International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 28611-28624

http://www.ripublication.com

Implementation of Network Security: Voice Based Authentication Using Kerberos Protocol

Ms. Apoorva P
Lecturer, Department of Computer Science, Amrita Vishwa Vidyapeetham, Mysuru Campus, Karnataka, India
apoorvaap7@gmail.com

Abstract

Authentication of a person is an important task in many areas of day-to-day life including electronic commerce, system security and access control. This work presents Kerberos a client\server authentication protocol which can perform a secure communication over unsecured environments (internet). A Kerberos protocol to solve the problem of authentication between client and server is proposed. Authentication using Kerberos requires a series of messages to be exchanged between user and the authentication server (the client and server). Tickets must be obtained from the authentication server and then exchanged between the client and server to perform authentication. If the authentication is not being done then there is compromise in security. The problem here is to authenticate using the information of the user without compromising in the security as well as the leakage of information of the individual. To overcome these problems, the proposed system uses voice authentication along with Kerberos Protocol. The voice data used for authentication is represented using a novel representation technique. It can successfully run over public network for remote access. It can also be implemented to take care of authentication between client and server.

Keywords: Kerberos, Ticket Granting Server, MFCC, Vector Quantization, Authentication.

Introduction

Network Security has become very significant in today's world, as a result of which various methods are adopted to find a way around it. Network Security is the most vital component in information security as it is responsible for securing all information passed through networked computers. Analyzing computer network security is to integrate resources related to computer network technology and security system to build a computer network security model. Computer network security is

fundamentally network information security. It refers to the network system that we use to preserve and flow information and data which may otherwise be exposed to accidental or deliberate damage, leaks or changes. Generally speaking, network security is inextricably related to the confidentiality integrity, authenticity and reliability of network. Its control technologies and concepts are necessary to analyze.

This work outlines a method for secure interaction between the client and the server with a strong authentication protocol Kerberos where it uses encrypted voice as the key.

Authentication is the major phase in network security. It is a concept to protect network and data transmission over wired as well as wireless networks. Authentication is one of the primary techniques of ensuring that the person who is transmitting the information is whom he says he is. It is thus the process of determining the actual identity of users, systems or any other entity in network. To verify someone's identity, password is mostly used. To authenticate user or machines, different techniques can be used to perform authentication between user and machine or machine and another machine too.

Related Work

Introduction to Kerberos Authentication

Kerberos is a network authentication protocol. It is designed to provide strong authentication for client/server applications by using secret-key cryptography. It has the following characteristics:

- It is secure: it never sends a password unless it is encrypted.
- Only a single login is required per session. Credentials defined at login are then passed between resources without the need for additional logins.
- The concept depends on a trusted third party – a Key Distribution Center (KDC). The KDC is aware of all systems in the network and is trusted by all of them.
- It performs mutual authentication, where a client proves its identity to a server and a server proves its identity to the client.

Kerberos introduces the concept of a Ticket-Granting Server (TGS). A client that wishes to use a service has to receive a ticket – a time-limited cryptographic message – giving it access to the server. Kerberos also requires an Authentication Server (AS) to verify clients. The two servers combined make up a KDC. Active Directory performs the functions of the KDC. The following figure shows the sequence of events required for a client to gain access to a service using Kerberos authentication. Each step is shown with the Kerberos message associated with it, as defined in RFC 4120 *"The Kerberos Network Authorization Service (V5)"*.

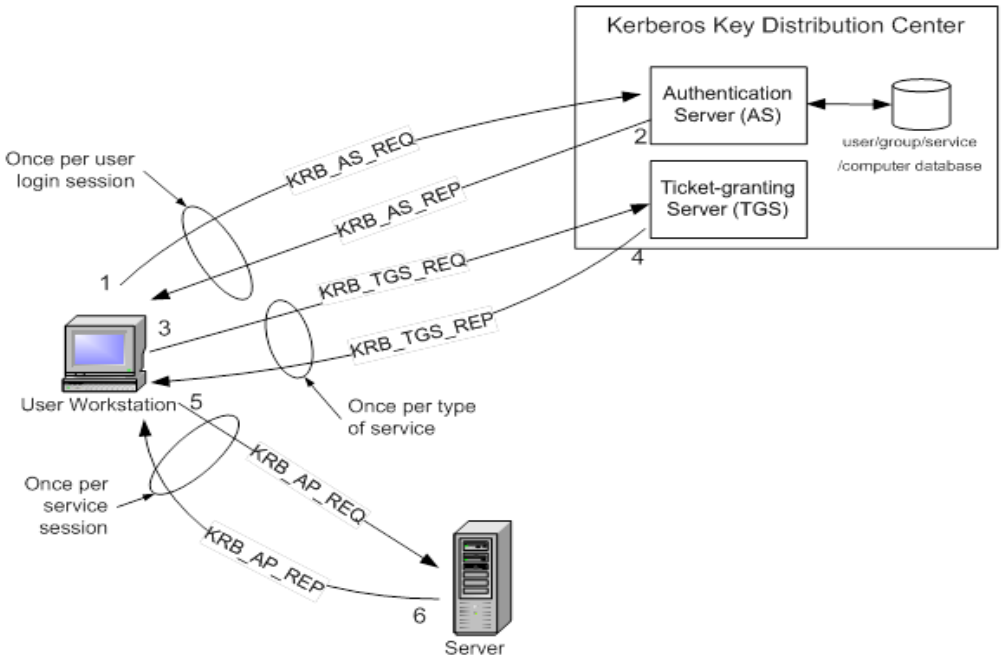


Figure 1: The Kerberos Network Authorization Service (V5)

Step 1: The user logs on to the workstation and requests service on the host. The workstation sends a message to the Authorization Server requesting a ticket granting ticket (TGT).

Step 2: The Authorization Server verifies the user's access rights in the user database and creates a TGT and session key. The Authorization Sever encrypts the results using a key derived from the user's password and sends a message back to the user workstation. The workstation prompts the user for a password and uses the password to decrypt the incoming message. When decryption succeeds, the user will be able to use the TGT to request a service ticket.

Step 3: When the user wants access to a service, the workstation client application sends a request to the Ticket Granting Service containing the client name, realm name and a timestamp. The user proves his identity by sending an authenticator encrypted with the session key received in Step 2.

Step 4: The TGS decrypts the ticket and authenticator, verifies the request, and creates a ticket for the requested server. The ticket contains the client name and optionally the client IP address. It also contains the realm name and ticket lifespan. The TGS returns the ticket to the user workstation. The returned message contains two copies of a server session key – one encrypted with the client password, and one encrypted by the service password.

Step 5: The client application now sends a service request to the server containing the ticket received in Step 4 and an authenticator. The service authenticates the request by decrypting the session key. The server verifies that the ticket and authenticator match, and then grants access to the service. This step as described does not include the authorization performed by the Intel AMT device, as described later.

Step 6: If mutual authentication is required, then the server will reply with a server authentication message.

The Kerberos server knows "secrets" (encrypted passwords) for all clients and servers under its control, or it is in contact with other secure servers that have this information. These "secrets" are used to encrypt all of the messages shown in the figure above. To prevent "replay attacks," Kerberos uses timestamps as part of its protocol definition. For timestamps to work properly, the clocks of the client and the server need to be in synch as much as possible. In other words, both computers need to be set to the same time and date. Since the clocks of two computers are often out of synch, administrators can establish a policy to establish the maximum acceptable difference to Kerberos between a client's clock and server's clock. If the difference between a client's clock and the server's clock is less than the maximum time difference specified in this policy, any timestamp used in a session between the two computers will be considered authentic. The maximum difference is usually set to five minutes.

Proposed Approach

Voice based cryptosystems join together cryptography and voice to promote from the strengths of both fields. In such systems, whereas cryptography provides high and adjustable security levels, voice brings in non repudiation and eliminates the must to remember passwords or to carry tokens etc. In voice cryptosystems, a cryptographic key is formed from the voice template of a user stored in the database in such a way that the key cannot be revealed without a successful voice authentication. The overall architecture of the voice based system to advance the network security is shown in figure 2. The Server preserves a database where the encrypted template of the user's voice is stored. In this arrangement, users communicate with the server for the principle of user authentication, by rendering users' voice, which is transformed into a long secret detained by the server in its database.

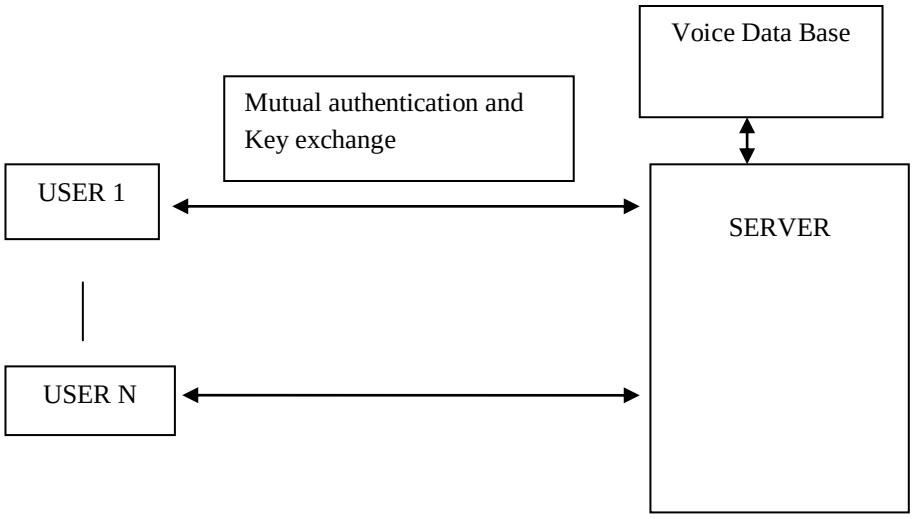


Figure 2: Voice Based System

Figure 3 shows a range of steps involved in the proposed system for network security using voice.

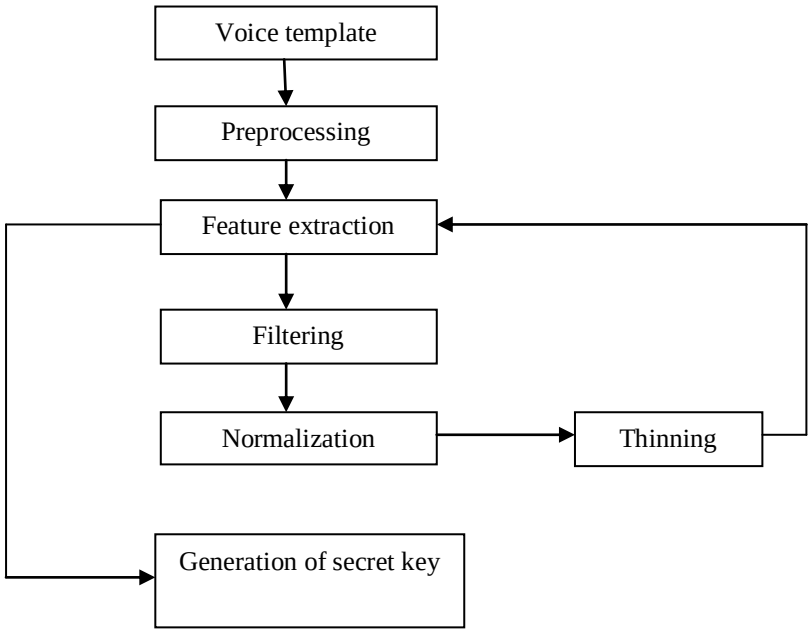


Figure 3: Steps Involved In Feature Extraction

A. User Registration

This step is popularly known as Enrolment phase. In all the security system to enroll as a rightful user in a service, a user must previously register with the service provider by ascertaining his/her identity with the provider. Therefore a mike is used to record the voice of the user to reveal his/her identity for the first time. The voice therefore obtained undergoes a series of enhancement steps. This is described in the subsequent section of this proposed paper. The analytical speech signal is first transformed into numerical data by digitizing the speech file. The MFCC features are extracted for each speech file.

B. Minutiae feature extraction and Generation of Secret Key

i) Mel frequency cepstral coefficients:

The Mel-scaled Cepstrum is a signal representation scheme used in the analysis of speech signals. Due to its reported superior performance, especially under adverse conditions, it is a popular choice as feature extraction front end to spoken language systems. It evolved over a period of more than fifty years. The Mel-scaled Cepstrum plays a major role in pattern recognition community in acoustical research. It is computationally efficient. In this section we clarify some of the issues regarding the Mel-scaled Cepstrum algorithm and its implementation as an approach to speech signal feature extraction.

Mel frequency Cepstral Coefficients are coefficients that represent audio based on human ear's' non-linear frequency characteristic perception. It is derived from the Fourier Transform of the audio clip. In this technique the frequency bands are positioned logarithmically, whereas in the Fourier Transform the frequency bands are not positioned logarithmically. As the frequency bands are positioned logarithmically in MFCC, it approximates the human system response more closely than any other system. These coefficients allow better processing of data. In the Mel Frequency Cepstral Coefficients the calculation of the Mel Cepstrum is same as the real Cepstrum except the Mel Cepstrum's frequency scale is warped to keep up a correspondence to the Mel scale.

The Mel scale was projected by Stevens, Volkmann and Newman in 1937. The Mel scale is mainly based on the study of observing the pitch or frequency perceived by the human ear. The scale is divided into the unit called mel. Figure 3.1 shows the example of normal frequency is mapped into the Mel frequency.

We know that human ears, for frequencies lower than 1 kHz, hears tones with a linear scale instead of logarithmic scale for the frequencies higher that 1 kHz. The mel-frequency scale is linear frequency spacing below 1000 Hz and a logarithmic spacing above 1000 Hz. The voice signals have most of their energy in the low frequencies. It is also very natural to use a mel spaced filter bank showing the above characteristics. For each tone with an actual frequency, *f*, measured in Hz, a subjective pitch is measured on a scale called the 'mel' scale. The pitch of a 1 kHz tone, 40 dB above the perceptual hearing threshold, is defined as 1000 mels.

$$mel(f) = 2595 * \log_{10}(1 + f/700)$$

The equation above shows the mapping the normal frequency into the Mel frequency.

ii) Calculating MFCC

The flow of calculating MFCC parameter is as follows:

Let the *N*-sample speech signal be

$$\mathbf{x} = x_0, \ldots, x_{N-1} \quad (1)$$

Step 1: Pre emphasis

The speech signal is pre emphasized to compensate for spectral tilt (i.e. $S'(w) = S(w).w^a$). This is a high-pass filtering operation and can be executed in either the time or frequency domain. The filter in the time-domain is of the form

$$x_i = x_i - a x_{i-1}, \quad 0.9 \le a \le 1.0 \quad (2)$$

Where the parameter *a* is not critical and is usually taken to be 0.95.

Step 2: Normalization

The maximum signal amplitude is normalized to one.

$$x_i = \frac{x_i}{Max_{j=0\ldots N-1}|x_j|} \quad (3)$$

Step 3: Blocking

The filtered, normalized signal is broken into *M* over-lapping frames and stored in an $M \times W$ matrix Y with its rows y_i representing the frames. *V* is the step size and *W* the frame size.

$$y_{ij} = x_{Vi+j},\ j = 0, \ldots, W-1,\ i = 0, \ldots, M-1 \quad (4)$$

Step 4: Windowing

Each frame is multiplied with a window function to minimize signal discontinuities in the time domain and the resulting spectral artifacts.

$$y_{ij} = y_{ij} w_j,\ j = 0, \ldots, W-1,\ i = 0, \ldots, M-1 \quad (5)$$

The hamming window is a popular choice.

$$w_j = 0.54 - 0.46 \cos\left(\frac{2\pi j}{W-1}\right),\ j = 0, \ldots W-1 \quad (6)$$

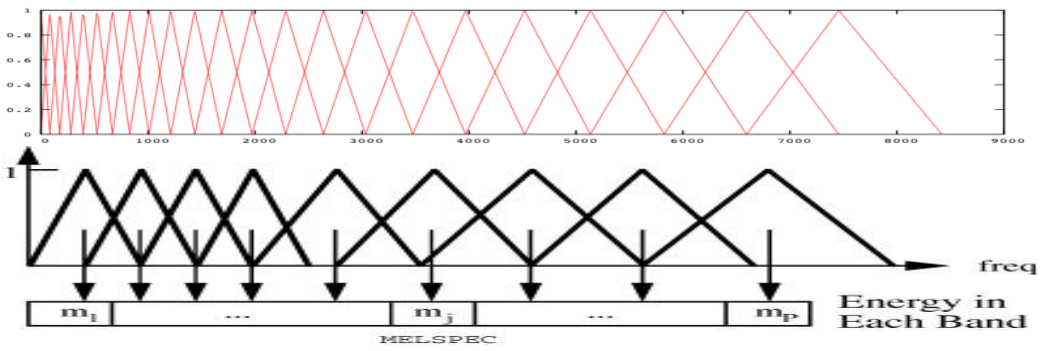
Step 5: Power Spectrum

The power spectrum of each window is calculated and represented by the $M \times U$ matrix S. *W* (and *U*) will be constrained by the FFT algorithm in a practical implementation. We used a prime factor FFT which gives more freedom in the choice of *W* than the standard radix-2 algorithms. Still, *W* needs to be one of a limited set of integers that will in general not be the same as the number determined by the choice of frame size. To work around this, y_i can be zero-padded or the frame size can be adjusted to coincide with a valid number.

$$s_i = |\mathrm{fft}(y_i)|^2,\ i = 0, \ldots, M-1 \quad (7)$$

Step 6: Mel Filter Bank

The triangular mel-filters in the filter bank are placed in the frequency axis so that each filter's center frequency follows the mel scale, in such a way that the filter bank mimics the critical band, which represents different perceptual effect at different frequency bands. Additionally, the edges are placed so that they coincide with the center frequencies in adjacent filters. Pictorially, the filter bank looks like:



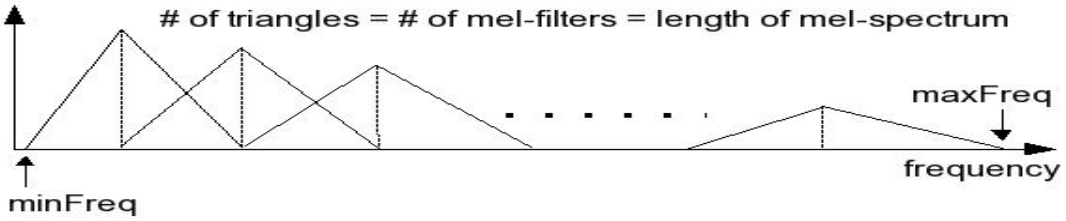


Figure 4: A Mel-Filter Bank

A common model for the relation between frequencies in mel and linear scales is as follows:

Mel Frequency = 2595 * log(1 + linear Frequency/700)

Step 7: Log Energy Filter Coefficients

To compensate for the increasing bandwidths of the filters, the energies are normalized by (10). This part of the processing is completed by taking the logarithm of each energy coefficient in a crude attempt to model the non-linear intensity-loudness relationship which is logarithmic in nature. These operations result in the $M \times K$ matrix P.

$$p_{ij} = \log_{10}\left(\frac{1}{A_j}\sum_{k=0}^{U-1} s_{ik} f_{jk}\right), \quad \begin{matrix} j = 0,....K-1 \\ i = 0,.....M-1 \end{matrix}$$

Where

$$A_j = \sum_{k=0}^{U-1} f_{jk} \qquad \textbf{(10)}$$

Step 8: Inverse Discrete Cosine Transform

The inverse cosine transform is used to orthogonalise the filter energy vectors. We can shorten the vector to L components, resulting in the $M \times L$ matrix Q

$$q_{ij} = \frac{1}{K}\sum_{k=0}^{K-1} p_{ik} \cos\left((k-0.5)\frac{\pi j}{L}\right) \quad \begin{matrix} j = 0,....L-1 \\ i = 0,.....M-1 \end{matrix} \qquad \textbf{(11)}$$

L is chosen to be less than K, usually somewhere between 9 and 15. This then constitutes a mel-scaled cepstrum feature vector.

Then by using vector quantization compression technique we reduce the dimension of the MFCC feature to code books of a fixed size ‘k’. Where k is the number of code vectors. The dimension is reduced from thousands of source vectors obtained in MFCC to code book of code vector size ‘k’. Each code vector represents the speech file.

iii) Vector quantization

Vector quantization (VQ) is a lossy data compression method based on the principle of block coding. It is a fixed-to-fixed length algorithm.

Design Problem

The VQ design problem can be stated as follows. Given a vector source with its statistical properties known, given a distortion measure, and given the number of code vectors, find a codebook (the set of all red stars) and a partition (the set of blue lines) which result in the smallest average distortion.

We assume that there is a *training sequence* consisting of ***M*** source vectors:

$$\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M\}.$$

This training sequence can be obtained from some large database. For example, if the source is a speech signal, then the training sequence can be obtained by recording several long telephone conversations. ***M*** is assumed to be sufficiently large so that all the statistical properties of the source are captured by the training sequence. We assume that the source vectors are **k**-dimensional, e.g.,

$$\mathbf{x}_m = (x_{m,1}, x_{m,2}, \ldots, x_{m,k}), \quad m = 1, 2, \ldots, M.$$

Let ***N*** be the number of code vectors and let

$$\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_N\},$$

Represents the codebook. Each code vector is **k**-dimensional, e.g.,

$$\mathbf{c}_n = (c_{n,1}, c_{n,2}, \ldots, c_{n,k}), \quad n = 1, 2, \ldots, N.$$

Let $\boldsymbol{S_n}$ be the encoding region associated with code vector $\boldsymbol{C_n}$ and let

$$\mathcal{P} = \{S_1, S_2, \ldots, S_N\},$$

Denote the partition of the space. If the source vector $\boldsymbol{X_m}$ is in the encoding region $\boldsymbol{S_n}$, then its approximation (denoted by $\mathbf{Q(X_m)}$) is $\boldsymbol{C_n}$:

$$Q(\mathbf{x}_m) = \mathbf{c}_n, \quad \text{if } \mathbf{x}_m \in S_n.$$

Assuming a **squared-error distortion measure**, the average distortion is given by:

$$D_{ave} = \frac{1}{Mk} \sum_{m=1}^{M} ||\mathbf{x}_m - Q(\mathbf{x}_m)||^2,$$

Where $||\mathbf{e}||^2 = e_1^2 + e_2^2 + \ldots + e_k^2$. The design problem can be succinctly stated as follows: Given ***T*** and ***N***, find ***C*** and ***P*** such that $\boldsymbol{D_{ave}}$ is minimized.

LBG Design Algorithm

The LBG VQ design algorithm is an iterative algorithm which alternatively solves the above two optimality criteria. The algorithm requires an initial code book. This initial

codebook is obtained by the *splitting* method. In this method, an initial code vector is set as the average of the entire training sequence. This code vector is then split into two. The iterative algorithm is run with these two vectors as the initial codebook. The final two code vectors are splitted into four and the process is repeated until the desired number of code vectors is obtained. The algorithm is summarized below.

LBG Design Algorithm

Step1: Given ***T***. Fixed ***ε>0*** to be a ``small" number.

Step2: Let ***N=1*** and

$$\mathbf{c}_1^* = \frac{1}{M}\sum_{m=1}^{M}\mathbf{x}_m.$$

Calculate

$$D_{ave}^* = \frac{1}{Mk}\sum_{m=1}^{M}||\mathbf{x}_m - \mathbf{c}_1^*||^2.$$

Step 3: Splitting: For *i=1,2…..N,* set

$$\begin{aligned}\mathbf{c}_i^{(0)} &= (1+\epsilon)\mathbf{c}_i^*,\\ \mathbf{c}_{N+i}^{(0)} &= (1-\epsilon)\mathbf{c}_i^*.\end{aligned}$$

Set ***N=2N***.

Step 4: Iteration: Let $D_{ave}^{(0)} = D_{ave}^*$. Set the iteration index ***i=0***.

For ***m=1,2…...M***, find the minimum value of

$$||\mathbf{x}_m - \mathbf{c}_n^{(i)}||^2,$$

Over all ***n=1,2…...N***. Let n^* be the index which achieves the minimum. Set

$$Q(\mathbf{x}_m) = \mathbf{c}_{n^*}^{(i)}.$$

For ***n=1,2…...N***, update the code vector

$$\mathbf{c}_n^{(i+1)} = \frac{\sum_{Q(\mathbf{x}_m)=\mathbf{c}_n^{(i)}}\mathbf{x}_m}{\sum_{Q(\mathbf{x}_m)=\mathbf{c}_n^{(i)}}1}$$

Set ***i=i+1***.

Calculate

$$D_{ave}^{(i)} = \frac{1}{Mk}\sum_{m=1}^{M}||\mathbf{x}_m - Q(\mathbf{x}_m)||^2.$$

If $(D_{ave}^{(i-1)} - D_{ave}^{(i)})/D_{ave}^{(i-1)} > \epsilon$, go back to Step (a).

Set $D_{ave}^{*} = D_{ave}^{(i)}$. For ***n=1,2…..N***, set

$$\mathbf{c}_n^* = \mathbf{c}_n^{(i)}$$

As the final code vectors.

Step 5: Repeat Steps 3 and 4 until the desired number of code vectors is obtained.

Data features are clustered to form a codebook for each speaker.

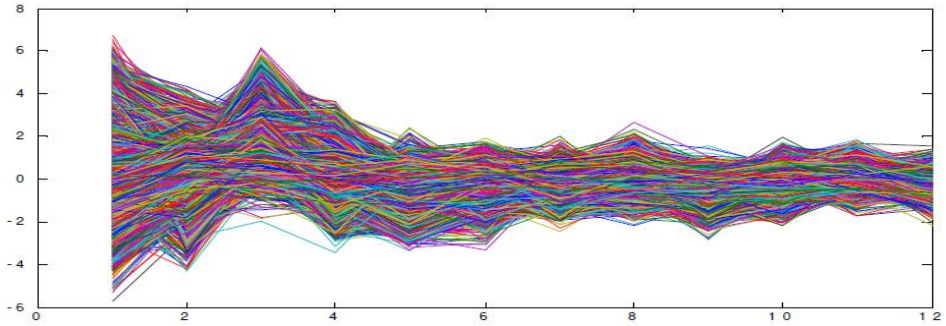


Figure 5: The vectors generated from voice before VQ

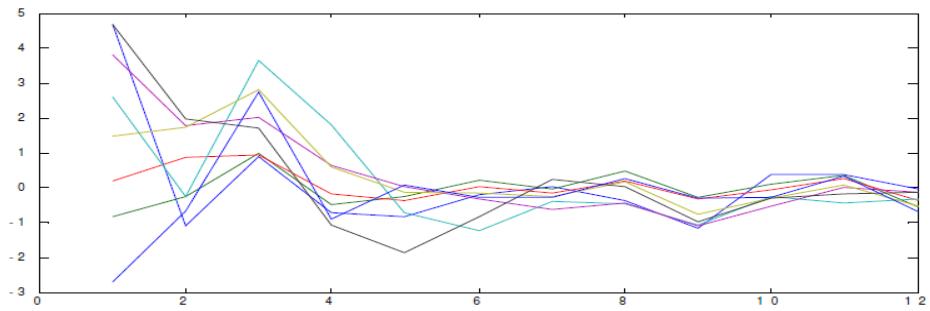


Figure 6: The representative feature vectors resulted after VQ

The obtained code vectors for each speech file are then used to form a symbolic representation. The speech file from each speaker is represented using the interval features.

Let [D_1, D_2, D_3……D_n] be a set of 'n' speech files. Let X_m={$x_{m,1}$,$x_{m,2}$,……$x_{m,k}$} be k-dimensional code vectors (vector quantized) characterizing the speech file D_n. We have computed the mean and standard deviation of the code vectors in each file. Then we add mean and standard deviation to obtain the maximum interval and we subtracted the mean and standard deviation to obtain the minimum interval. The obtained intervals of all speech files with respect to each category are combined to form a feature vector of length k. This process is repeated for all the speech files.

This is a vector of interval-valued features and this symbolic feature vector is stored in the knowledge base as a representative of the voice file. Similarly we compute symbolic feature vectors for all files and store them in the knowledge base.

Given a test speech, which is described by a set of 'm' feature values that is code vectors derived from the vector quantization compare it with the corresponding interval type feature values of the respective class that is stored in the knowledge base.

The above algorithm creates a database of voice. This generated key in the form of voice features is given to the user as a secret key. Kerberos uses this key as symmetric key. Based on the voice authentication a time stamp is given to the user. Kerberos caches this authentication tokens on the client side. Resource sever will compare the time stamp of the user against the local time. If the time skew between these two time stamps is too big, then the resource server will reject the authentication attempt.

Algorithmic module:

Step 1: The user logs on to the workstation and requests service on the host. The workstation sends a message to the Authorization Server requesting a ticket granting ticket (TGT).

Step 2: Input Voice file(s).

Step 3: Find MFCC feature for each file.

Step 4: Calculate code vectors for each train file using Vector Quantization.

Step 5: Calculate the Mean of code vectors representing each file.

Step 6: Calculate the Standard Deviation of code vectors representing each file.

Step 7: Calculate minimum interval for file by Subtracting standard deviation from mean.

Step 8: Calculate Maximum interval for file by Adding standard deviation to mean.

Step 9: Store these symbolically encrypted voice data in the Database.

Step 10: generate a secret key from the database for the user

Step 11: The Authorization Server verifies the user's access rights in the user database and creates a TGT and session key.

Step 12: The user proves his identity by sending an authenticator encrypted with the session key received in Step 11.

Step 13: The TGS decrypts the ticket and authenticator, verifies the request, and creates a ticket for the requested server.

Step 14: The client application now sends a service request to the server containing the ticket received in Step 13 and an authenticator.

Step 15: If mutual authentication is required, then the server will reply with a server authentication message.

Performance Measures

This section of the paper explains the performance measures of our approach. The voice processing has been done using MFCC in MATLAB. Password used for the transformation is 'FEATURES'. The performance measures obtained, exposed that the

proposed method effectively provides network security. Therefore it can be directly applied to strengthen existing standard security applications.

The minutia template supposed to be encrypted with symmetric cipher and is then transmitted to the server for storage in the database, so that it should not be possible for an outside attacker to determine the feature by an exhaustive search either at the server side or by meet in the middle attack.

Conclusion

Authentication is critical for the security of computer systems. Without knowledge of the identity of a principal requesting an operation, it is difficult to decide whether the operation should be allowed. Traditional authentication methods are not suitable for use in computer networks where attackers monitor network traffic to intercept passwords. The Kerberos authentication system is well suited for authentication of users in such environments.

This paper proposes an approach for network security by means of voice. The human voice can be effectively used to ensure the network security. In voice based cryptosystems, a cryptographic key is obtained from the voice template of a user stored in the database in such a way that the key cannot be revealed without a successful voice authentication. The primary advantage of the proposed approach is that we are able to achieve classification of strongly encrypted features. The proposed work is extremely secure under a variety of attacks.

References

[1] Atal B.S., 1974 *"Effective of linear prediction characteristics of speech wave for automatic speaker identification and verification",* J. Acoust. Soc. Am. 55 1304-1312.

[2] Campbell J. P. 1997 *"Speaker recognition: A tutorial. Proceedings of IEEE",* 85(9), 1437–1462.

[3] Clifford Neuman And Theodore Ts'o "Kerberos: An Authentication Service For Computer Networks ". – 2001.

[4] Fabrice Kah Giac-*"Understanding Kerberos V5 Authentication Protocol Security Essentials Certification (Gsec)"* - November 2003.

[5] Guru, D.S., Harish, B.S. and Manjunath .S.*"Symbolic representation of text documents."* In proceedings of third annual ACM Bangalore conference. 2010.

[6] J Kohl, C Newman –*"The Kerberos Network Authentication Service (V5)"* – 1993.

[7] Linde Y, Buzo A, & Gray R. M. 1980 *"An algorithm for vector quantizer design. IEEE Transactions on Communications"*, 28, 84–95.

[8] Roger M. Needham and Michael D. Schroeder, ''Using Encryption for Authentication in Large Networks of Computers,'' Communications of the ACM 21(12), pp. 993-999 (December, 1978).

[9] S. M. Bellovin and M. Merritt, *''Limitations of the Kerberos Authentication System''* Computer Communications Review 20(5), pp. 119-132 (October 1990).

[10] S. P. Miller, B. C. Neuman, J. I. Schiller, and J. H. Saltzer, Section E.2.1: Kerberos Authentication and Authorization System, M.I.T. Project Athena, Cambridge, Massachusetts (December 21,1987).