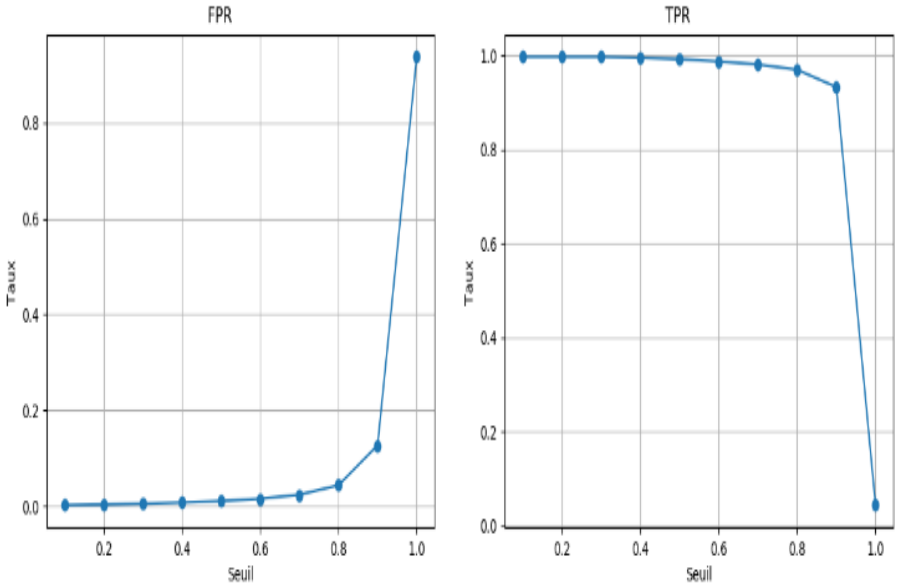



FIGURE 3.12 – Évaluation de TPR et FPR par rapport aux différentes seuils.

Les résultats montrent que le TPR et FPR est acceptable. en outre dans notre méthode le résultat avec une valeur proche à 1 est meilleur que celui avec un petit TPR, de même, la méthode avec un petit FPR est plus performante que avec un grand FPR.

## 3.5 Conclusion

Nous conclu d'après les différents tests effectués sur notre système pour évaluer ses performances à travers les différentes attaques appliquées sur les images d'empreintes digitales, que notre méthode est robuste contre quelques attaques acceptables.Également nous concluons que notre méthode ne donne pas des bons résultats dans le cas de la rotation 45, scaling 0.5 et 1.5. les résultats et leur analyse montrent l'efficacité de la technique proposée en termes de performance.