Màster Interuniversitari en Seguretat de les TIC (MISTIC)

Treball Final de Master – 2on semestre, curs 2019-2020

Seguretat en aplicacions web

Generació de claus criptogràfiques amb la veu

Òscar Raya i Casanova

Antoni Martínez Ballesté

9 de Juny de 2020

Resum

En els últims anys, la seguretat informàtica ha esdevingut un actiu molt important per a les empreses. Existeixen molts mètodes per tal de protegir l'accés a una aplicació web. Alguns d'ells però han deixat de ser fiables o són massa difícils de fer servir per a l'usuari. És per això que moltes empreses han explorat l'ús de la biometria per tal d'identificar els usuaris davant d'una aplicació web. Aquest treball se centra en el reconeixement de la veu com a mètode d'identificació. Es parteix de l'estat de l'art per tal de crear un algorisme capaç de generar una clau criptogràfica a partir de la veu. Finalment es desenvolupa una aplicació que faci servir aquest algorisme com a mètode d'identificació.

Paraules clau: clau criptogràfica, identificació de la veu, característiques de la veu

Abstract

In past years, computer security has become a very important asset for companies. There are many methods to protect access to a web application. Some of them however have ceased to be reliable or are too difficult for the user to be used. This is why many companies have explored the use of biometrics in order to identify the users in front of a web application. This work focuses on voice recognition as a method of identification. It starts from the state of the art in order to create an algorithm capable of generating a cryptographic key from the voice. Finally an application is developed that uses this algorithm as a method of identification.

Keywords: criptografic key, voice recognition, voice features

Índex

1 Introducció **3**
1.1 Context . . . 3
1.2 Objectius . . . 4
1.3 Estat de l'art . . . 5
1.3.1 Detecció de la veu . . . 5
1.3.2 Extracció de característiques . . . 5
1.3.3 Modelatge de la veu . . . 6

2 Metodologia de treball **7**
2.1 Metodologia àgil . . . 7

3 Anàlisi i disseny **8**

4 Implementació **9**
4.1 Implementació MFCC . . . 9
4.1.1 Divisió del senyal en trams petits i superposats . . . 9
4.1.2 Obtenció de les potencies de densitat espectrals de la DFT . . . 10
4.1.3 Aplicació de filtres en l'escala de mel . . . 10
4.1.4 Aplicació del logaritme en base 10 . . . 12
4.1.5 Obtenció de la transformada discreta del cosinus . . . 12
4.2 Implementació generador de claus . . . 13
4.3 Implementació aplicació Android . . . 13

5 Experiments i resultats **14**
5.1 Experiments previs . . . 14
5.2 Resultats producte final . . . 15

6 Conclusions **16**
6.1 Treball futur . . . 16

1 Introducció

1.1 Context

Les empreses i institucions han donat importància a la seguretat informàtica per tal de protegir els seus secrets i actius. Així doncs, durant molts anys, la seguretat informàtica s'ha vist com un component indispensable en aplicacions bancàries, mediques, etc. A més, en els últims anys, la població ha estat conscienciada sobre la seguretat informàtica no només a l'hora de treballar amb dades sensibles i/o importants com vénen a ser dades bancaries o mediques, sinó també en qualsevol àmbit.

Les empreses han vist, doncs, com la seguretat informàtica ha passat de ser una necessitat per l'empresa a ser un actiu que pot oferir als seus usuaris.

Diversos mètodes han estat implementats o fins i tot dissenyats per tal de millorar la seguretat en l'accés a aplicacions web. En un primer moment les empreses es van centrar a implementar millores simples per tal de millorar aquesta seguretat. Les més comunes que podem trobar són:

- Enfortiment de les contrasenyes - les aplicacions cada cop demanen requeriments més estrictes als seus usuaris a l'hora de crear contrasenyes. Avui en dia, és molt comú que una aplicació restringeixi les contrasenyes dels usuaris de manera que aquestes hagin de tenir una llargada de com a mínim 8 caràcters i contenir com a mínim un número, una majúscula, una minúscula i moltes vegades també un caràcter especial. Moltes plataformes també obliguen a els seus usuaris a canviar la contrasenya periòdicament.
- Limitar el nombre de vegades que es pot intentar accedir a l'aplicació amb un usuari determinat o des d'una IP determinada. Evitant, d'aquesta manera, atacs per força bruta.
- Ús de codis Captcha - aquests codis permeten distingir si l'usuari que vol accedir a la plataforma és un humà o una màquina. Així doncs, ens trobem amb un altre mecanisme per evitar atacs per força bruta.
- Verificació en dos passos - l'aplicació obliga a l'usuari a verificar l'accés que esta intentant fer a traves d'una altre plataforma com podria ser el correu electrònic, el telefon mòbil o un segon mètode d'identificació.

Cal dir que des que la seguretat informàtica s'ha convertit en un actiu per a les empreses aquestes busquen mètodes més simples per als seus usuaris. Així doncs moltes empreses han defugit de mètodes que poden arribar a molestar als seus usuaris com ara els codis Captcha.

És aquí on es destaquen els mètodes d'identificació mitjançant biometria. Aquests mètodes han resultat ser bastant fàcils d'utilitzar sobretot per les noves generacions.

Existeixen molts de mètodes d'identificació biomètrics. Però la majoria d'ells necessiten equipament especial per tal de ser efectius. És el cas, per exemple, de la identificació de l'iris. Si bé la majoria de gent disposa d'una càmera en el telèfon mòbil, aquesta no és prou bona per a utilitzar-se per a aquest mètode. D'altra banda, les càmeres que tenen els telèfons mòbils sí que són adequats per al reconeixement facial.

El reconeixement facial és un dels mètodes d'identificació biomètrica més fàcils d'enganyar. A més, la cara no és un caràcter tan distintiu de la persona com o poden ser l'iris o l'empremta dactilar. És per això que moltes empreses de fabricació de mòbils van començar a afegir sensors d'empremta dactilar, ja que existeix un tipus de sensor d'empremta dactilar que és relativament barat i d'aquesta manera afegien valor al seu producte amb un mecanisme de seguretat ràpid, segur i relativament barat.

Així i tot, els mòbils que disposen de sensor d'empremta dactilar són molt pocs. Així doncs és necessari oferir un altre mètode d'identificació biomètric alternatiu. Com a tret biomètric la veu presenta una major fiabilitat que la cara. A més, com que els mòbils són una eina que el seu principal ús històric és el de la comunicació oral, disposen de micròfons de qualitat ideals per al reconeixement de la veu. D'altra banda no tots els mòbils tenen càmeres de prou qualitat per al reconeixement facial.

Per aquests motiu, en aquest treball ens centrarem amb el reconeixement de veu com a mètode d'identificació.

1.2 Objectius

El propòsit d'aquest treball és establir un mètode d'identificació de la veu ràpid, fiable i a l'abast de tothom utilitzant un algorisme de **hash** per tal de generar una clau. Per tal d'aconseguir-ho s'han establit els següents objectius:

O1. Estudiar quines característiques de la veu humana es poden utilitzar per obtenir claus criptogràfiques, a partir d'una anàlisi de la literatura científica.

O2. Dissenyar un mètode amb el qual obtenir una contrasenya a partir de la veu: La idea és enregistrar una paraula, extreure'n característiques, passar-les per un hash i així obtenir la contrasenya. El sistema ha de ser prou robust perquè sempre o gairebé sempre generi la mateixa contrasenya.

O3. Implementar una eina per posar en pràctica el mètode dissenyat: Fer un web o una aplicació que permeti demostrar el mètode dissenyat. L'objectiu serà desxifrar (o poder accedir) a arxius o a contingut.

O4. Provar el que s'ha implementat: Descriure com s'entrena el sistema i provar el sistema amb diferents usuaris i mostrar estadístiques pel que fa a intents, encerts, etc.

1.3 Estat de l'art

El reconeixement de veu és una tecnologia que ens permet identificar l'usuari d'aquesta. La veu té tonalitats, sonoritats i timbres molt variats entre diferents parlants. Això fa que la veu sigui un mètode per a poder identificar una persona. Durant molts anys els autors han investigat aquest camp i sembla que l'estat de l'art ha quedat bastant consolidat. Així doncs la majoria d'autors presenten sistemes de reconeixement de veu dividits en tres etapes essencials: detecció de la veu, extracció de característiques i modelatge de la veu.

1.3.1 Detecció de la veu

La detecció de la veu o voice activity detection (VAD) és una tècnica que permet diferenciar la presencia o absència de veu humana. Aquesta tècnica és utilitzada per moltes aplicacions de comunicació per tal d'estalviar paquets de dades enviant-se quan l'interlocutor es troba parlant. També és capaç de distingir entre la veu i el soroll de fons. Autors com Athulya et al. [1] o Ong i Tan [2] mostren la importància d'utilitzar VAD a l'hora de fer sistemes de reconeixement de veu per tal d'utilitzar només aquelles parts que contenent veu i netejar-les del soroll extern.

1.3.2 Extracció de característiques

A part d'informació per poder identificar la persona, la veu conte moltíssima més informació. En primer lloc, trobaríem el processament de la veu per saber allò que està dient la persona. És a dir, que la maquina pugui entendre l'usuari. Feng [3] mostra l'ús de la veu en aquest camp. Autors com Chen i Yang [4] també fan ús de la veu per detectar les diferents emocions d'una persona o autors com Zvarevashe i Olugbara [5] l'utilitzen per detectar el gènere. Fins i tot s'ha arribat a investigar per tal d'utilitzar-la a l'hora de detectar patologies tal com mostren Al-Nasheri et al. [6].

Tots aquests autors utilitzen diferents característiques de la veu en funció del camp que tracten. És doncs, essencial saber quines característiques de la veu ens poden ajudar a reconèixer l'interlocutor.

La gran majoria d'autors, a l'hora de desenvolupar un sistema de reconeixement de veu, tenen en consideració els mel-frequency cepstral coefficients (MFCCs) com a principal característica distintiva de la veu [7 - 9]. Altres autors també utilitzen els linear predictive coefficients (LPCs) sols [10, 11] o com a característica complementaria als MFCCs [1, 12].

Alguns autors han usat altres característiques com ara els power normalized cepstral coefficients (PNCCs) i la spectral centroid frequency (SCF) proposats per Wang i Wang [13], les cosine distance features (CDFs) proposades per Balasingam i Kumar [14], o els gammatone frequency cepstral coefficients (GFCC) proposats per Athulya et al. [1].

Mel-frequency cepstral coefficients

Els coeficients cepstrals en les freqüències de mel són coeficients que representen la parla i estan basats en la percepció auditiva humana. Els MFCCs han estat l'estat de l'art pel que fa a reconeixement de veu des dels anys vuitanta. Aquests coeficients es poden calcular amb els següents passos:

1. Dividir el senyal en trams petits i superposats.
2. Obtenir les potencies de densitat espectral (PSDs) de la transformada discreta de Fourier (DFT) de cada tram.
3. Aplicar filtres en l'escala de mel a les PSDs obtingudes.
4. Aplicar el logaritme en base 10 a les freqüències de mel.
5. Obtenir la transformada discreta del cosinus (DCT) dels logaritmes anteriors.

Els MFCCs són les amplituds de l'espectre obtingut amb la DCT. Normalment se solen utilitzar els coeficients del 2 al 13 i es descarten la resta.

1.3.3 Modelatge de la veu

Un cop extretes les característiques que aporten informació sobre d'identitat de la persona s'ha de crear un model que sigui capaç d'identificar la persona a partir de les característiques de la seva veu.

A l'hora d'escollir un tipus de model, l'estat de l'art es decanta clarament pels gaussian mixture models (GMMs) tal com fan Chakroun et al. [15] i molts altres autors [1, 11, 13].

Tot i que l'estat de l'art està dominat pels GMMs, de tant en tant alguns autors utilitzen altres alternatives. Com ara Sadeghi i Marvi [9] que el 2017 van proposar fer servir probabilistic neural networkss (PNNs).

Gaussian Mixture Model

Els GMMs són models probabilístics que busquen representar diferents subconjunts (o clústers) en distribucions normals. Aquesta classe de models pertanyen als mètodes d'aprenentatge no vigilat. És a dir, són models capaços d'assignar una dada a un conjunt de dades sense estar aquestes etiquetades.

Els GMMs es caracteritzen per tenir un paràmetre que els indica quants subconjunts han d'extreure. Així doncs, el model agruparà les dades en tants subconjunts distribuïts de forma normal com li hàgim indicat.

2 Metodologia de treball

S'ha seguit una metodologia de treball àgil.

2.1 Metodologia àgil

Les metodologies àgils són metodologies que es basen en l'adaptabilitat de qualsevol canvi per tal d'augmentar les possibilitats d'èxit del projecte que s'està desenvolupant.

Moltes metodologies àgils minimitzen el risc fent un desenvolupament en iteracions incrementals. Cadascuna d'aquestes iteracions busquen ser prou completes per a poder-se alliberar al mercat. A més, com que cada iteració representa un producte acabat també es realitzen proves al final de cada iteració. Un cop acabada una iteració es revaluen les prioritats del projecte i s'escull el següent objectiu a desenvolupar.

En el cas del nostre projecte s'han decidit petits objectius de desenvolupament iteratius i incrementals entre ells. Aquests objectius, i la seva prioritat, han anat canviant al llarg del projecte.

3 Anàlisi i disseny

El producte final serà una aplicació Android en la qual l'usuari hagi d'utilitzar la seva veu per tal d'accedir-hi. Aquesta aplicació demanarà a l'usuari registrar-se utilitzant la seva veu.

Com que seguim una metodologia àgil s'han marcat diversos objectius de desenvolupament els quals poden arribar a ser eliminats o modificats a mesura que avança el desenvolupament. Aquests objectius, després de revisar l'estat de l'art, són:

- Implementació d'un algorisme per detectar la presència de veu (VAD).
- Implementació d'un algorisme que extregui característiques. Més concretament s'extrauran els MFCCs.
- Implementació d'un algorisme per extreure una clau a partir de les característiques extretes. Aquest algorisme es basarà en els GMMs.
- Implementar una aplicació amb Android que sigui capaç de capturar la veu de l'usuari a través del micròfon. S'implementarà un mòdul de captura de la veu completament diferenciat per tal de fer-lo servir en diverses pantalles.
- Establir una pantalla de registre a l'aplicació d'Android en la qual es registra la veu de l'usuari i se n'obté una clau diversos cops fins que es decideix quina clau ha d'utilitzar l'usuari (es buscarà utilitzar la clau que més vegades s'hagi obtingut).
- Establir una pantalla d'inici de sessió a l'aplicació d'Android en la qual generi una clau a partir de la veu de l'usuari i es comprovi si l'usuari està registrat al sistema.

De tots aquests objectius s'ha decidit començar per l'extracció de característiques, ja que és el primer que necessitem si volem treballar amb dades.

Mentre és desenvolupada el projecte, i per falta de temps, s'ha decidit prescindir del sistema de detecció de presència de veu (VAD), ja que no és una funcionalitat essencial per al projecte. A més inicialment no és plantejava utilitzar la clau que més cops s'hagi generat en el registre sinó que el plantejament era combinar les característiques obtingudes de tots els registres de veu per tal de generar una única clau.

4 Implementació

Com hem indicat a l'apartat anterior s'ha començat el desenvolupament del projecte implementant l'extracció de MFCCs. Seguidament s'ha començat la implementació de l'aplicació Android de manera que es pogués capturar la veu. La següent implementació ha estat la de l'algorisme que genera la clau a partir del MFCCs. Finalment s'ha procedit a implementar el registre i l'accés a l'aplicació Android.

4.1 Implementació MFCC

Com que el nostre objectiu final és tenir una aplicació Android que utilitzi la veu com a metode d'identificació hem de ser capaços de extreure els MFCCs en un entorn Java (ja que és el llenguatge majoritàriament utilitzat per fer aplicacions Android).

Per tal d'implementar l'extracció de MFCCs en Java s'han seguit els passos prèviament comentats a l'estat de l'art mentre és feien comparatives amb les funcions pròpies de Matlab.

A continuació descriurem cadascun dels passos i com s'han implementat.

4.1.1 Divisió del senyal en trams petits i superposats

Un senyal d'àudio es troba constantment canviant cosa que dificulta el modelatge de la veu. Així doncs, per tal de fer possible el modelatge, assumirem que el senyal no canvia tant per a períodes de temps curts. A l'hora de calcular els MFCCs normalment es divideix el senyal en trossos superposats de 20 a 40 mil·lisegons. La figura 1 mostra aquest procés.

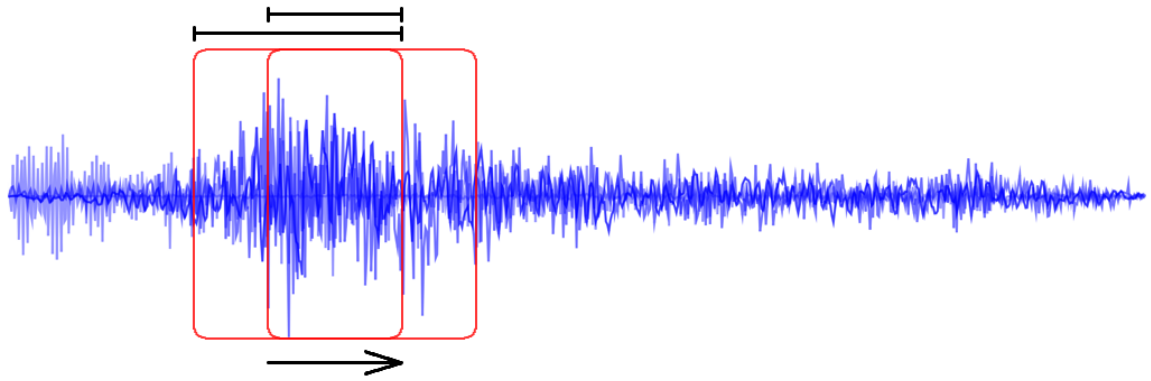

Figura 1: Representació d'una finestra lliscant superposada que divideix el senyal.

En el nostre cas s'ha decidit utilitzar la mateixa mida de finestra i de superposició que s'utilitza a Matlab per a fer el càlcul dels MFCCs, és a dir, 30 i 20 mil·lisegons respectivament. Aquestes mides es calculen a partir de la freqüència de mostreig i procurant que la superposició sigui menor a la mida de finestra.

4.1.2 Obtenció de les potencies de densitat espectrals de la DFT

Un cop dividit el senyal en finestres obtindrem la potència espectral de cadascuna d'elles. Aquesta operació es fa per tal d'emular com l'oïda humana identifica una persona.

Per fer-ho primer s'ha d'obtenir la DFT de cada finestra. La DFT és una transformació d'un senyal x a un senyal X on cada mostra del senyal compleix la fórmula 1 on N és el nombre de mostres del senyal.

$$X_n = \sum_{t=0}^{N-1} x_t e^{-knt} \qquad k = \frac{2\pi i}{N} \qquad n = 0, ..., N-1 \tag{1}$$

Fixem-nós que k és un nombre complex per tant el resultat de la DFT seran nombres complexos. Java no disposa d'un tipus bàsic per als nombres complexos així que se n'ha implementat un amb les operacions bàsiques per poder operar amb nombres complexos.

Per tal de millorar l'eficiència de l'algorisme s'ha reduït el càlcul de e^{knt} utilitzant la fórmula d'Euler: $e^{xi} = \cos x + i \sin x$. Per al nostre cas $x = -\frac{2\pi}{N} nt$.

Un cop s'ha obtingut la DFT del senyal només ens queda obtenir les potències de densitat espectrals. Per fer-ho només cal multiplicar totes les mostres pel seu conjugat (formula 2).

$$X_n = x_n \overline{x_n} \qquad n = 0, ..., N-1 \tag{2}$$

La multiplicació d'un nombre complex pel seu conjugat es pot reduir a la suma dels quadrats de la part real i imaginaria tal com és mostra al següent desenvolupament:

$$\begin{aligned}
z\overline{z} &= (a+bi)(a-bi) \\
z\overline{z} &= a^2 + abi - abi - (bi)^2 \\
z\overline{z} &= a^2 - b^2 i^2 \\
z\overline{z} &= a^2 - b^2(-1) \\
z\overline{z} &= a^2 + b^2
\end{aligned}$$

4.1.3 Aplicació de filtres en l'escala de mel

L'oïda no és capaç de diferenciar entre dues freqüències properes entre elles. No sols això sinó que com més altes siguin les freqüències més difícil esdevé per l'oïda diferenciar-les. És per això que més que treballar amb l'energia per a cadascuna de les freqüències ens interessa més obtenir quanta energia existeix al voltant de determinades freqüències.

Els filtres en l'escala de mel ens ajudaran a obtenir el que volem. El primer que hem de fer és decidir el nombre de filtres que volem fer servir, és a dir, quantes agrupacions volem. Normalment s'utilitzen entre 20 i 40 filtres essent 26 filtres l'estàndard.

Per crear els filtres s'escollirà una freqüència de començament, normalment 0, i una de fi, de com a màxim la meitat de la freqüència de mostreig. Aquest màxim respon al fet que per poder avaluar una freqüència necessitem el doble de freqüència de mostreig. El següent pas és passar aquestes freqüències a l'escala de mel. L'equació 3 mostra com transformar la freqüència a l'escala de mel i la 4 mostra com fer l'operació inversa.

$$M(f) = 2595 \log_{10}(1 + f/700) \tag{3}$$

$$M^{-1}(m) = 700(m/2595^{10} - 1) \tag{4}$$

Crearem tans punts equidistants en l'escala de mel com filtres vulguem entre les freqüències de començament i de finalització. Aquests punts seran el centre dels nostres filtres. Si volem 26 filtres obtindrem un total de 28 punts (26 centrals, el punt mínim i el punt màxim).

Revertirem l'escala d'aquests 28 punts fent us de l'equació 4. I, com que volem aplicar aquests filtres al resultat d'una DFT canviarem l'escala d'aquests punts per tal d'adaptar-se a la mida de la DFT.

El càlcul d'aquests punts s'ha fet directament amb l'equació 5 on N és el nombre de filtres a utilitzar, $NFFT$ és la mida de la DFT, fs és la freqüència de mostreig i M^{-1} és l'equació 4.

$$\begin{aligned} f(k) &= \lfloor (NFFT + 1) * M^{-1}(lowFreq + k * step)/fs \rfloor \\ step &= (highFreq - lowFreq)/(N + 1) \\ &\text{per} \quad k = 0, ..., N + 1 \end{aligned} \tag{5}$$

A continuació ja podem crear els N filtres utilitzant els punts anteriors i l'equació 6.

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \frac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \leq k \leq f(m) \\ \frac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \leq k \leq f(m+1) \\ 0 & k > f(m+1) \end{cases} \tag{6}$$
$$\text{per} \quad m = 1, ..., N \quad \text{i} \quad k = 0, ..., NFFT - 1$$

Finalment aplicarem els filtres per cada finestra utilitzant l'equació 7 on N és el nombre de filtres, L és la mida de la finestra i H_m és un filtre.

$$X(m-1) = \sum_{k=0}^{L-1} x(k) * H_m(k) \quad \text{per} \quad m = 1, ..., N \tag{7}$$

Observem com en aquest pas la dimensionalitat s'ha reduït enormement.

4.1.4 Aplicació del logaritme en base 10

Els humans tampoc percebem la sonoritat de manera linear. És per això que de l'energia obtinguda aplicarem un logaritme. En el nostre sistema s'ha decidit per aplicar un logaritme en base 10 però també s'haurien pogut fer servir altres bases.

4.1.5 Obtenció de la transformada discreta del cosinus

Les energies obtingudes dels filtres en l'escala de mel estan sobreposades i altament correlacionades entre si. Per tal de descorrelacionar-les aplicarem una DCT a cada finestra.

La DCT és una transformada basada en la DFT la qual només treballa en nombres reals al contrari que la DFT que treballa amb nombres complexos. L'equació 8 defineix la DCT on N és el nombre de coeficients generats desprès d'aplicar els filtres de mel.

$$\begin{aligned} F(k) &= \frac{2c(k)}{N} \sum_{j=0}^{N-1} f(j) \cos\left[\frac{(j+0.5)k\pi}{N}\right] \\ \text{on} \quad c(k) &= \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & 0 < k < N \end{cases} \\ \text{per} \quad k &= 0, ..., N-1 \end{aligned} \tag{8}$$

Com es comentava a l'estat de l'art normalment es solen mantenir només 12 dels 26 coeficients s'ignora el primer coeficient i els restants a partir del 12.

4.2 Implementació generador de claus

El nostre algorisme per generar claus a partir dels MFCCs està basat en els GMMs. Aquesta classe de models ajusten cada característica, en aquest cas els MFCCs, a una distribució normal de manera que es formaran agrupacions amb les diverses distribucions normals. Les distribucions normals es poden definir amb dos paràmetres: mitjana i desviació estàndard. Quan el sistema rep una nova mostra calcularà aquests dos paràmetres per cada característica i mirarà quina agrupació s'ajusta més bé a la mostra.

Prenent això com a partida s'ha decidit implementar un algorisme que calculi aquests dos paràmetres, mitjana i desviació estàndard, per a cadascun dels coeficients i generi una clau utilitzant-los. La idea és assignar una lletra als diferents valors de cada paràmetre de manera que una lletra pugui representar un conjunt de valors prou ampli perquè es generi la mateixa clau sempre però prou estret perquè altres usuaris no tinguin la mateixa clau.

Aquest algorisme requereix experimentació per tal d'ajustar-lo adequadament.

4.3 Implementació aplicació Android

Abans d'haver implementat l'algorisme generador de claus es va començar l'aplicació d'Android. Això es va fer per tal de comprovar si tindríem alguna limitació a l'hora de fer l'algorisme que genera les claus.

S'ha creat doncs una classe que permet capturar el so en segon pla. Aquesta classe té 3 funcions per tal de controlar-la: `start()`, `stop()` i `getRecordedAudio()`. Utilitzant aquestes 3 funcions podem fer que l'aplicació gravi mentre prenem un botó i quan parem de prémer pari la gravació i la processi.

Un cop implementat l'algorisme per tal de generar claus podem continuar desenvolupant l'aplicació d'Android. Aquest algorisme s'utilitzarà en dos àmbits diferents: registre i accés.

El registre és la fase d'entrenament de la nostra aplicació un cop dins es demanarà a l'usuari que enregistri la seva veu diversos cops fins que s'hagi pogut escollir una clau. Una clau és acceptada per l'aplicació un cop aquesta ha estat generada 3 cops. Així doncs mentre l'usuari no aconsegueixi generar la mateixa clau 3 cops l'aplicació continuarà demanant-li que enregistri la seva veu.

L'accés és la fase de test de la nostra aplicació. L'usuari enregistrarà la seva veu amb la qual es generarà una clau. Un cop generada aquesta clau es comprova si coincideix amb la clau escollida en el registre. En cas afirmatiu l'usuari és redirigit a la pantalla final altrament l'aplicació indica a l'usuari que ho torni a provar.

En les dues fases l'aplicació espera gravacions de més de 2 segons. En cas que la gravació sigui més curta serà automàticament descartada.

5 Experiments i resultats

L'algorisme generador de claus s'ha testejat abans de posar-lo a l'aplicació d'Android per tal de trobar els paràmetres adequats. Un cop fet això s'ha passat l'algorisme a l'aplicació d'Android i s'ha comprovat el seu rendiment.

5.1 Experiments previs

Els primers experiments s'han fet utilitzant el data-set TIMIT [16]. Aquest data-set és una col·laboració entre diverses organitzacions i s'ha fet servir molt en els últims anys pel que fa a identificació de la veu.

El data-set TIMIT conte 6300 mostres de veu de 630 persones de parla anglesa. Cada parlant ha enregistrat 10 mostres de veu diferents d'una durada entre 5 i 10 segons. A més els parlants estan distribuïts en 8 regions diferents dels estats units. Un 70% dels parlants són homes i la resta són dones.

Aquestes dades han estat prèviament processades amb Matlab per tal d'oferir una lectura rapida per al programa fet amb Java (alguns arxius `.wav` contenien còdecs que eren difícils de llegir per a Java sense utilitzar llibreries).

En aquesta etapa s'han fet diversos tests per tal de veure l'eficiència de l'algorisme. En fer els primers testos s'ha marcat un temps d'execució mitjana d'uns 7 segons per mostra.

El primer test s'ha fet amb la intenció de determinar el correcte funcionament de l'algorisme de resum. S'ha observat que utilitzar només la mitjana de cada coeficient descartant la desviació estàndard oferia més robustesa en l'algorisme. Així doncs els següents tests s'han efectuat utilitzant només la mitjana.

S'han generat claus per a cada mostra i s'han avaluat els resultats segons la regió en termes de robustesa i singularitat de les claus generades. La taula 1 mostra la robustesa i la singularitat de les claus generades per a cada regió.

Regió	Participants	Robustesa	Claus úniques	Claus compartides
1	49	16,33%	43	3
2	102	18,33%	61	19
3	102	17,94%	61	18
4	100	18,30%	63	13
5	98	18,67%	66	13
6	46	18,48%	38	4
7	100	17,30%	63	17
8	33	16,67%	24	4

Taula 1: Robustesa i singularitat de les claus generades

La robustesa s'ha calculat observant quantes vegades una persona és capaç de crear la mateixa clau (s'utilitza la clau que ha creat més vegades com a referència). La diversitat s'ha calculat observant quantes claus úniques i compartides es generen.

Com es pot observar a la taula 1 la robustesa es troba al voltant d'un 18% el que significa que l'algorisme és capaç de generar la mateixa clau per a la mateixa persona un 18% de les vegades. Quant a la singularitat es pot observar com el 66,5% dels participants aconsegueixen obtenir una clau única dins la seva regió.

5.2 Resultats producte final

El producte final es tracta d'una aplicació d'Android que utilitza el generador de claus per a identificar l'usuari a través de la veu. Els experiments amb l'aplicació s'han dut a terme amb un únic subjecte.

A causa de la poca robustesa mostrada en els experiments anteriors s'ha optat per intentar fer servir la mateixa frase durant tot el registre i l'accés. D'aquesta forma es busca augmentar la robustesa de l'algorisme en utilitzar la mateixa frase.

En aquest punt s'ha pogut observar un augment en la robustesa fins a arribar al 33% de robustesa.

6 Conclusions

Com s'ha vist en els resultats un usuari requeriria una mitjana de 6 intents abans de poder accedir a l'aplicació i una mitjana de 17 intents per tal de registrar-se. La cosa no millora gaire a l'hora d'utilitzar sempre la mateixa frase on es necessitarien una mitjana de 9 intents per poder fer el registre i 3 per accedir-hi.

Podem concloure doncs que amb els resultats obtinguts aquest mètode no és prou robust per a poder-se utilitzar com a mètode identificatiu davant d'una aplicació. A més l'algorisme tampoc és prou sensible a les diverses veus, ja que per a grups de 100 usuaris diferents 40 d'ells compartirien clau amb alguna altra persona.

Fent una anàlisi del mètode utilitzat sembla que utilitzar la veu per generar una clau és un repte difícil que poder no té resposta. Això es deu principalment al fet que les diferents característiques extretes es troben molt sobreposades entre els diferents usuaris i resumint-les es perd la informació necessària per a poder distingir realment entre un usuari i un altre.

6.1 Treball futur

A continuació s'exposa un seguit de tasques que és creuen necessàries o que be podrien fer que l'algorisme de generador de claus obtingues resultats prou bons per tal d'utilitzar-se en un producte.

- Implementació de VAD. Les mostres utilitzades per tal de fer els primers experiments estaven retallades per tal contenir només el moment en què el subjecte parla. Així i tot, els subjectes fan pauses quan parlen. En l'aplicació d'Android ni tan sols es talla el principi i el final de la mostra. Així doncs un algorisme de detecció de presència de la veu podria millorar en gran manera la robustesa i sensitivitat de l'algorisme generador de claus en utilitzar només aquelles parts amb veu.
- Millorar l'algorisme que calcula la DFT. El càlcul d'aquesta transformada és el que posa més carrega en l'aplicació, ja que és d'ordre n^2 per a cada finestra. Existeix un algorisme anomenat fast fourier transform (FFT) que aprofita les similituds en les matrius per a fer el càlcul molt més ràpid. El mateix es pot fer amb l'algorisme que calcula la DCT que també és d'ordre n^2 per a cada finestra. Així i tot, aquest últim no suposa una càrrega tan gran, ja que s'utilitza després de reduir la dimensionalitat.
- Afegir el càlcul dels *deltas* i els *deltaDeltas* als MFCCs. Els *deltas* són coeficients que representen la velocitat en la qual canvia l'energia calculada amb els filtres de mel. Els *deltaDeltas* calculen la velocitat en què canvien els *deltas*. Utilitzar aquests coeficients podria millorar la sensitivitat de l'algorisme de resum.

- Explorar l'ús d'altres característiques per tal de millorar l'eficiència de l'algorisme com ara els LPCs, els PNCCs, els GFCCs, etc.
- Afegir pesos als diversos coeficients per tal de poder controlar la sensitivitat de cadascun d'ells a l'hora de fer el resum Aplicar pesos en els diferents coeficients a l'hora de fer el resum per tal de controlar la sensitivitat de cadascun d'ells.

Referències

[1] M. S. Athulya, Vinayshankar i P. S. Sathidevi, «Mitigating effects of noise in Forensic Speaker Recognition», a *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 2017, pàg. 1602 - 1606.

[2] W. Q. Ong i A. W. C. Tan, «Robust voice activity detection using gammatone filtering and entropy», a *2016 International Conference on Robotics, Automation and Sciences (ICORAS)*, 2016, pàg. 1 - 5.

[3] J. Feng, «A general framework for building natural language understanding modules in voice search», a *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pàg. 5362 - 5365.

[4] L. Chen i Y. Yang, «Emotional speaker recognition based on i-vector through Atom Aligned Sparse Representation», a *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pàg. 7760 - 7764.

[5] K. Zvarevashe i O. O. Olugbara, «Gender Voice Recognition Using Random Forest Recursive Feature Elimination with Gradient Boosting Machines», a *2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD)*, 2018, pàg. 1 - 6.

[6] A. Al-Nasheri, G. Muhammad, M. Alsulaiman, Z. Ali, K. H. Malki, T. A. Mesallam i M. Farahat Ibrahim, «Voice Pathology Detection and Classification Using Auto-Correlation and Entropy Features in Different Frequency Regions», *IEEE Access*, vol. 6, pàg. 6961 - 6974, 2018.

[7] S. Tripathi i S. Bhatnagar, «Speaker Recognition», a *2012 Third International Conference on Computer and Communication Technology*, 2012, pàg. 283 - 287.

[8] Z. Wanli i L. Guoxin, «The research of feature extraction based on MFCC for speaker recognition», a *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*, 2013, pàg. 1074 - 1077.

[9] M. Sadeghi i H. Marvi, «Optimal MFCC features extraction by differential evolution algorithm for speaker recognition», a *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*, 2017, pàg. 169 - 173.

[10] M. Chougala i S. Kuntoji, «Novel text independent speaker recognition using LPC based formants», a *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016, pàg. 510 - 513.

[11] M. Dai, G. Dai, Y. Wu, Y. Xia, F. Shen i H. Zhang, «An Improved Feature Fusion for Speaker Recognition», a *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, 2019, pàg. 183 - 187.

[12] Y. Zhou, J. Wang i X. Zhang, «Research on Speaker Recognition Based on Multifractal Spectrum Feature», a *2010 Second International Conference on Computer Modeling and Simulation*, vol. 1, 2010, pàg. 463 - 466.

[13] N. Wang i L. Wang, «Robust speaker recognition based on multi-stream features», a *2016 IEEE International Conference on Consumer Electronics-China (ICCE-China)*, 2016, pàg. 1 - 4.

[14] M. D. Balasingam i C. S. Kumar, «Refining Cosine Distance Features for Robust Speaker Verification», a *2018 International Conference on Communication and Signal Processing (ICCSP)*, 2018, pàg. 0152 - 0155.

[15] R. Chakroun, L. Beltaïfa Zouari, M. Frikha i A. Ben Hamida, «Improving text-independent speaker recognition with GMM», a *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, 2016, pàg. 693 - 696.

[16] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren i V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*, ISBN: 1-58563-019-5, 1993.