Manuscript received Dec. 25, 2007; revised Apr. 27, 2008

Speaker Recognition Techniques for Remote Authentication of Users in Computer Networks

[1,2]ADRIANO PETRY, [1]SIDCLEY S. SOARES,
[1]GILBERTO F. MARCHIORO, [1]ANALÚCIA S. M. DE FRANCESCHI

[1]Área de Tecnologia e Computação
Universidade Luterana do Brasil
Rua Miguel Tostes 101 – Canoas – RS
BRAZIL

[2]Curso de Engenharia de Sistemas Digitais
Universidade Estadual do Rio Grande do Sul
Estrada Santa Maria 2300 – Guaíba – RS
BRAZIL
adpetry@pesquisador.cnpq.br

Abstract: - This work shows the main characteristics of a system designed to improve security in Internet transactions using a biometric measure: the speech. First, it is given a general overview of the proposed system, developed using Java and C++ programming languages. After that, the main characteristics of the system building blocks are shown, focusing on the techniques for speaker recognition, database, client/server communication and management, and user interface application. The processing load is distributed in this system. The client side is basically responsible for speech acquisition and feature parameters extraction. The server side is responsible for voiceprint generation (on the training phase) or similarity measure between a voiceprint and some feature parameters (on recognition phase). In order to reduce the amount of data transmission, this distributed architecture presents the advantage of transmitting only the feature parameters over the network, instead of the complete wave signal. The acceptance or rejection of a speech identity is based on a pre-defined similarity threshold. Tests using speech database composed of 50 different speakers presented equal error rate (EER) of approximately 1.3%. At the end, the main results obtained from the interaction of the system with several different users are discussed. Besides that, accuracy test results are shown, and future works are addressed.

Key-Words: - Speaker recognition, Speech processing, Biometrics, Authentication, Computer networks, Internet security

1 Introduction

In recent years, enterprises from a large variety of sectors had searched new ways to increase the offer of products and services for their clients. The use of the Internet for online business operations is assuring wider visibility for these enterprises. The need of privacy and security of exchanged information has grown with the commercial use of the Internet. Also, the existence of more reliable user authentication methods is a condition for the increase of e-commerce popularity.

The user authentication may be reached through the use of different techniques, each one of them presents some particular characteristics. An authentication with a secret information (or password) is used in majority of the systems that require some kind of authentication. This methodology has vulnerabilities such as the user password may be easy to discover, or the secret information can be stolen using computer viruses. Another way to avoid fraudulent activities is the employment of physical devices, like magnetic or smart cards, to provide the authentication. Actually, this kind of protection has been adopted in systems that need to ensure a higher degree of security, as the banking transactions. However, the access device can still be deviate (it may be lost or stolen) and the password can be discovered, allowing a possible unauthorized access. The use of biometrics for user authentication appears as an alternative to minimize the chances of security failure. The biometric analysis with no human intervention (automatic) may be also used together with or substituting the traditional methods already in use.

Some important examples of biometric techniques are iris recognition, face recognition, digital fingerprint analysis [1], speaker recognition, hand geometry recognition, signature analysis and others [2][3]. Every technique presents its particularities and application fields. The biometric authentication based on speech analysis, also known as automatic speaker recognition, has some advantages over other techniques: facility of speech recording using low cost hardware; the signal acquisition may be accomplished with no discomfort or user physical contact; and the system can be used without previous training. Although many recent advances and successes in speaker recognition have been achieved, such as in noise cancelling [4] and noise activity detection [5], there are still many problems for which good solutions remain to be found, like handling with vocal tract pathologies, the changes in the user emotional state, the control over recording environment, etc.

This work describes an application of speaker recognition techniques as the biometric measure for remote authentication of users in computer networks. First, we present the main features of a prototype that was developed to authenticate the user through speech over the Internet. After, the main results obtained from tests with several users are shown. Finally, conclusions and future work are discussed.

2 Proposed System

The proposed system is based on the client/server model. All user information and speech patterns (voiceprints) can be accessed from a database located on the server side. In addition, the server side is responsible of generating the voiceprints using the information extracted from the speech signal. The communication is reached through the Internet, using cryptographic techniques to guarantee data integrity. On the client side there is the user interface module, responsible for interacting with users and capturing users' speech samples using a regular sound board. This sound board needs to be properly installed in the client computer prior to using this system. Part of the speech digital processing is accomplished on the client side, which sends to the server side only the information (features) extracted from the speech signal. This choice was made to reduce the traffic through the network and to minimize the server processing load. Figure 1 illustrates the main building blocks of the developed prototype, which will be detailed in the following subsections.

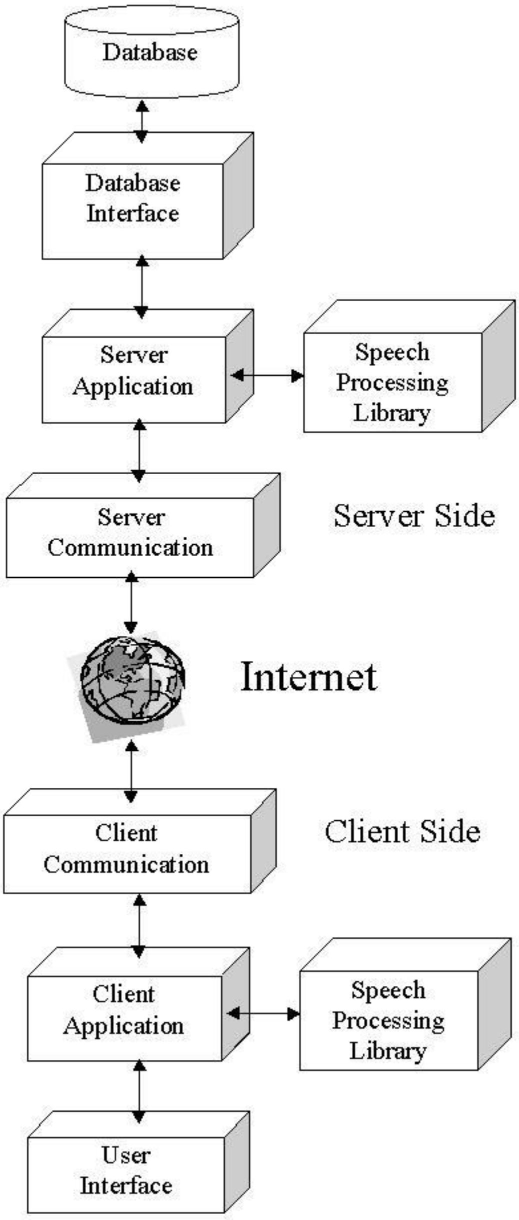

Fig. 1: Prototype building blocks

2.1 Software technology for prototype development

Two different programming languages were used in this prototype: Java and C++. The programming language used in the most parts of the system was Java. This programming language offers important resources for intercommunication and graphical interfaces. In addition, it supports different hardware architectures and operating systems with the use of Java Virtual Machines (JVMs). The integrated development environment (IDE) used was NetBeans 3.6, with Java 2 SDK standard edition version 1.4.1. Only the speech processing algorithms were implemented using C++ programming language, in Visual Studio 6.0 IDE. This speech processing library was connected to Java platform using Java Native Interface (JNI). The reasons of C++ usage instead of Java were the reduction of required processing time at both server and client sides and the possibility of immediate code reuse, developed in previous work [15]. The database used was MySQL, which was accessed through Java Database Connectivity (JDBC) [6].

2.2 Speaker Recognition Techniques

Speaker recognition, which can be classified into identification and verification, is the process of automatically recognizing who is speaking by analyzing the information included in speech waves. Speaker identification is the process of determining which registered speaker provides a given utterance. On the other hand, speaker verification, focused in this work, is the process of accepting or rejecting the identity claimed from a speaker [16].

A speaker verification system is basically composed by training and recognition phases. The training phase is accomplished first by the user. This stage includes acquisition of several utterances, extraction of representative features, speech pattern (voiceprint) generation and the identification of the similarity thresholds associated to the voiceprint. The recognition phase (also known as authentication) includes utterance acquisition, extraction of the same features from that utterance, and comparison of them with the voiceprint, generated in the early stage. At this moment, a pre-defined similarity threshold will indicate if the identity is accepted or rejected. Figure 2 shows the stages of training and recognition for a general speaker verification system.

In the proposed system, the algorithms for speech processing have been implemented in C++ programming language. The utterances are recorded at sample rate of 8000Hz, with 16 bits resolution in one channel (mono). After voice acquisition, the speech samples are pre-emphasized to cancel a spectral distortion introduced by the lips [7]. A first order finite impulse response (FIR) filter can give the desired spectral response of +6dB/octave. The FIR filter was implemented as:

$$y(n) = x(n) - 0{,}95x(n-1) \qquad 1 \le n < M \qquad (1)$$

where M is the number of samples in the speech signal x(n), and y(n) is the pre-emphasized signal.

After that, a 45ms hamming window is applied every 10ms of speech signal. The feature parameters are extracted from each pre-emphasized and windowed piece of speech signal. After that, a 30ms hamming window is applied every 10ms of speech signal. Silence frames are discarded based on energy estimation. The feature parameters used in this work are 16 mel-cepstral and 16 delta mel-cepstral coefficients, obtained from each pre-emphasized and windowed piece of speech signal. The techniques used in pre-emphasis and feature extraction phases are detailed in several works, as in [7][8]. A system based on Gaussian mixture models (GMMs) as the speakers' voiceprints, obtained from the adaptation of a universal background model (UBM), was used in the prototype. A total of 80 Gaussians are used in this work to represent a speaker's voiceprint. Further details related to the GMM-UBM classification process can be obtained in [9][10].

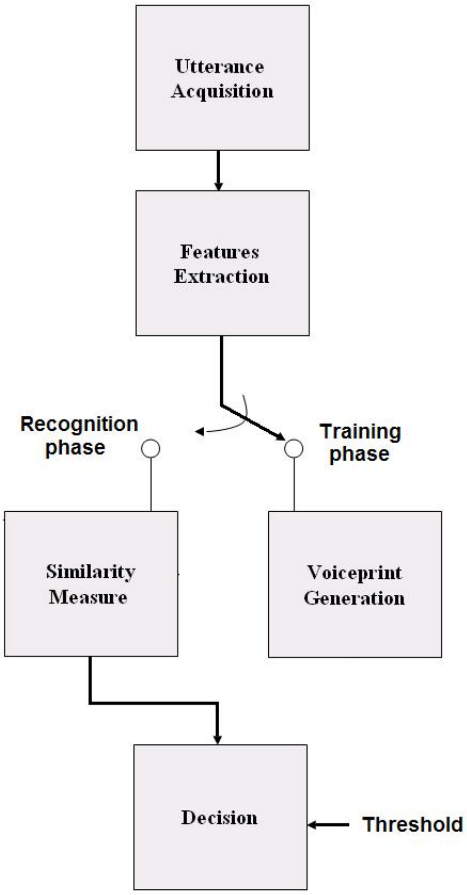

Fig. 2: Training and recognition phases for a generic speaker verification system

2.2.1 GMM-UBM system

GMM-based speaker recognition systems have been using extensively in recent years, showing good results. In order to improve speaker recognition accuracy under mismatched conditions, many techniques have been developed. One important technique is based on the use of a universal background model (UBM), where a GMM is obtained from the adaptation of a general speaker-independent GMM, trained using a large number of speakers. In the GMM-UBM speaker verification approach, a log-likelihood ratio can be estimated using the scores of the sequence of feature vectors $X=\{x_1,\ldots,x_T\}$ applied to both GMMs – the claimed speaker's GMM, denoted as λ_{spk}, and the UBM, denoted as λ_{UBM}:

$$\Lambda(X) = \log p(X / \lambda_{spk}) - \log p(X / \lambda_{UBM}) \quad (2)$$

where $p(X/\lambda_i)$ is the likelihood function of the sequence of feature vectors X in model i.

For Gaussian mixture speaker models, the log-likelihood of a model for X is usually computed as

$$\log p(X / \lambda) = \frac{1}{T}\sum_{t=1}^{T}\log\left(\sum_{k=1}^{N} w_k\, p_k(x_t)\right) \quad (3)$$

with mixture weights w_k and Gaussian densities $p_k(x)$. For a set of training vectors, the maximum likelihood parameters are estimated using the iterative Expectation-Maximization (EM) algorithm, well detailed in [10].

In a GMM-UBM speaker recognition system, instead of generating a GMM using directly the feature vectors extracted from the correspondent speaker's utterances, every registered speaker's GMM is derived from the UBM, using Bayesian adaptation and a relevance factor. A good description of this technique can be found in [10][11].

Some advantages can be outlined in the choice of GMM-UBM systems. First an intrinsic robustness to speech degradation can be observed, since the comparison based on both the claimed speaker's GMM and the UBM. Hence, if corrupted speech is introduced to the system, probably both scores will be reduced, but the log-likelihood ratio tends to be more stable. Also, the adoption of thresholds for speech acceptance or rejection becomes easier, since it does not vary so much. Another important issue concerns the immediate possibility of updating the speaker's GMM with time, by using low relevance factors that, during a long period, can maintain the speaker's voiceprints up-to-date with the speaker's speech. However, the voiceprint update was not implemented in this work

2.3 Database

It has been created a database for the voiceprint, features and user information storage. The database and database interface (shown in Figure 1) provide the server application with information needed to perform an user enrollment (on the training phase) or an authentication (on the recognition phase). Data referring to the system usage have been stored either, such as client IP address, connection information and users actions. A MySQL database has been accessed through Java Database Connectivity (JDBC) [6]. The JDBC employment allows easy integration with different databases, including the most popular ones. It allows the system to virtually change the database technology without concerning about changes on the rest of the code.

The database has been modeled with only two tables. The User table stores user-specific information: identification (CodID field), user name (UserName field), voiceprint (Model field), UBM used to generate the voiceprint (UBM field), date of voiceprint generation or last update (DateUpdate field), and all information concerning the speaker verification engine parameters (Tune field). This "tune" information refers to the recording process, feature parameters extraction, and classifier used in the authentication. The Historic table has the information about the events of the system usage: code of event (CodHist field), client IP address (IP field), features sent to server (Features field), and date of access (Date field). Historic table records are associated with Users table records through the Users table primary key CodID, considered as Historic table foreign key. Figure 3 illustrates the database modeling, indicating the tables used, fields presented on each table and the association between them.

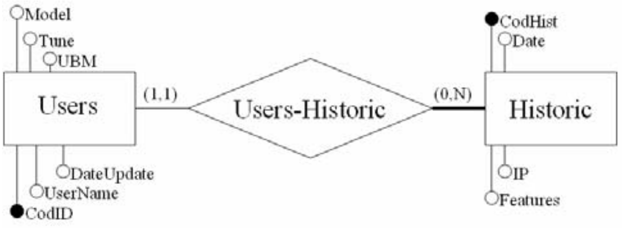

Fig. 3: The system's database model

2.4 Client-Server Communication

The communication between client side and server side was implemented using the Internet. The use of a distributed architecture like this allows the distribution of processing load between both client and server sides. However, it is important to note that this communication channel is widely known as insecure, if precautions are not taken. Since a safe communication is an important issue in any authentication system, the use of cryptography becomes essential.

2.4.1 Processing Load Distribution

When the client/server model is used, it is very important to define the system's distribution of processing load. In the case of speech processing, there are some alternatives. The speech signal could be sent to the server computer, which would be responsible for all processing. This alternative would increase the data traffic in the network, since the speech signal requires an excessive amount of bytes for storage or transmission. Considering the voiceprint generation process which uses twelve utterances composed by five digits, as suggested by [2] in a commercial product, then the amount of speech signal recorded will correspond to approximately one minute of speech. If the speech is recorded using a sample of 8000 Hz, 16 bits per sample, in only one channel, the amount of bytes will reach almost 1Mb. Even for the authentication phase, that uses less speech data than voiceprint generation (training phase), the amount of data is still considerable. For example, an authentication, composed of only 3 seconds of speech would require the transmission of 48Kb of data. On the other hand, if the feature parameters are extracted on the client side and are sent to the server, it is achieved a reduction of about 20% in the number of bytes that need to be transmitted. This calculation considered frames of 10 ms, the extraction of 16 mel-cepstral and 16 delta mel-cepstral coefficients from every frame, and 4 bytes for the storage of every floating number. Furthermore, if several simultaneous client requests are made, this reduction of transmission data will probably be significant to reduce the network overload.

On the other hand, since the proposed system processing is composed by well defined stages, as shown on figure 2, the overall processing load may be distributed. The proposed approach is to make the client computer acquire the utterance and extract the features parameters from the speech signal. Then, only the extracted features are sent to the server. This approach reduces dramatically the amount of bytes that need to be transmitted. Besides, the voiceprint generation and similarity measure remain on the server side, not compromising security policies. The communication between client side and server side uses cryptographic algorithms to guarantee privacy. In addition, the server processing load is reduced, since part of the processing is accomplished on client machine. This approach is used in the developed prototype.

2.4.2 Communication Cryptography

Cryptography (encryption and decryption) is composed by a set of techniques designed to protect information in situations where data has to be transmitted from a source to a destination part. In this system, after the client-server connection establishment, a trusted information traffic may be assured by employing cryptography techniques. The two components required to encrypt data are an algorithm and a key. The algorithm is generally known, and the key is kept secret. In a symmetric cryptosystem, the same key is used for encryption and decryption. In an asymmetric cryptosystem, the key (secret) used for decryption is different from the key (public) used for encryption.

In this work, the better choice for cryptographic communication would be the use of an asymmetric cryptosystem, such as Algorithm Encryption Standard (AES). However, it was not possible to verify the existence of a free-of-charge version of an asymmetric algorithm. So, it has been used Data Encryption Standard (DES) [12][13][14] symmetric key algorithm. Although this algorithm cannot be considered totally secure in the present days, this encryption engine has been implemented with Java 2 SDK version 1.4.1, and was used in this prototype. An important issue that has to be addressed is about the secret key establishment. The safe cryptographic key exchange between client and server machines is a difficult task, treated using Diffie-Hellman key exchange algorithm.

By defining a common "secret" on the client and server computers, used as DES cryptographic key, it has been avoided the key transmission throughout an unreliable communication path. It was done by using a pair of asymmetric keys, one public and another secret. Only the public keys have been delivered through the transmission path. This issue was implemented in a protocol and applied in this work, as shown in Figure 4. The prototype has been implemented it using Java classes for the cryptography, the CriptoClient and the CriptoServer. Some specific Java classes have been used for the application management and the *transport control protocol* (TCP) communication – Client and Server Classes.