International Journal of Computer Science
and Engineering (IJCSE)
ISSN(P): 2278-9960; ISSN(E): 2278-9979
Vol. 4, Issue 5, Aug - Sep 2015, 17-26
© IASET

International Academy of Science,
Engineering and Technology
IASET Connecting Researchers; Nurturing Innovations

ENCRYPTION KEY GENERATION BY USING MODIFIED HAND-GEOMETRY BASED CRYPTOSYSTEM TO SECURE SMS IN ANDROID

SREYA BHAR

Department of Computer Science and Engineering, Guru Nanak Institute of Technology, Tamil Nadu, India

ABSTRACT

A biometric is defined as a unique, measurable, biological characteristic or trait for automatically recognizing or verifying the identity of a human being. Statistically analyzing these biological characteristics has become known as the science of biometrics. The very basis of this Biometric Cryptosystem lies on the very fact that some features of human body are significantly unique to each and every human in the world, such as fingerprint, DNA sequence, Iris, etc. Using those biometric we can generate an exclusive key that will be unique for each and every individual. Now having generated these keys we can use them for encrypting our message. And as because these keys are uniquely generated for individual persons there's no chance of there will be a matching keys. Moreover as I use RSA algorithm based encryption technique so the encryption lies on two basic sets of keys to decrypt the message. Hand geometry is a kind of biometric measure where Data is read and processed independently of the position of the user hand. This is done by analyzing the curvature profile of the hand contour, making the feature extraction process rotation and translation invariant. In my proposed work first I identify keys from hand geometry and after that by using those keys I can encrypt an SMS in mobile android by using RSA algorithm.

KEYWORDS: Android, Biometric Cryptosystem, Feature Extraction, Hand Geometry, RSA

INTRODUCTION

Hand geometry is a biometric that identifies users by the shape of their hands. Hand geometry readers measure a user's hand along many dimensions and compare those measurements to measurements stored in a file. In this work the fingers were identified and segmented by analyzing the closed shape formed by the contour of the hand, looking for a sequence of maxima of curvature along the way Our main goal in this project was to be able to acquire the images free from any restriction by allowing the user to put his hand in virtually any position inside the scanning area of the input device.

MODULES OF A BIOMETRIC SYSTEM

A biometric system comprises of three important modules. Preprocessing, Feature Extraction and Matching. When the input data is fed into the biometric system it may be unsuitable for feature extraction. This is due to the several noise elements which may creep into the data. Noise may be the result of the atmospheric conditions or the surroundings. It may also be introduced by the equipment used for collecting the data. Also the users may introduce some noise inadvertently. The primary job of the preprocessing module is to clean up the noise introduced o that the features can be extracted correctly. The proposed system first runs the input image through a noise removal algorithm for this purpose.

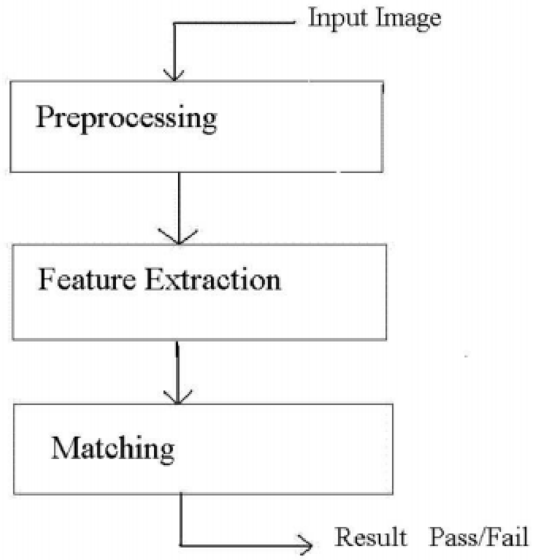


Figure 1: Basic Biometric System

MY PROPOSED APPROACH

In my approach I can generate two keys (private and public) after modifying pre existing hand geometrical approach and by using those key I can encrypt a SMS in android by using RSA algorithm. To discuss it I can segregate those parts onto different proposed methodology.

METHODOLOGY-I- MODIFIED HAND GEOMETRY BASED CRYPTOSYSTEM

Step 1: Noise removal

Step 2: Edge detection

Step 3: Features Extraction

Step 3a: Finger length detection by using convolution method

Step 3b: Finger width detection by using convolution method

Step 4: Determine the perimeter of Palm

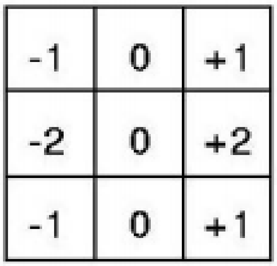
Step 5: Matching the Palm pattern

NOISE REMOVAL

Ideally the scanned input image should contain no noise. However due to dust and dirt both on the palm and on the scanner bed, even in minute quantities may produce differences between the actual image scanned and the palm print. The noise removal algorithm in this case utilizes the fact that the required handprint is present in only a portion of the total image i.e in the lower central side of the image. This can be done using the binary search algorithm. The algorithm starts from the center of the image and works its way upwards. Since binary search is used the number of rows searched is very low. When such a row is determined all the points above that row are set as black. This eliminates all the noise above this row. Using the binary search algorithm again a column is identified left of which is not required for feature extraction. This is any column between the left margin and the center of the image which has no lit pixels. All the pixels left of this column are set to black. Similarly a column right of which no features are to be extracted is determined. All the pixels right to this column are set to black. This reduces the noise present in the image considerably without affecting the actual palm print.

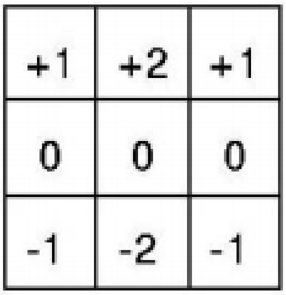
EDGE DETECTION

According to Canny's edge diction algorithm we can calculate the gradient along X axis as well as Y axis.



-1	0	+1
-2	0	+2
-1	0	+1

Figure 2: Calculate the Gradient along X Axis



+1	+2	+1
0	0	0
-1	-2	-1

Figure 3: Calculate the Gradient along Y Axis

Algorithm:

Step 1: Determine grad x and grad y, the values returned by the kernels.

Step 2: Determine the angle of the edge theta = tan-1 (grad x/grad y).

Step 3: Approximate theta to one of these values 45 90, 135 and 0 or 180.

Step 4: Traverse along the edge in the direction of the approximated theta and set to 0 any pixel which is not along theta

FINGER LENGTH DETECTION BY USING CONVOLUTION METHOD

The algorithm to determine the tip of the other fingers starts by utilizing the previous tip found. Since the little finger is the first the algorithm starts from (0, 0), the top left corner of the image. It then finds the first lit pixel traversing column wise. This lit pixel is somewhere along the left boundary of the palm. Now the algorithm has to traverse along lit pixels so as to reach the tip of the finger. During this traversal the value of the y co-ordinate is constantly decreasing while value of the x co-ordinate may increase or decrease. The algorithm halts when the value of the y co-ordinate can no longer decrease. It n assigns the point as the tip of the little finger (x1, y1). The system now finds the bottom of the finger (x2, y2) by using a line detection algorithm.

Step 1: Determine the tip of the finger (x1, y1) as described earlier.

Step 2: Apply the line detection kernels at the current pixel and obtain the responses.

Step 3: Pick the kernel with the highest response.

Step 4: Move to the pixel indicated by the kernel with the highest response subject to the condition that the value of y should always increase as the length is being measured from top to bottom.

Step 5: If no kernel shows a positive response for which the next y value will increase then mark the point (x2, Y2) and move to Step 7.

Step 6: Repeat Step 2 to Step 5.

Step 7: Obtain the length of the line as the distance between points (x1, y1) and (x2, y2).

FINGER WIDTH DETECTION BY USING CONVOLUTION METHOD

From the algorithm for determination of finger length the values of the tip of the finger (x1, y1) and the bottom (x2, y2) are obtained. By dropping perpendicular lines from these points to the line, obtained using the least square method the starting and ending points of the line are determined. With the starting and ending points of the line known the line can be divided into 3 equal parts generating two more points (x3, y3) and (x4, y4). A line perpendicular to the interpolated line and starting at (x3, y3) is drawn toward s the other boundary of the finger. The width is considered to be the distance between the starting point (x3, y3) and the point where the perpendicular meets the other boundary of the finger. To determine the point where the perpendicular meets the other boundary of the finger the algorithm traverses along a line parallel to the X axis using the point (x3, y3) as the starting point. The algorithm traverses till it encounters a lit pixel. This pixel is on the other boundary of the finger. The algorithm then attempts to find a pixel closest to the perpendicular line by considering all lit pixels within a certain distance from the current pixel.

Step 1: Draw a line perpendicular to the interpolated line starting at (x3),(y3).

Step 2: Traverse to the right parallel to the x axis until a lit pixel is encountered.

Step 3: Take a 5X5 pixels section of the image with the pixel encountered in step 2 as the center.

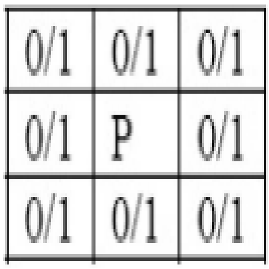
Step 4: For each lit pixel calculate the distance from the perpendicular line.

Step 5: The pixel for which the distance to the line is minimum is the new center for the 5X5 section.

Step 6: Repeat Steps 4 and 5 till the minimum stops changing.

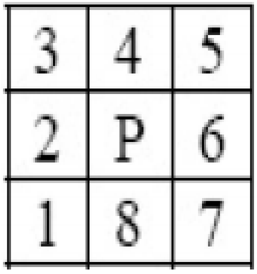
Step 7: Obtain the finger width which is the distance between the pixel at the center and (x3, y3).

DETERMINING THE PERIMETER OF PALM DIAMETER



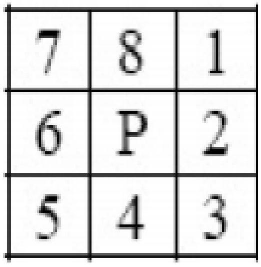
0/1	0/1	0/1
0/1	P	0/1
0/1	0/1	0/1

Figure 4: Probability Matrix



3	4	5
2	P	6
1	8	7

Figure 5: Priority Matrix



7	8	1
6	P	2
5	4	3

Figure 6: Mirror of the Priority Matrix

Step 1: Starting from left bottom find the lower boundary of the left side of the palm(l).

Step 2: Use the priority matrix to determine the next pixel.

Step 3: Save the value of the priority matrix used as value.

Step 4: Set the value of current pixel to 0 move to the next pixel chosen by the matrix.

Step 5: If value >= 6 then use mirror of the priority matrix.

Step 6: Repeat Step 3 to Step 5 till rightmost end is reached.

MATCHING THE PALM PATTERN

The features obtained from the input image are matched against the images in the database. Even under the best of conditions it cannot be expected that the features obtained match exactly with the features of the previous image of the same individual. The extracted features are in the form of positive integers. These are referred to as magnitude of the features. So to obtain a match the sum of the difference between the features obtained from the input image and the image in the database is calculated.

diff = magnitude of Database image – magnitude of input image

sum = magnitude of Database image + magnitude of input image

Using the value of sum the difference between the features can be viewed in proper context. So the actual match of the feature is:

match = sum/diff

METHODOLOGY-II- RSA ALGORITHM FOR ENCRYPTION OF SMS IN ANDROID

Key Generation Algorithm

Step 1: Generate two large random primes, p and q.

Step 2: Compute n = p*q and z = (p-1)(q-1).

Step 3: Choose a number relatively prime to z and call it d.

Step 4: Find e such that e*d=1 mod z.

Step 5: The public key is (n, e) and the private key is (n, d).

Encryption

Sender A does the following:

Step 1: Obtains the recipient B's public key (n, e).

Step 2: Represents the plaintext message as a positive integer m.

Step 3: Computes the ciphertext c =m^e mod n.

Step 4: Sends the ciphertext c to B.

Decryption

Recipient B does the following:-

Step 1: Uses his private key (n, d) to compute m = c^d mod n.

Step 2: Extracts the plaintext from the integer representative m.

METHODOLOGY-III- PROPOSED SMS ENCRYPTION IN ANDROID BY USING RSA

SMS is a communication service standardized in the GSM mobile communication systems. It can be sent and received simultaneously with GSM voice, data and fax calls. This is possible because whereas voice, data and fax calls take over a dedicated radio channel for the duration of the call, short messages travel over and above the radio channel using the signaling path. SMS contains some meta-data:

- Information about the senders (Service center number, sender number).
- Protocol information (Protocol identifier, Data coding scheme).
- Timestamp.

WORKING PROCEDURE OF SMS IN ANDROID

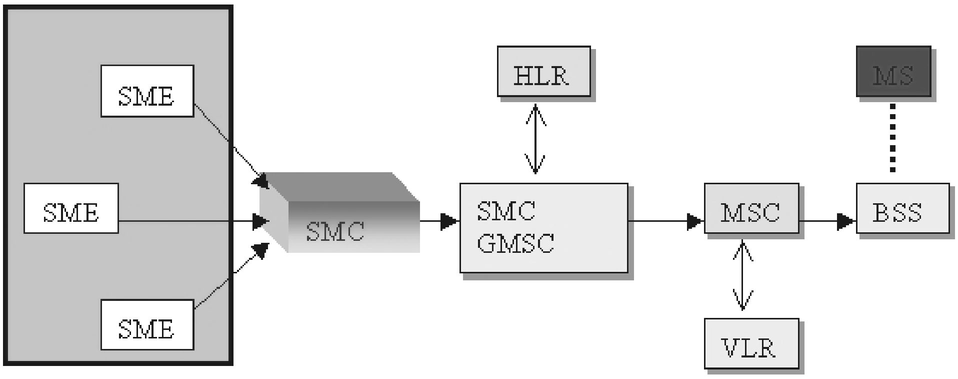


Figure 7: SMS Transfer

SMC (Short Message Center) is the entity which does the job of store and forward of messages to and from the mobile station. The SME (Short Message Entity) which can be located in the fixed network or a mobile station, receives and sends short messages.

SMS GWMS (SMS gateway MSC) is a gateway MSC that can also receive short messages. The gateway MSC is a mobile network's point of contact with other networks. On receiving the short message from the short message center, GMSC uses the SS7 network to interrogate the current position of the mobile station form the HLR, the home location register.

HLR is the main database in a mobile network. It holds information of the subscription profile of the mobile and also about the routing information for the subscriber, i.e. the area (covered by a MSC) where the mobile is currently situated. The GMSC is thus able to pass on the message to the correct MSC.

MSC (Mobile Switching Center) is the entity in a GSM network which does the job of switching connections between mobile stations or between mobile stations and the fixed network.

A VLR (Visitor Location Register) corresponds to each MSC and contains temporary information about the mobile, information like mobile identification and the cell (or a group of cells) where the mobile is currently situated. Using information from the VLR the MSC is able to switch the information (short message) to the corresponding BSS (Base Station System, BSC + BTSs), which transmits the short message to the mobile. The BSS consists of transceivers, which send and receive information over the air interface, to and from the mobile station. This information is passed over the signaling channels so the mobile can receive messages even if a voice or data call is going on.

SECURED SMS IN ANDROID SECURITY MODEL

The Android operating system's goal is to protect user data, protect system resources, and provide application isolation. To achieve these goals the following security features are provided [Security Overview]:

- Robust security at the OS level through the Linux kernel
- Mandatory application sandbox for all applications
- Secure inter process communication

- Application signing
- Application-defined and user-granted permissions

Following Figure shows the different components and considerations of the Android software stack. Each part of the stack operates under the assumption that everything below it is properly secured. The core of the Android security model is the Linux kernel. Linux itself has been around for a very long time and is a very robust kernel now after being constantly improved. It is used in the industry and trusted by many professionals. This kernel provides the Android OS with a user-based permissions model, process isolation, a mechanism for secure IPC, and the ability to remove parts of the kernel.

SMS ENCRYPTION-DECRYPTION IN ANDROID

- The sender communicates with the receiver through any confidential discussion.
- When both the sender and receiver are ready with their application, the sender types in the recipient’s number and the body of the message and clicks Send.
- On this command, the RSA algorithm is triggered at the sender side and the keys are generated. The sender’s Public Key is then sent to the receiver.
- The receiver acknowledges this by clicking the Read button where the received key is read by the receiver application and the RSA algorithm is triggered at receiver’s end.
- The receiver’s Secret Key is generated and its Public Key is sent to the sender.
- On receiving the Public Key from the receiver, sender’s Secret Key is generated at the sender side and the message is encrypted using RSA algorithm and sent to the receiver.
- The receiver receives the message and decrypts it using his Secret Key with the RSA algorithm to obtain the original message.

CONCLUSIONS

In this paper i can generate two keys (private and public) after modifying pre existing hand geometrical approach and by using those key I can encrypt a SMS in android by using RSA algorithm. I can make a little bit modification in hand geometry key extraction and other features. In RSA algorithm instead of two prime no I can generate two keys and using those keys I can encrypt and activate SMS services on android system. Through this approach I can provide the protection of user data, system resources, and also provide application isolation.

REFERENCES

1. Xin Zhou, Xiaofei Tang, “Research and Implementation of RSA Algorithm for Encryption and Decryption”, IEEE, 6th International Forum on Strategic Technology, pp- 1118 – 1121
2. Sonal Sharma, Saroj Hiranwal, Prashant Sharma,"A NEW VARIANT OF SUBSET-SUM CRYPTOSYSTEM OVER RSA", International Journal of Advances in Engineering & Technology, Jan 2012.ISSN: 2231-1963

3. Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, and September 2000.

4. Rashmi Ramesh Chavan, Manoj Sabnees,‖ Secured Mobile Messaging‖ 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET]

5. David Lisoněk, Martin Drahanský ―SMS Encryption for Mobile Communication‖ 2008 International Conference on Security Technology

6. Tarek M. Mahmoud, Bahgat A. Abdel-latef, Awny A. Ahmed, Ahmed M. Mahfouz, ―Hybrid Compression Encryption Technique for Securing SMS‖, International Journal of Computer Science and Security (IJCSS), Volume (3): Issue (6)

7. H. Marko, H. Konstantin, "Strong Mobile Authentication",Proceedings of 2nd International Symposium on WirelessCommunication Systems, Sept 5-7 2005, pp.96-100.

8. Xinmiao Zhang and Keshab K. Parhi, "Implementation Approaches for the Advanced Encryption Standard Algorithm", 1531-636X/12, IEEE 2002.

9. Chun Yan, Yanxia Guo, "A Research and Improvement Based on Rijndael Algorithm", 2009 First International Conference on Information Science and Engineering,Nanjing, Jiangsu China, December 26- December 28, ISBN:978-0-7695-3887-7