Finger Print Based Authentication and Key Exchange System Secure Against Dictionary Attack

Rajeswari Mukesh
Dept of Computer Science & Engg
Easwari Engineering College
Chennai- 600 089

A.Damodaram
Vice Principal
JNTU College of Engineering
Hyderabad-500 072

V.Subbiah Bharathi
Dean Academics
DMI College of engineering
Chennai-601 302

Summary
The Biometric based user authentication systems are highly secured and efficient to use and place total trust on the authentication server where biometric verification data are stored in a central database. Such systems are, prone to dictionary attacks initiated at the server side. Compromise of the authentication server by either outsiders or insiders do all user private data to exposure and may have serious repercussions to an organization. In this paper, we present a practical fingerprint based user authentication and key exchange system. In this system, the minutia extracted from the fingerprint is stored in the encrypted form in the server's database, to overcome the dictionary attacks mounted by the server. The image processing techniques are used to extract a biometric measurement from the fingerprint image. During login procedure the mutual authentication is done between the server and user and a symmetric key is generated on both sides, which could be used for further secure communication between them. Thus meet-in-the middle attack that happens between the user and the server can also be overcome. This system can be directly applied to strengthen existing password or biometric based systems without requiring additional computation.

Key words:
Finger Print, Authentication, Key Exchange, Dictionary Attack

1. INTRODUCTION

Accurate and automatic identification and authentication of users is a fundamental problem in network environments. Shared secrets such as Personal Identification Numbers or Passwords and key devices like Smart cards are not just enough in some cases. What is needed is something that could verify that you are physically the person you claim to be. The biometrics is enhancing our ability to identify people. And a biometrics system allows the identity of a living person based on a physiological characteristic or a behavioral trait to be verified or recognized automatically. Some of the biometrics used for authentication are Finger Print, Iris, palm print, Hand Signature stroke etc. In the Table I various biometric technologies have been compared based on various characteristics.

Among all the biometric techniques, today fingerprints are the most widely used biometric features for personal identification because of their high acceptability, Immutability and individuality. It is a well-known fact that fingerprint is unique to each & every person. These features make the use of fingerprints extremely effective in areas where the provision of a high degree of security is an issue.

TABLE I
COMPARISON OF VARIOUS BIOMETRIC TECHNOLOGIES

Biometric identifier	Un	Di	Pm	Co	Pf	Ac	Ci
Face	H	L	M	H	L	H	H
Finger print	M	H	H	M	H	M	M
Hand Geometry	M	M	M	H	M	M	M
Iris	H	H	H	M	H	L	L
Key stroke	L	L	L	M	L	M	M
Signature	L	L	L	M	L	H	H
Voice	M	L	L	M	L	H	H

Un- Universality Pf – Performance Di– Distinct
Ci– Circumvention Pm– Permanence L – Low
Co – Collectability M – Medium H- High

In this paper, hence we consider fingerprint for providing mutual authentication between the server and the user. We make use of Image processing technique to extract the biometric measurement called minutiae from the user's fingerprint. Further, a final post processing is performed to eliminate false minutiae. The minutiae point locations of a fingerprint is expressed in a Cartesian coordinate system as a 4-vector with its elements in order. Thus the user's full finger print image is converted and stored as encrypted binary template, which is used for authentication by the server. Thus the user's biometric verification data are first transformed into a Strong secret and is then stored in the server's database during registration. During log-in procedure authentication is done both at user's side and server side without transmitting the biometric measurement from the user to the server. Further the user and the server communicate with each other with a secret session key that is generated from the biometric for the rest of the transactions. This concept can also be applied to strengthen the existing single server password based authentication systems.

Manuscript received October 5, 2008
Manuscript revised October 20, 2008

1.1. *Related Work*

A lot of research has been carried out in the field of Authentication and Key Exchange protocols, which are based on passwords [1,2]. The Password based user authentication systems are low cost and easy to use but however, the use of passwords has intrinsic weaknesses. The user chosen passwords are inherently weak since most users choose short and easy to remember passwords. In particular, passwords are normally drawn from a relatively small dictionary; thereby prone to Brute-force dictionary attacks, where an attacker enumerates every possible password in the dictionary to determine the actual password.

These systems are essentially intended to defeat offline dictionary attacks by outside attackers and assume that the server is completely trusted in protecting the user password database. Once an authentication server is compromised, the attackers perform an offline dictionary attacks against the user passwords. To eliminate this single point of vulnerability inherent in the single-server systems, password systems based on multiple servers were proposed. The principle is distributing the password database well as the authentication function to multiple servers [3],so that an attacker is forced to compromise several servers to be successful in offline dictionary attacks. In such multi server password systems, either the servers are equally exposed to the users and or a user has to communicate in parallel with several or all servers for authentication. Recently, Brainard et al. [3] proposed a two-server password system in which one server exposes itself to users and the other is hidden from the users. Subsequently, Yang et al.[4] extended and tailored this two-server system to the context of federated enterprises, where the back-end server is managed by an enterprise headquarter and each affiliating organization operates a front-end server.

1.2. *Our Contribution*

Continuing this line of research on the authentication and key exchange paradigm, instead of traditional password based systems; we go for biometric information for mutual authentication and key generation. The biometric based key generation is unforgeable to a certain extent as biometric identities like finger print are unique to each and every individual. This system is a biometric-only system in the sense that it requires no users key cryptosystem and, thus, no Public Key Infrastructure (PKI). This makes the proposed system very attractive considering PKIs are proven notoriously expensive to deploy in real world. Moreover, the proposed system is particularly suitable for online web applications due to its efficiency in terms of both computation and communication.

2. OVERALL ARCHITECTURE

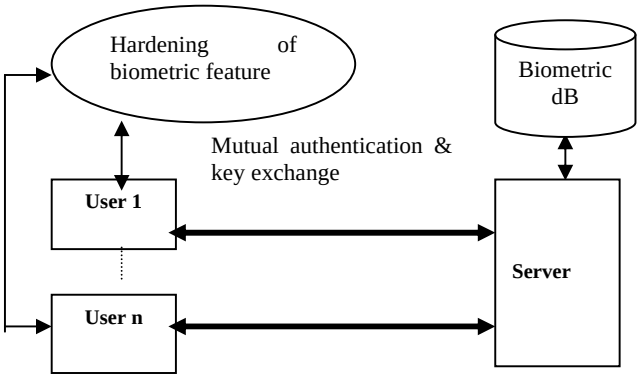


Fig.1 The Biometric and authentication architecture

The overall architecture of the biometric authentication And key exchange system is shown in the figure 1. The Server maintains a database where the encrypted minutia template of the user's finger print is stored. In this setting, users communicate with the server for the purpose of user authentication, by rendering his/her fingerprint, which is transformed into a long secret held by the server in its database.

3. THE BIOMETRIC AUTHENTICATION PROTOCOL

The main part of the protocol design is the defense against offline dictionary attacks by the servers when adversaries control them and also to overcome the meet-in-the-middle attack done between the user and the server. The intuition is to "harden" a user's fingerprint (FP) into a strong secret and store it in the database of the server during registration phase and authenticating the user using this hardened fingerprint during every login. Encrypted storage of the minutia template of the fingerprint is done in such a way that they are no longer subjected to offline dictionary attack During user login, the server using its encrypted fingerprint does user authentication. During authentication, an User using FP mutually authenticate each other and negotiate a secret session key. With the help of FP, the Secret Session Key, thus generated is used to perform encryption of any file to be downloaded.

3.1. User Registration

In any secure system, to enroll as a legitimate user in a service, a user must beforehand register with the service provider by establishing his/her identity with the provider. For this, the user provides his/her fingerprint through a finger scanner. The finger print image thus obtained undergoes a series of enhancement steps. This is followed by a Finger print hardening protocol with servers to obtain a hardened finger print FP which is stored into the server's database.

3.2. Fingerprint Enhancement

A fingerprint is made of a series of ridges and furrows on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows. Minutiae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending. A ridge termination is defined as the point where a ridge ends abruptly. A ridge bifurcation is defined as the point where a ridge forks or diverges into branch ridges (Fig.2.).

The quality of the ridge structures in a fingerprint image is an important characteristic, as the ridges carry the information of characteristic features required for minutiae extraction.

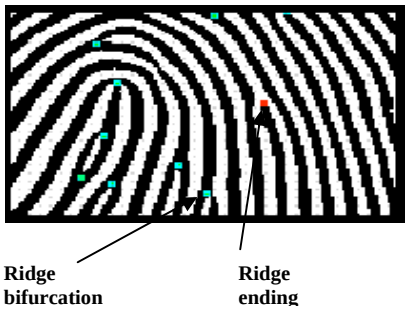


Fig.2. Example of Ridge bifurcation & Ridge ending

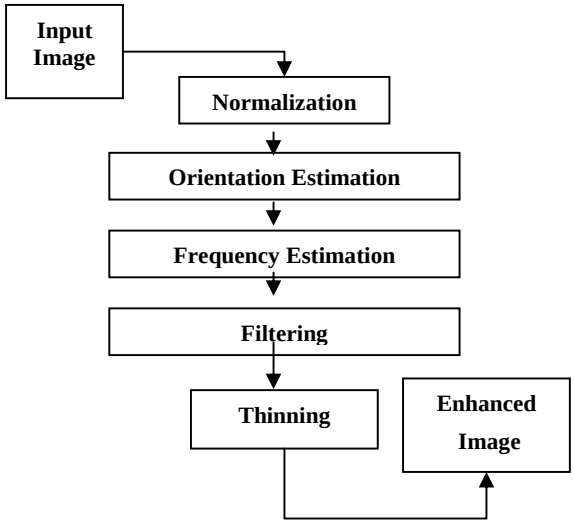


Fig 3. Block diagram for finger print enhancement

In practice, a fingerprint image may not always be well defined due to elements of noise that corrupt the clarity of the ridge structures. Thus, image enhancement techniques are often employed to reduce the noise and enhance the definition of ridges against valleys [6].

Figure 3 illustrates the different steps involved in the development of the Enhancement Finger print. The details of these steps are given in the following subsections.

3.2.1. Normalization

Normalization is used to standardize the intensity values in an image by adjusting the range of gray-level values so that it lies within a desired range of values. It does not change the ridge structures in a fingerprint; it is performed to standardize the dynamic levels of variation in gray-level values, which facilitates the processing of subsequent image enhancement stages. Fig. 4(a & b) shows the original fingerprint & the results of a normalized fingerprint.

(a) **(b)**

Fig. 4 (a) Original (b) Normalized image

3.2.2. Orientation Estimation

The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint (see Fig. 5). The orientation estimation is a fundamental step in the enhancement process as the subsequent Gabor filtering stage relies on the local orientation in order to effectively enhances the fingerprint image. Fig. 6(a & b) illustrates the results of orientation estimation & smoothed orientation estimation of the fingerprint image.

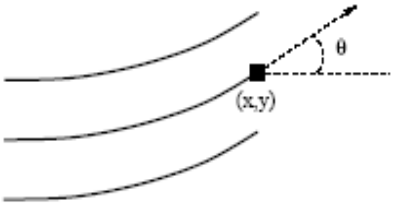
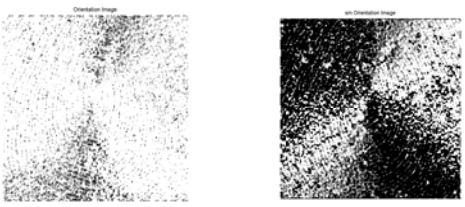


Fig. 5. The orientation of a ridge pixel in a fingerprint.



(a) **(b)**

Fig. 6. (a) Orientation Image (b) Smoothed orientation image

3.2.3. Local Frequency Estimation

In addition to the orientation image, another important parameter that is used in the construction of the Gabor filter is the local ridge frequency. The frequency image represents the local frequency of the ridges in a fingerprint. Fig. 7. shows the results of the local frequency estimation.

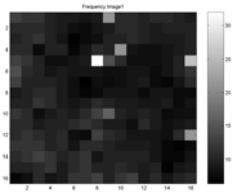


Fig.7. Frequency Image

3.2.4. Gabor Filtering

Once the ridge orientation and ridge frequency information has been determined, these parameters are used to construct the even-symmetric Gabor filter. Gabor filters are employed because they have frequency-selective and orientation-selective properties. These properties allow the filter to be tuned to give maximal response to ridges at a specific orientation and frequency in the fingerprint image. Therefore, a properly tuned Gabor filter can be used to effectively preserve the ridge structures while reducing noise. An even symmetric Gabor filter in the spatial domain is defined as,

$$G(x,y;\theta,f) = \exp\left\{\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\} \cos(2\pi f x_\theta),$$

$$x_\theta = x\cos\theta + y\sin\theta,$$

$$y_\theta = -x\sin\theta + y\cos\theta,$$

where θ is the orientation of the Gabor filter, f is the frequency of the cosine wave, σx and σy are the standard deviations of the Gaussian envelope along the x and y axes, respectively, and $x\theta$ and $y\theta$ define the x and y-*axes* of the filter coordinate frame, respectively. Fig.8 illustrates the results of using Gabor filter to a fingerprint image.

Fig. 8. Filtered Image

3.2.5. Thinning

The final image enhancement step typically performed prior to minutiae extraction is thinning. Thinning is a morphological operation that successively erodes away the foreground pixels until they are one pixel wide. The application of the thinning algorithm to a fingerprint image preserves the connectivity of the ridge structures while forming a skeleton version of the binary image. This skeleton image is then used in the subsequent extraction of minutiae. The process involving the extraction of minutiae from a skeleton image will be discussed in the next section. Fig. 9 illustrates the results of thinning to a fingerprint image.

Fig. 9 Thinned Image

4. FEATURE EXTRACTION

After a fingerprint image has been enhanced, the next step is to extract the minutiae from the enhanced image. Following the extraction of minutiae, a final image post processing stage is performed to eliminate false minutiae[7].

4.1. Minutiae Extraction

The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept [7]. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3x3 window. The CN value is then computed, which is defined as half the sum of the differences between pairs of adjacent pixels in the eight-neighborhood. Using the properties of the CN as shown in Table II, the ridge pixel can then be classified as a ridge ending, bifurcation or non-minutiae point. Fig 10 shows the list of minutiae in a fingerprint image.

TABLE 2

PROPERTIES OF CROSSING NUMBER

CN	Property
0	Isolated point
1	Ridge ending point
2	Continuing ridge point
3	Bifurcation point
4	Crossing point

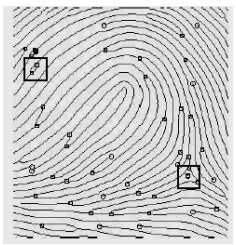


Fig. 10. Minutiae extraction on a fingerprint image.

4.2. Fingerprint Image Post Processing

False minutiae may be introduced into the image due to factors such as noisy images, and image artifacts created by the thinning process. Hence, after the minutiae are extracted, it is necessary to employ a post processing stage in order to validate the minutiae.Fig. 11 illustrates some examples of false minutiae structures, which include the spur, hole, triangle and spike structures. The subsequent steps of the algorithm depend on whether the candidate minutiae point is a ridge ending or a bifurcation.

Fig.11 Examples of typical false minutiae structures.

1. For a candidate Ridge ending point:

i) First, label with a value of 1 all the pixels in *M*, which are eight-connected with the ridge ending point

ii) The next step is to count in a clockwise direction, the number of 0 to 1 transitions (*T*01) along the border of image *M*. If *T*01 = 1, then the candidate minutiae point is validated as a true ridge ending.

2. For a candidate Bifurcation point:

i) Examine the eight neighboring pixels surrounding the bifurcation point in a clockwise direction. For the three pixels that are connected with the bifurcation point, label them with the values of 1, 2, and 3, respectively.

ii) The next label the rest of the ridge pixels that are connected to these three connected pixels. This labeling is similar to the ridge ending approach; however, instead of labeling a single ridge branch, three ridge branches are now labeled. Let *l* = 1, 2 and 3 represent the label for each ridge branch. For each *l*, label with *l* all the ridge pixels that have a label of 0, and are connected to an *l* labeled pixel.

iii) The last step is to count in a clockwise direction, the number of transitions from 0 to 1 (T_{01}), 0 to 2 (T_{02}), and 0 to 3 (T_{03}) along the border of image *M*.

If $T_{01} = 1 \wedge T_{02} = 1 \wedge T_{03} = 1$, then the candidate minutiae point is validated as a true bifurcation.

4.3. Mapping Function

The coordinate system used to express the minutiae point locations of a fingerprint is a Cartesian coordinate system. The *X* and *Y* coordinate of the minutiae points are in pixel units. Angles are expressed in standard mathematical format, with zero degrees to the right and angles increasing in the counter-clockwise direction. Thus, we can express a minutia as a 4-vector with its elements in order, the type *t*, the X and Y coordinates (*x, y*), and the direction θ (Angle value is a non-negative value between 0 and 179, in units of degree). And so, every minutia can be stored as a binary string $t|x|y|\theta$. While according to [7], each minutiae point can be recorded in 27 bits: 1 bit for the minutiae type, 9 bits each for minutia's *X* coordinate and *Y* coordinate, and 8 bits for the minutia angle.

Suppose $Mi = (ti, xi, yi, \theta i)$ (i = 1, . . . *n*) are the all extracted minutiae for a fingerprint image. Then we can arrange these minutiae points in a list from left to right by ascending *X*-coordinate, if equal by ascending *Y* -coordinate (first *X*, then *Y)* as follows:

$$Mi1Mi2 \cdots Min$$

Thus, we get a binary representation of minutiae. The result of the feature extraction stage is what is called a *minutia template (FP).* An approximate range on the number of minutiae found at this stage is from 10 to 80. If each minutia is stored with type (1 bit), location (9 bits each for *x* and *y*), and direction (8 bits), then each will require 27 bits and the template will require up to 270 bytes. Then this binary representation is mapped on to an finger print hardening protocol for the generation of Strong secret. .

The fingerprint processing has been done in MATLAB 7. Some of the minutia extracted from a sample finger print are listed in Table 3.

Table 3

List of Minutia

Type	x	y	Direction
1	37	120	2.96
1	52	85	2.97
1	250	139	0.06
0	12	131	0.26
0	21	61	2.84
0	23	137	0.31

5. THE FINGER PRINT HARDENING PROTOCOL

There are two requirements for registration using Finger Print.

1. The user should obtain the biometric feature from his finger print using appropriate image processing techniques as one mentioned in the previous section.

2. The minutia template should be encrypted with AES 128 bit 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 biometric

feature by an exhaustive search either at the server side or by meet in the middle attack.

The assumptions made in this design are the following

1. The finger print has been taken from the user for an optimal n number of times to get the correct set of minutia.

2. The key used for AES encryption has been already distributed to the user and the server by a TTP.

The following computations take place at the user side during registration process:

1. The user is asked to give the fingerprint input at least five times and the similar minutia are extracted to form minutiae template (FP).

2. The user then encrypts the minutia template using AES-128 bit symmetric cipher in ECB mode.

3. The user then sends ($U_{ID,}$ E_{AES}(FP)) to the server for storage in its database.

Thus the Implementation of Finger Print hardening protocol leads to the generation of Strong secret.

6. THE FINGER PRINT AUTHENTICATION PROTOCOL

The Algorithm makes the following Assumptions:-

1. Let p, q be two large prime numbers such that

$p = 2q + 1.$

2. Let $g \in QRp$ are of order q where QRp is the group of quadratic residues modulo p.

The outline of the fingerprint Authentication protocol is given below to enable mutual authentication and key exchange between the User and the Server.

Step 1: To initiate a request for service, user computes his FP1 = E_{AES}(FP).

Step 2: The user Computes $B_1 \equiv g^{FP1} (\mathrm{mod}\, p)$. The user sends the user ID along with B_1 to the server.

Step 3: Server then selects the encrypted minutia template with the user-Id using a table look-up procedure and computes $B_2 \equiv g^{FP2} (\mathrm{mod}\, p)$, where FP2 is the encrypted minutiae template stored at the server side during registration. Then the server compares whether $B_1 \equiv B_2$ (mod p). If it holds the server is assured of the authenticity of the user otherwise aborts the authentication protocol. Then the server sends B_2 to the user.

Step 4: Upon reception of B_2, User verifies whether $B_1 \equiv B_2$ (mod p). If so authenticated otherwise aborts the authentication protocol. If authenticated the user computes the session key by using the formula

$$Ks = H_{SHA1} (U_{ID}, FP1) ;$$

Step 5: Simultaneously the server also generates the session key using the formula

$$Ks = H_{SHA1} (U_{ID}, FP2) ;$$

7. STRENGTH OF THE PROTOCOL

The analysis for the security of the protocol is based on the following Deffie-Hellman (DDH) assumptions [8]:

Assumption 1: For a cyclic group G, generated by g, we are given g and g^n, $n \in N$, the challenge is to compute n.

Assumption 2: Given g, g^a, g^b, it is hard to compute g^{ab}

Clearly if these assumptions are not satisfied then C, an adversary, can gain access to the key gab. The relationship between these two assumptions has been extensively studied. It is clear that assumption 2 will not be satisfied in a group where finding a discrete logarithm solution is easy In Maurer and Wolf(1999), Boneh and Lipton (1996), the authors show that in several settings the validity of assumption 2 and the hardness of the discrete logarithm problem are in fact equivalent.

8. CONCLUSION

This Biometric Authentication and key exchange system together with its practical applications offers many appealing performance features. The salient features of this proposal make it a suitable candidate for number of practical applications like Biometric ATMs and in future, Biometric online web applications etc. Compared with previous solutions, our system possesses many advantages, such as the secure against dictionary attack, avoidance of PKI, and high efficiency in terms of both computation and communications. In this system, we have reused ideas in the areas of image processing technique to extract the minutiae from biometric image. Therefore it can be directly applied to fortify existing standard single-server biometric based security applications.

REFERENCES

[1] W. Ford and B S. Kaliski Jr., “Server-Assisted Generation of a Strong Secret from a Password,” Proc. IEEE Ninth Int’l Workshop Enabling Technologies, 2000.

[2] M.Bellare, D. Pointcheval, and P. Rogaway, “Authenticated Key Exchange Secure Against Dictionary Attacks,” Advances in Cryptology Eurocrypt ’00, 2000 pp. 139-155.

[3] J. Brainard, A. Juels, B. Kaliski, and M. Szydlo, “A New Two - Server Approach for Authentication with Short Secrets,” Proc. USENIX Security Symp., 2003.

[4] Y.J. Yang, F. Bao, and R.H. Deng, "A New Architecture for Authentication and Key Exchange Using Password for Federated Enterprises, " Proc. 20th Int'l Federation for Information Processing Int'l Information Security Conf. (SEC '05), 2005

[5] Y.J. Yang, F. Bao, and R.H. Deng "A Practical Password – Based Two Server Authentication and Key Exchange System", IEEE Transactions on Dependable and Secure Computing , Vol 3,No . 2, April-June 2006

[6] L. Hong, Y. Wan, and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," IEEE *Trans. Pattern Analysis and Machine* Intelligence, vol. 20, no.8, 1998, pp.777-789.

[7] S. Kasaei, M. D., and Boashash, B. Fingerprint feature extraction using block-direction on reconstructed images. In *IEEE region TEN Conf., digital signal Processing applications, TENCON* (December 1997), pp. 303– 306.

[8] D. Boneh,"The Decision Diffie-Hellman Problem," Proc. Third Int'l Algorithmic Number Theory Symp., 1998 pp. 48-63,.

About The Authors

Rajeswari Mukesh is working as Assistant Professor in the Department of Computer Science and Engineering at Easwari Engineering College, Chennai, India. She has done her B.E and M.E in Computer Science and Engineering and currently pursuing Ph.D at JNTU Hyderabad. Her area of interests include Network Security and Image Processing.

Dr. A. Damodaram is Vice Principal and Professor of Computer Science and Engineering, JNTU College of Engineering, Hyderabad, India. His research interests include Software Engineering, Computer Networks and Image Processing. Prof. Damodaram was awarded his Ph.D. in Computer Science and Engineering from JNTU. He has a rich experience of 17 years in Teaching, Research and mentoring research scholars in his respective areas. He is Member of Academic Council in Cochin University of Science and Technology, Cochin. He is a member of AIEEE, New Delhi and Governing Council, JNTU College of Engineering, Hyderabad.

Dr.V.Subbiah Bharathi is working as Dean Academics at DMI Engineering College, Chennai, India. He has received Ph.D from Manonmaniam Sundaranar University. He has got 25 national and international papers published in reputed journals including ACM. His area of research include Image processing and Network Security .