Hybrid Steganography Toolkit for Mobile Phones using in-Air Signature Security

Aastha Raina[1] Jyoti Singh[2] AnujaKamble[3] NishuKumari[4]
[1,2,3,4] Department of Computer Engineering
[1,2,3,4]Bharti Vidyapeeth's College of Engineering For Womens, Pune, India

Abstract— improving the security of multimedia data is one of the crucial points required to assure the personal information. This paper presents a biometric authentication procedure consisting of verifying the identity of valid users by making a signature in the air while holding the mobile phone embedded with accelerometer. The verification of the identity will open up the application of steganography which consist of image on which various processing techniques can be applied The proposed scheme is based on hybrid steganography techniques which uses combination of Chaos, Difference expansion and Alpha channel embedding algorithms, to achieve high capacity digital data security and provide copyright protection to the multimedia data. Combination of both provides superior security control. Thesuggested algorithm is messy watermarking technique. Watermarking is a process that embeds data, called a watermark, digital signature, or tag, into a multimedia object. The watermark can then be extorted or detected or tampered to make a claim about the multimedia object. This paper also proposes the scheme to verify the tampered watermarks. Watermarks have a variety of uses that include personal protection and deterrence against theft to maintain data integrity, secrecy and security.

Key words: Steganography, In-Air Signature, RGB color separation, Digital Watermarking, Difference Expansion, Alpha Channel Embedding, Chaos Algorithm

I. INTRODUCTION

In this digital era, ubiquitous network environment has promoted the rapid delivery of digital multimedia data. Users are eager to share the various media information in a cheap way without awareness of the possibly violating copyrights. The issue of copyright protection for digital media has become a problem with rapid advancements in online media storage. As more of these storage systems are made, more information will be taken from them. For all of the good that comes from new technology, an equal amount of illegal activity can spore from it. Watermarking has become one of the most popular copyright protection methods. This paper presents different ways in which the multimedia data can be secured and protected. One can embed visible and invisible watermarks into pictures to prevent image piracy. In this paper, we proposed a copyright embedding system for Android platform. Using this system, pre-specified copyright information is embedded into pictures.

II. PROPOSED METHODOLOGY

The proposed watermarking scheme in this paper works in four stages excluding the authentication stage. The former stage is for verifying the user. And then comes the main steganography application, which open and performs the work on an image based on the later stages.

In this the first stage analyse the colour plane for generating watermark using the RGB separation algorithm. The second stage generates the watermark using the reference colour plane, using chaos algorithm.Then the embedding process is carried out using difference expansion in the third stage. The fourth stage performs the extraction and verification process. Thus in these four stages the image can be protected. These stages are explained further.

A. Authentication Stage:

In this stage In-Air signature, user will make signature in air for authentication purpose. First three to four attempts of signature of one person will be taken and will get stored in database. After at the time of authentication, user will make signature in air same as made before and the application will check both the signatures with Euclidian algorithm by comparing its x,y,z coordinates values. If both signature match, then steganography application will get open for that authenticated user. As this is an android application, we have the hardware named accelerometer already embedded in the smart phones which is the required sensor for this step. This sensor automatically senses the coordinate values when the phone is rotated in any sequence and pattern. The algorithm which gets the values using an accelerometer is given further.

– All the Air-Signatures are Stored in Database

X, Y, Z Co-ordinates	Shapes	Threshold	Average	Result
x_1= 2.46754 y1 = 6.34234 z1= 9.34226		Threshold = 2	r1= Average (x1, y1, z1) r1=5.0507133	Stored in DB
x1 = 4.45675 y1= 5.44334 z1= 0.54789		Threshold = 2	r2 = Average(x1, y1, z1) r2=3.4826600	Stored in DB
x1= 9.54321 y1= 6.88767 z1= -0.89543		Threshold = 2	r3 = Average(x1, y1, z1) r3=4.7754366	Stored in DB
x1= -0.45786 y1 = 8.07842 z1 = 1.34678		Threshold =2	r4 = Average(x1, y1, z1) r4=3.2943533	Stored in DB

– Checking the current signature with the Stored Signature in Database

X, Y, Z Co-ordinates	Shapes	Thres hold	Average	Result
x2=3.54346 y2=8.56345 z2=6.99324		Thresh old = 2	t1 =4.70005 m1=| r1 –t1| = 0.85066 < 2(Noise)	REJECTED
x2=5.78968 y2=7.66786 z2=8.32471		Thresh old = 2	t2=7.26075 m1=|r1 - t2| = 2.210036 > 2(Noise)	ACCEPTE D
X2=6.4532 1 Y2= 3.83653 Z2 = 1.43287		Thresh old = 2	t3=3.907536 6 m2=|r3 – t3| =0.1678999 < 2(Noise)	REJECTED
X2 = 9.43523 Y2 = 8.34123 Z3 = 2.12786		Thresh old = 2	t4=5.634773 3 m2=|r3 – t4| = 2.689768 > 2(Noise)	ACCEPTE D

B. First Stage: Analysis of the Color Plane:

Pixels are stored as Integers. The integers can be 8-bit, 24-bit or 32-bitdepending on the image type. Most popular are 24 bit colour images where 8bits each for Red, Green and Blue colour values are used to represent a 24-bit pixel value. 8 bit images are gray scale images whereas 32bit images have an additional transparency channel.

Sample PIXEL value in HEX = 0EDEB5

- In programming the hex numbers are represented as 0x0EDEB5. 0x prefix is for hex notation.
- Then individual colour channels:
- 0E (red) - DE (green)- B5 (blue)
- 00001110 – 11011110 – 10110101

Traverse Through Entire Image
for(y=0;y<height;y++) {
for(x=0;x<width;x++) {
pix = input[y][x];
Extract 8-bit R, G and B values from
24-bit Color Value
b = pix& 0xff;
g = (pix>> 8) & 0xff;
r = (pix>> 16) & 0xff;

1) Example to Explain RGB Separation Algorithm:

Assume PIXEL value is 0x435A56 where 0x43 is red, 0x5A is green and 0x56 is blue component. Now to separate blue we can use the LOGICAL AND operator to mask or filter the blue component from the rest. Since AND'ing with 1 makes no difference where asAND'ing with 0 will force the bit to0.

435A56
AND 0000FF

0x000056 - blue separated

For Green we shall first right shift the pixel value by 8 bits so that green component is now at LSB position. And then repeat the masking process.

435A56 >> 8 = 435A
0x435A
AND 0x00FF

0x005A - green separated

Similarly we shall right shift by 16 bits so that red component will be at the LSB position and then do the masking

C. Second Stage: Generate Watermark Using Messy Systems

This stage uses the chaos algorithm in which noisy planned randomness is generated using a particular mathematical equation.

Chaos is a kind of behaviour about nonlinear dynamics law control. This paper adopts Logistic-mapping method to generate chaotic sequence:

$$a_{k+1} = \mu \cdot a_k \cdot (1 - a_k), k = 0,1,2,\cdots$$

The value traverses in the interval [0, 1], and μ is a control parameter or a bifurcation parameter. When 3.5699456…< u <= 4 , the logistic map works in chaotic state. The data stream generated is disordered, and it's similar to random noise.

Messy system is a dynamical system whose behaviour changes with time. These changes are very sensitive to theinitial conditions.Thus, the behaviour of messy system appears to be random,though they are deterministic. The dynamic changes of thissystem are completely defined by their initial conditionswithout any random elements. Therefore, the watermark isgenerated through messy system using the reference colourplane as initial condition. Thereby, the watermark is generated dynamically. A general messy system is defined by thefollowing equation:

$$x_{n+1} = f(x_n)$$

The watermark is generated through messy system using the reference color plane as initial condition.The initial value is designed by:

C_seq(k,0)=a*floor(s(k)/2^l)*2+b*pos+c*key
where,
s(k)=Pixel values of reference color plane
a,b,c= Predefined constants
l = Embedding depth
pos= Position information
key=secret key.

For the kth pixel the sequence is referred as c_seq (k, i) and i=l, 2, 3 ... 1 .

Now, messy sequence is generated by substituting C_seg(k,0) value for x(n) in the next equation

f(x(n)) = 4 sin^ 2 (x(n) - 2.5)

After obtaining the value of x(n) substitute into

$$x_{n+1} = f(x_n)$$

Data of the previous pixel is stored into the next pixel.Sequence contain floating no which is converted into binary sequence .i.e images with only black and white color. Thresholding is used to convert the sequence c_seq(k,i) from floating to binary sequence w(k,i).

$$w(k,i) = \begin{cases} 1 & c\_seq(k,i) > T \\ 0 & Elsewhere \end{cases}$$

Fig. 1: Image

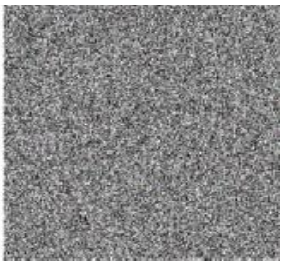

Fig 2: Watermarked Image

D. Third Stage: Embedding Watermark

The watermark generated in previous stage will now be embedded by using intra-plane difference expansion.

- In the embedding stage, the original image Img (I, J, K) is divided into color planes. Here
  I= number of rows,
  J= number of columns,
  K= number of planes.
- Since, the input image is in RGB (Red, Green, Blue) mode, here k=3.
- Pixel pair is formed from the red and blue color planes of the image.

By checking overflow and underflow condition watermark is embedded by expanding difference.

$$m = \left\lfloor \frac{x+y}{2} \right\rfloor$$

Integer transform is given by (m,d) where

m= integer average

d=difference

$$d=x-y$$

In the integer transform, the difference(d) is modified based on the watermarking bit to hide the bit into the pixel pair. The modification of the difference d`

d`=2*d + bit

Only expandable difference is used for the embedding.

Example to explain the difference expansion:

x=110

y=97

m= (110+97)/2=103

d=13 converted to binary 11011 watermarked bit.

E. Fourth Stage: Extraction and Verification

In the extraction process, the watermarked image w_img is processed in the same way as the original image processed for embedding. Extraction process produces the reference sequence using messy system and green colorplane as seed. The embedded watermark is extracted by applying inverse integer transform

Where, the LSB (Least Significant Bit) of the difference value gives the embedded watermark bit.

$$x = m + \left\lfloor \frac{d+1}{2} \right\rfloor$$

$$y = m - \left\lfloor \frac{d}{2} \right\rfloor$$

Then for verification purpose the most reliable alpha channel process is used in the proposed scheme.

For this process the gray scale value is calculated, the average of the values of the three colors in a pixel. This value is known as alpha. This calculated 8 bit value is stored at the beginning of the RGB value.

This value if tampered, the image gets tampered and by comparing the values of alpha, user comes to know that the image has been attacked.

Fig. 3: Watermark Embedding

Fig. 5: Stego Image Fig. 6: Tampered Image

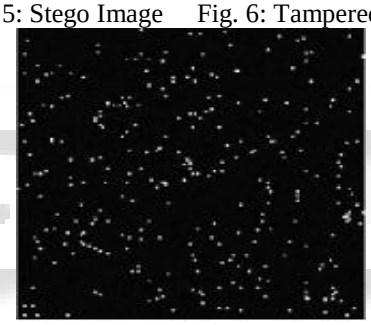

Fig. 7: Modified Template

III. EXPERIMENTAL RESULTS

The experimental work of the former step i.e., authentication step, of this paper has been developed from a database of in-air signatures made up of the samples of 10 people who have performed their signature in the air while holding a mobile device in their hand. This mobile device included a 3-D accelerometer programmed toobtained the acceleration values on each axis at a samplingrate of 100-180 Hz. (different smart phones give different sampling rates). The embedded accelerometer measures the acceleration force in m/s^2 that is applied to a device on all three physical axes, including the force of gravity. It detects motions like shake, rotate, tilt etc.

To quantitatively justify the performance of the later stages, watermarking stages, on various requirements and parameters, experiments were conducted on the images, which can be taken from gallery or can be captured by camera using the smartphone. These images can be of any format but later need to be stored in bmp format of any size supported by RAM. The colour mode is RGB (Red, Green, Blue). All the experiments are conducted using java and android platform on processors with 4 2 & 1 GB RAM.

On basis of the above experiments performed, security, authenticity, reliability and fragility of the mentioned scheme were verified as a result.

Fig. shows percentage of modification in watermark as per modification in image in number of bits.

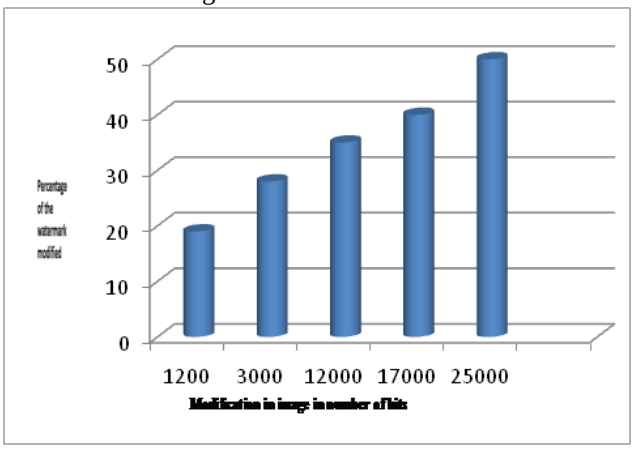

Fig. 8: Keep Watermark as per image

IV. Conclusion

This proposed system uses a biometric authentication procedure consisting of verifying the identity of users by making an air signature holding smart-phone. After the verification of user, it will open up the application that will apply the steganography techniques on the image which is loaded. First, it separates RGB pixels from the image and dynamically generates the watermark by applying chaos algorithm. Then, watermark is embedded inside the image by expanding difference between any two colour planes, this is known as intra plane difference expansion algorithm. All this comes under primary watermarking. It could precisely locate the tampered region. At last, Alpha channel embedding algorithm is used for providing transparency, this is secondary watermarking.

The final image generated is send over the network and authentication checks that the image is modified or not.

V. Acknowledgements

First and foremost we would like to express our gratitude to Prof. D.D Pukale, our internal guide and HOD, for his guidance and support throughout the project. Without his cooperation, it would have been extremely difficult for us to complete the project part of this semester. . We would like to thank the entire teaching and non-teaching staff of the Computer Department for giving us an opportunity to work on such an exciting project. Last but not the least, we are extremely grateful to our family, friends and colleagues who have supported us right from the inception of the project. Thanks for all your encouragement and support.

References

[1] S. Poonkuntran, R. S. Rajesh, "A Messy Watermarking for Medical Image Authentication" Department of Computer Science and Engineering ManonmaniamSundaranar University Proceedings of the 2012 IEEE transactions.

[2] Smita P. Bansod, Vanita M. Mane, Leena R. Ragha,"Modified BPCS steganography using Hybrid Cryptography for Improving Data embedding Capacity"2012 International Conference on Communication, Information & Computing Technology (ICCICT), Oct. 19-20, Mumbai, India

[3] M. Natarajan, GayasMakhdumi, "Safeguarding the Digital Content: Digital Watermarking" Department of Library & Information Science, JamiaMilliaIslamia University, New Delhi,2009,Page No[29-35]

[4] Javier Guerra-Casanova, Carmen Sa´nchez A´ vila, Gonzalo Bailador" Time series distances measures to analyze in-air signatures to authenticate users on mobile phones" Universidad Polit´ecnica de Madrid Campus de Montegancedo 28223 Pozuelo de Alarc´on, Madrid, SPAIN.

[5] Yueh-Hong Chen, Hsiang-Cheh Huang, "A Copyright Information Embedding System for Android Platform" 2011 Seventh International Conference on Intelligent Information Hiding and Multimedia Signal Processing.

[6] PravinRaut and SnehalGolait, "Review on surround sense handgestures for mobile devices" Department of Computer Engineering RTM Nagpur University, IEEE 2014DOI 10.1109/ICESE.2014.32,Page No[156-159]

[7] P.Shanthi, R.S.Bhuvaneswaran, "Robust chaos based image watermarking scheme for Fractal- Wavelet" Applied Mathematical Science,Anna University, Chennai2014, Page No [1593-1604]

[8] J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, "Digital Watermarking and Steganography". London: Elsevier Science & Technology, November 2007.

[9] J.-S. Pan, H.-C. Huang, and L. C. Jain, Eds., Intelligent Watermarking Techniques. London: World Scientific Publishing Company, April 2004.

[10] F.AAllaert, L.Dusserre, "Security of Health System in France. What we do will no longer be different from what we tell", International Journal of Biomedical Computing, vol. 35, no. Suppl. 1, pp. 201- 204,1994.

