Hi-Tech Authentication for Palette Images Using Digital Signature and Data Hiding

Arockia Jansirani, Rengansivagurunathan Rajesh, Ramasamy Balasubramanian, and Perumal Eswaran
Department of Computer Science and Engineering, Manonmaniam Sundaranar University, India

Abstract: *A scheme that integrates digital signature and data hiding to provide hi–tech authentication for palette images is proposed in this paper. The scheme extracts digital signature from the original palette image and embeds it back into the same palette image, avoiding additional signature file. Digital signature generation is employed using elliptic curve based public key cryptosystem. The performance of elliptic curve based public key cryptosystems is mainly appointed by the efficiency of the underlying finite field arithmetic. Instead of directly sending an original palette image to recipients, only the embedded copy is sent associated with signed digital signature. Experimental results show that security is achieved without sacrificing the image quality.*

Keywords: *Digital signature, elliptic curve cryptography, karatsuba multiplication, data hiding, palette image, and color mapping function.*

Received June 17, 2008; accepted May 17, 2009

1. Introduction

In a decade ago, multimedia documents are rarely available to the mass consumer market. However, as the rapid development of the pervasive digital information technology, everyone's computer can have high quality image compression, increasing network bandwidth and accessibility, dense portable storage media, and compounding processing power. Nevertheless, these technological advances lead to another crisis. Multimedia users had the ability to tamper with, produce copies of, and illegally redistribute digital contents. Without solving this security issue, digital multimedia products and services cannot take–off in an e–commerce setting. To solve this problem, a comprehensive approach for palette image authentication using digital signature and data hiding technique is introduced here.

Palette images are popular in multimedia and internet applications. Each palette image is composed of a color palette and a set of color indexes.

The color palette is a list of entries of representative colors in the image, and the color indexes are some pointers to those palette entries that specify the red-green-blue (RGB) colors in the image. Use of this type of palette image format has the effect of image compression, which helps saving storage space and reducing transmission time. An example of palette image is that of the Graphics Interchange Format (GIF) [6].

Digital signature and cryptography [10] are currently two standardized approaches to protect digital contents. Digital signature [4] is an electronic signature that is used to authenticate the identity of the sender of the palette image and to ensure that the original document of the palette image that has been sent is unchanged. First the image sender extracts some information dependent on the content of the original palette image and encrypts it into a small size file, which is called signature. Then the signature file is sent to the recipients along with the original palette image. The recipients use the same algorithm to extract the content-dependent information of the received palette image. If the recipients–extracted information matches with the signature, the ownership and the integrity of the received palette image are authenticated.

An obvious drawback of conventional digital signature schemes is the extra bandwidth needed for transmission of the signature. To overcome this drawback, the combined digital signature [14] and digital data hiding [12] scheme is proposed for palette image authentication. The basic idea of the combination is as follows: The image provider extracts the content–dependent signature from the original palette image, and then embeds it back into the same image as a hidden data. These palette images can conceal secret data without arousing suspicion when the resulting stego-images are inspected or transmitted over the Internet.

The receiver extracts the signature and the hidden data from the received image at the same time. If the signature and the hidden data match, the received palette image is thought to be authentic. Chih-Husan [15] proposed robust data hiding technique in palette images. This idea is extended to digital signature applications here, forming the content-based digital

signature embedding and extraction scheme, which is robust against conventional compression algorithms.

In this paper, an efficient Elliptic Curve Cryptography (ECC) technique using a new Galois field processor in the implementation of elliptic curve groups is used to generate the digital signature for palette images and data hiding technique using binary valued color- mapping function to embed the generated digital signature onto the same palette image.

The remainder of this paper is organized as follows: in section 2, a brief description of the digital signature scheme is given. In the next section an overview of the data hiding technique is given. In Section 4 the proposed architecture is presented. Experimental results are described in section 5. Conclusion and future work is given in section 6.

2. Digital Signature

Digital signatures are analogous to the hand written signatures. Digital signatures and hand written signatures are based on the fact that it is very hard to find two people with the same signature. A major difference between handwritten and digital signatures is that a digital signature cannot be a constant; People use public key cryptography to compute digital signatures by associating something unique with each person. When public key cryptography is used to compute digital signatures, the sender encrypts the image with his own private key. This signature can later prove the ownership, identify a misappropriating person, trace the marked document's dissemination through the network, or simply inform users about the rights–holder or the permitted use of the data.

A Digital Signature Algorithm (DSA) was specified in a U.S. Government Federal Information Processing Standard (FIPS) called the digital signature standard (DSS). Its security [13] is based on the computational intractability of the Discrete Logarithm Problem (DLP) in prime–order sub–groups of Z_P*. The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the DSA. ECDSA was first proposed in 1992 by Scott Vanstone in response to National Institute of Standards and Technology (NIST) request for public comments on their first proposal for DSS. It was accepted in 1998 as an ISO (International Standards Organization) standard (ISO 14888-3), in 1999 as an ANSI (American National Standards Institute) standard (ANSI X9.62), in 2000 as an IEEE (Institute of Electrical and Electronics Engineers) standard (IEEE 1363-2000) and a FIPS standard (FIPS 186-2). It is also under consideration for inclusion in some other ISO standards.

Figure 1 shows that Elliptic curves are not ellipses. They are named so, because they are described by cubic equations similar to those used for calculating the circumference of an ellipse. An elliptic curve [3, 11], may be defined as a set of points on the coordinate planes, satisfying the equation of the form

$$y^2[+xy] = x^3 + ax^2 + b \quad (1)$$

The square bracket means that the term is optional. x and y are variables, a and b are constants. However these quantities are not necessarily real numbers; instead they may be values from any field i.e., x, y, a & b are chosen from a finite set of distinct values.

2.1. Elliptic Curves over Galois Field

This section defines a group constructed from points on elliptic curves over Galois Field (2^m) [7, 2] and the efficient implementation of operations in this group. A non–singular elliptic curve E over GF (2^m), E (GF (2^m)) is the set of solutions to the following equation with co–ordinates in the algebraic closure of E.

$$y^2 + xy = x^3 + ax^2 + b \quad (2)$$

where a, b are in GF (2^m), and b is non-zero. Such an elliptic curve is an abelion group.

The number of points in this group is denoted by # E (GF (2^m)).The crucial property of an elliptic curve [9] is that, the resultant point obtained by adding two points on the curve is also on the curve. The addition rule satisfies the normal properties of addition. If P = (x_1, y_1) and Q = (x_2, y_2) are points on the elliptic curve the addition rule has the form

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3) \quad (3)$$

where $x_3 = L^2 + L + x_1 = x_2 + a \quad (4)$

$$y_3 = L*(x_1 + x_3) + x_3 + y_1 \quad (5)$$

$$L = (y_1 + y_2)/(x_1 + x_2) \quad (6)$$

& a is in GF (2^m) If $x_1 = x_2$ and $y_1 = y_2$ then

$$x_3 = L^2 + L + a \quad (7)$$

$$y_3 = x_1^{\,2} = (L+1)*x_3 \quad (9)$$

$$L = x_1 + (y_1 / x_1) \quad (10)$$

Again there are some special cases which must be considered: if $x_1 = x_2$ and $y_2 = x_1 + y_1$ then the result is zero, and if either point is zero, whereas if P and Q are not equal it is called point addition. Multiplication is defined by repeated addition i.e.,

$$Q = kP = P + P + P + ...k \text{ times} \quad (11)$$

This can be computed using point addition and point doubling. In particular for an elliptic curve E, it relies on the fact that it is easy to compute

$$Q = kP \quad (12)$$

for k in GF(2^m) and P, Q in E.

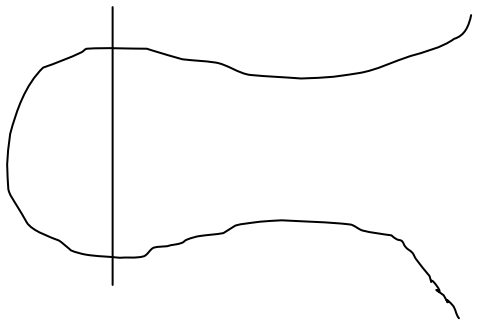


Figure 1. Elliptic curve.

The difficulty of the problem depends on the group, and at present, the problem in elliptic curve groups is orders of magnitude harder than the same problem in a multiplicative group of a finite field. This feature is a main strength of elliptic curve cryptosystems.

To perform multiplication[1, 8] of large numbers in (many) fewer operations than the usual brute–force technique of "long multiplication" in GF(2^m) karatsuba (Karatsuba and Ofman 1962), multiplication of two *n*-digit numbers can be done with a bit complexity of less than n^2 using identities of the form $(a+b.10^n(c+d.10^n)$ is equal to

$$ac+[(a+b)(c+d)-ac-bd]10^n+bd.10^{2n} \quad (13)$$

Proceeding recursively then gives bit complexity $O(n^{\log 3})$, where $\log^3 = 1.58 \ldots <2$ (Borwein et al. 1989). The best known bound is $O(n\log n)$ steps for $n >>1$ (Schonhage and Strassen 1971, Knuth 1981).

The steps involved in Elliptic Curve Digital Signature Algorithm (ECDSA) are key pair generation, signature generation and signature verification. For signature generation and verification, the well known Hash algorithm is used. For key pair generation Karatsuba multiplication and point addition are employed.

In key pair generation, the random or pseudorandom integer K_S is selected to be in our field GF(2^m). P is a point on the elliptic curve, known as the generating point and is obtained by multiplying two other points on the elliptic curve by karatsuba multiplication. The public key (K_P)is obtained by scalar multiplication (point addition) of K_S and P which is again a point that lies on the elliptic curve. The private key is kept as secret whereas the public key is known to the sender and receiver. The receiver who knows about the sender's public key can authenticate the signature using his private key. This ensures that anyone with access to the public key of the signer may verify the signature.

P_1(i-1,j-1)	P_2(i,j-1)	P_3(i+1,j-1)
P_4(i -1,j)	X(i, j)	

Figure 2. A Pixel X and its four precedent neighbours (P).

3. Color-Ordering Relationship and Color Mapping Function for Data Hiding

The idea of data hiding is to embed the secret information (digital signature) by modifying the given palette image [5] without creating noticeable artifacts. The recipient can correctly extract the embedded information from the stego–image, while the other people are unaware of the existence of the secret behind the stego–image.

This is a new data hiding technique which embeds data by modifying the given image attributes like its colors (palette images like GIF).This technique is based on the use of a new type of color-ordering relationship, from which a color–mapping function is defined with binary values as output. First, image pixels are classified as data embeddable or non-embeddable, and only the former ones are used to embed secret data.

When a secret data bit is to be embedded, the data embeddable pixel's color is adjusted based on the color mapping function output so that the secret information hidden in the stego–image is visually and statistically undetectable by the intruder. This technique provides a good balance between stego-image quality and data-embedding capacity. This adaptive method can be employed to conceal a moderate amount of data and has the least modification of pixel values.

Given a pixel *X* in the palette image, its precedent neighbors are those four neighbouring pixels, among the eight neighbouring ones in a 3×3 neighbourhood as shown in figure 2. The color–ordering relationship is defined as follows:

$$R_{co}=\begin{cases} c_1>c_2,\ if\ (v_1>v_2)\ or \\ \qquad (v_1=v_2\ and\ r_1>r_2)\ or \\ \qquad (v_1=v_2\ and\ r_1=r_2\ and\ g_1=g_2) \\ c_1=c_2\ \ if\ (r_1=r_2\ and\ g_1=g_2\ and\ b_1=b_2) \\ c_1<c_2,\ otherwise \end{cases} \quad (13)$$

where c_1 and c_2 be two colors with RGB values (r_1, g_1, b_1) and (r_2, g_2, b_2) respectively. The luminance value V_1 and V_2 of c_1 and c_2 is calculated as

$$\begin{aligned} V_1 &= 0.3\times r_1+0.59\times g_1+0.11\times b_1 \\ V_2 &= 0.3\times r_2+0.59\times g_2+0.11\times b_2 \end{aligned} \quad (14)$$

Given a pixel *X* in the palette image, its precedent neighbours are defined to be those four neighbouring pixels, among the eight neighbouring ones in a 3×3 neighbourhood of, which are visited in sequence before the other four during the line–by–line raster scanning. More specifically, if it is located at coordinates (i, j) in the input image, then its precedent neighbors are the four pixels located at $(i-1, j)$, $(i, j-1)$, $(i+1, j-1)$ and $(i-1, j-1)$. It is shown in Figure 2. The color mapping function f_{em} is defined as:

$$f_{em} = \begin{cases} 0, \ if \ c > c_1^{\ 1} \\ 1, \ if \ \ c_1^{\ 1} > c > c_2^{\ 1} \\ 0, \ if \ c_2^{\ 1} > c > c_3^{\ 1} \\ 1, \ if \ \ c_3^{\ 1} > c > c_4^{\ 1} \\ 0, \ otherwise \end{cases} \quad (15)$$

where $c_1{}^1$ through $c_4{}^1$ are the result of sorting the values of c_1 to c_4 according to the color–ordering relationship with $c_1{}^1$ being the largest. It can be seen that the function depends on the ordering of the color c of X among those of the four precedent neighbors of X. In addition, to reducing possible quality degradation in the resulting stego–image, the pixels in the cover image are classified into data embeddable and nonembeddable ones during the raster–scanning process. Only data–embeddable pixels are used for digital signature hiding; nonembeddable ones are skipped. Let c be the original color of a given pixel X and c^1 a possible replacement for c in the color palette. When the color of X is c, assume that the corresponding output of the color-mapping function of X is b, and that the corresponding maximum color difference between X and its four precedent neighbors is β. When the color c of X is replaced by c^1, assume that the corresponding values of b and β are changed to be b^1 and β^1 respectively. Also assume that the number of distinct colors of X's four precedent neighbours is α. A pixel is defined to be data embeddable if the following three conditions are satisfied:

1. α is larger than a threshold value T_d.
2. β is smaller than a threshold value T_c.
3. There exists a color c^1 with the corresponding b^1 being the inverse of b, and the corresponding β^1 being smaller than the threshold value T_d. Or equivalently, the data embeddability of a pixel is defined as follows: X is data embeddable, if $\alpha > T_c$, $\beta < T_d$, and there exists a c^1 such that $b^1 \neq b$ and $\beta^1 < T_d$.

4. Proposed Work

The proposed authentication scheme is a kind of sender–receiver protocol. The sender generates the signature and inserts it back into the original palette image as a hidden data. In the receiver's side, the ownership and integrity is verified by comparing the signature and embedded data both extracted from the received palette image. The procedures in both sender and receiver sides are described in detail below.

4.1. Digital Signature Generation

In signature generation, a generator point G is a scalar multiplied with a constant k in the field of GF (2^m) resulting in a point (P) on the elliptic curve. The public key (K_P) is computed using

$$K_P = K_S \cdot P \quad (16)$$

where, K_S is the secret key which is actually a random number from [1: n-1]; n is the number of pixels in the palette image I_P.

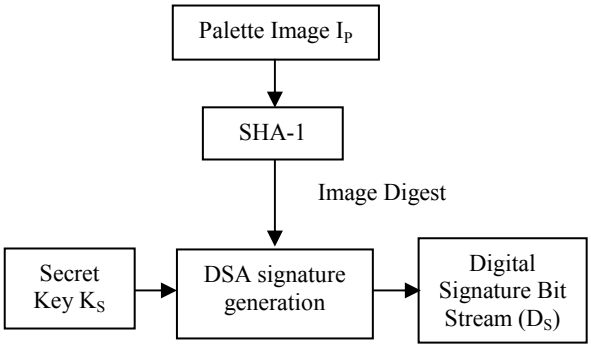


Figure 3. Digital signature generation.

The digital signature (D_S) is computed as follows:

$$D_S = q^{-1}\{SHA-1(I_P) + K_S.K_q\}(\mathrm{mod}\, n) \quad (17)$$

where, SHA–1 is the 160–bit hash function, $K_q = x_1(\mathrm{mod}\, n)$, x_1 is the x coordinate of the point P. Figure 3 shows the generation of digital signatures. This process uses the hash function of the palette image thereby resulting in the image digest. Hashing may be defined as the transformation of a string into a usually shorter and fixed length value or a key that represents the input palette image (I_P). During signature generation, the transmitter's secret key (K_S) is used along with the image digests to generate a bit stream (D_S). In this application, these bits are considered content dependent digital signature and will be embedded back into original palette image as a hidden data. Thus the signature for the image I_p is (D_S, K_q).

4.2. Digital Signature Embedding Process

The digital signature to be embedded is represented as a bit stream, denoted as $D_S = d_1\, d_2 \ldots\ldots d_n$. The basic idea of the data-embedding process is to check each pixel of the original palette image (I_P) in a raster–scanning manner for its data embeddability, and to embed each secret bit d_n of D_S sequentially into every data-embeddable pixel until the bit stream of D_S is exhausted. During each secret bit (digital signature bit) embedding step, if the binary output of the color–mapping function f_{em} is the same as the secret bit value to be embedded, the color c of the currently checked data–embeddable pixel is kept unchanged; otherwise is replaced with a color c_{opt}, called the optimal replacement color for X, this process is shown in Figure4. For a particular color c (data embeddable pixel color) the replacement color (c_R) is the color

with minimum color difference, selected from its palette.

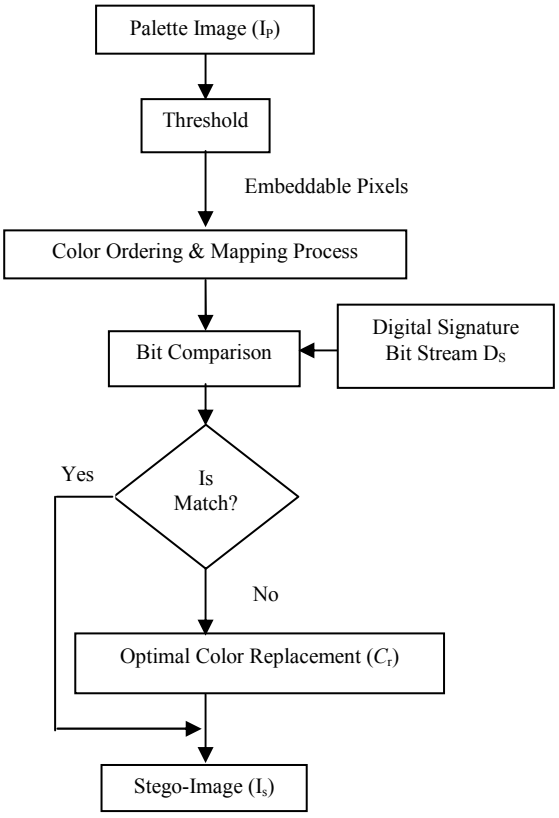


Figure 4. Digital signature hiding within the same palette mage.

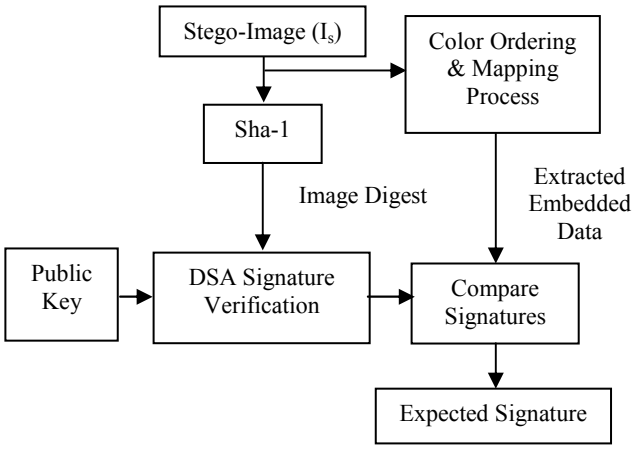


Figure 5. Digital signature extraction and verification.

This is given by, $|c - c_R| = \min|c - c_i|$, where c_i is the color from palette P that satisfies the conditions:

1. c_i together with its neighbors as input to f_{em} yields binary output b_0 that is the same as the digital signature bit b;
2. X is still data embeddable when its color is set to c_i.

If the color difference $|c - c_R|$ is smaller than the predefined threshold T_C or if N is empty then take c_R as the desired optimal replacement color c_{opt} for X and stop; otherwise, find the color c_i among those in N, whose color difference from c is the minimum, i.e., $|c - c_R| = \min|c - c_i| c_i \varepsilon N$, then take c_R as c_{opt} N denotes the subset. That contains the colors of the four precedent neighbors of X. Figure 2 the color difference $|c - c_i|$ between two colors c and c_i is the Euclidean distance between the RGB values (r, g, b) and (r_i, g_i, b_i) of c and c_i respectively. where

$$|c - c_i| = \left[(r - r_i)^2 + (g - g_i)^2 + (b - b_i)^2\right]^{1/2} \quad (18)$$

The resultant stego–image contains the digital signature embedded into it.

4.3. Verification

At the receiver, the received stego–image (I_S) is subject to two parallel processing namely, the DSA signature extraction process and the embedded data extraction process.

4.3.1. DSA Signature Extraction Process

The hash function for the stego–image (I_S) is Computed using $r = SHA-1(I_S)$ and also $D_{S1} = D_S^{-1} (\mathrm{mod}\, n)$ is calculated. The signature is verified using the following:

$$r_1 = r * D_{S1} (\mathrm{mod}\, n) \quad (19)$$

$$r_2 = K_q * D_{S1} (\mathrm{mod}\, n) \quad (20)$$

$$K_m = x_1 (\mathrm{mod}\, n) \quad where \quad x_1 = r_1.P + r_2 * K_p \quad (21)$$

The signature is accepted if K_m is equal to K_q.

4.3.2. Embedded Data Extraction Process

In this process the data embeddable pixels are identified from the stego–image (I_S). These pixels are given as input to the next stage i.e., the color of each embeddable pixel and those of its four precedent neighbours are given as input to the color ordering and mapping function. If the output is '1' then the extracted secret bit is taken to be '1' otherwise, '0'.

The extracted digital signature is compared with the extracted embedded data for verification. The verification process is shown in Figure 5.

4.4. Advantages

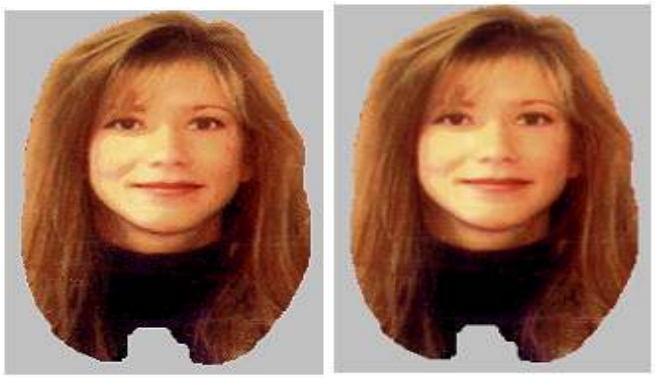
1. This method does not manipulate image palettes, resulting no abnormal palette structure.
2. Prevents the resulting stego-images from having outstanding pixels which are visually or statistically detectable.
3. Since this method does not alter the palette, it doesn't include special patterns (such as Twin peaks).

4. This technique provides a good balance between stego–image quality and data-embedding capacity.
5. Stego–image is visually and statistically undetectable by the intruder.

5. Experimental Results

This paper demonstrates the feasibility of constructing very fast and very secure public key systems with the use of karatsuba logic for multiplication. Some experiments are designed to prove the efficiency of the proposed scheme. First, the palette image quality after digital signature insertion is investigated. Secondly, the maximum number of bits that can be embedded into the given palette image is calculated. The 247 × 171 Girl. GIF image is used for experiments.

The Peak Signal to Noise Ratio (PSNR) is computed to evaluate the embedded image quality. PSNR is given by, PSNR = 10 $\log_{10}$ ($255^2/ \sigma^2$) where σ^2 is the mean square of the difference between the original palette image and the embedded one. Figure 6 shows the Girl images before and after digital signature insertion. No obvious degradation is observed in Figure 6 (b) who's *PSNR* is 35.1612.



(a) Original girl image. (b) Girl image after embedding digital signature.

Figure 6. 247 x 171 Girl .GIF image.

Table 1. Maximum number of embeddable pixels and quantitative. Measurement for Various Images.

Image	Total embeddable pixels		PSNR
	T_c=1,T_d=30	T_c=2,T_d=20	
Girl (247×171)	4721	33	35.1612
Fish (310×192)	4030	80	35.9060
Venus (256×256)	15590	5355	36.1151
Chrome_2 (320×320)	17443	1157	37.0842
Venus (512×256)	26938	5566	37.6203

From Table 1, it is observed that the number of bits that can be embedded is increased, when T_c is decreased and T_d is increased. Also it is clear that this method provides good trade-off between embedding capacity and image quality, and so is quite flexible.

When palette images contain limited colors that are visually uncorrelated, the proposed method can yield embedding results with better visual quality. From the PSNR values, it is observed that this technique doesn't introduce any visual artifacts and the signature can be extracted correctly.

6. Conclusions and Future Work

Digital signature and data hiding are two techniques used for copyright protection and authentication, respectively. In this paper, a combined signature and watermark scheme is proposed for image authentication. Conventional digital signature schemes usually encode the signature in a file separate from the original image, thus require extra bandwidth to transmit it.

The proposed scheme extracts signature from the original image and embeds them back into the image as hidden data, avoiding additional signature file. Furthermore, the scheme not only can verify the authenticity and the integrity of images, but also can locate the illegal modifications. Experiments show that our scheme is robust to reasonable compression rate while preserving good image quality, and capable to authentication.

Future work will be focused on more robust signature extraction method and possible ways to recover the illegally modified stego–image. Since this adaptive technique does not manipulate image palettes, the intruder cannot arouse suspicion for abnormal palette structure and special patterns (like twin peaks). It also prevents outstanding pixels. This technique provides good embedding capability keeping the stego–image quality. But the T_c and T_d values should be taken into account according to the length of digital signature.

References

[1] Agnew G., Mullin R., Onyszchuk I., and Vanstone S., "An Implementation for a Fast Public-Key Crypto Systems," *Computer Journal of Cryptology*, vol. 3, no. 2, pp. 63-79, 1991.

[2] Agnew G., Mullin R., and Vanstone S., "An Implementation of Elliptic Curve Cryptosystems over F_2155," *Computer Journal of IEEE on Selected Areas in Communication*, vol. 2, no. 5, pp. 804-813, 1993.

[3] Centricom Research, "The Elliptic Curve Cryptosystem," Certicom, http://www.certicom .com /index.php/ecc, Last Visited 2008.

[4] Elgamal T., "A Public Key Crypto Systems and a Signature Scheme Based on Discrete Logarithms," *Computer Journal of IEEE*

Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, 1985.

[5] Fridrich J. and Du R., "Secure Stegnographic Methods For Palette Images", *in Proceedings of 3rd International Workshop on Information Hiding*, Germany, pp.47-60,1999.

[6] "GIF Color Map Stegnography," http:// www. darkside.com.au/gifshuffle, Last Visited 2008.

[7] Hankerson D., Hernandez J., and Menezes J., "Software Implementation of Elliptic Curve Cryptography over Binary Fields," *in Proceedings of Workshop on Cryptographic Hardware and Embedded Systems*, pp. 1-24, 2000.

[8] Karu P, "Practical Comparison of Fast Public-Key Cryptosystems," http://www.tml.hut.fi/~pk/ crypto/fast_pk_crypto.pdf, Last Visited 2008.

[9] Koblitz N., "Elliptic Curve Cryptosystems," *in Proceedings of Mathamatics of Computation*, pp. 203-209, 1987.

[10] Menezes J., Oorschot C., Vanstone S., *Handbook of Applied Cryptography*, CRC press, 1997.

[11] Miller V., "Use of Elliptic Curve in Cryptography," *in Proceedings of CRYPTO'85, Springer Verlag Lecture Notes in Computer Science*, pp. 417-426, 1986.

[12] Petitcoals F., Anderson R., and Kuhn M., "Information Hiding: A Survey," *in Proceedings of the IEEE Special Issue on Protection of Multimedia Content*, pp. 1062-1078, 1999.

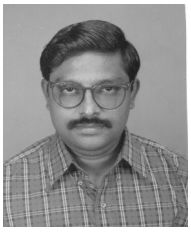
[13] Pointcheval D. and Stern J., "Security Proofs for Signatures," *in Proceedings of Eurocrypt*, pp. 159-162, 1996.

[14] Rivest R., Shamir A., and Adleman L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Computer Journal of Communications of the ACM*, vol. 21, no. 5, pp. 120-126, 1978.

[15] Tzeng C., Yang Z., and Tsai W., "Adaptive Data Hiding in Palette Images by Color Ordering and Mapping with Security Protection," *Computer Journal of IEEE Trans. on Communication*, vol. 52, no. 5, pp. 791-800, 2004.

Arockia Jansirani received the BSc and MSc degrees in electronics and communication engineering in 1996, and 2002, respectively, from Manonmaniam Sundaranar University, India. In 1997, she joined the Department of Electronics and Communication Engineering, Karunya Institute of Technology, Tamil Nadu and worked for a period of two years. In December 2003, she joined Manonmaniam Sundaranar University, Tamil Nadu, where she is currently working as assistant professor in the Department of Computer Science and Engineering. Her research interests include digital Image Processing, Neural networks, data mining, image security, wavelets, and vector quantization.

Renganasivagurunathan Rajesh received his BSc and MSC degrees in electronics and communication engineering from Madurai Kamaraj University, India in the year 1988 and 1989, respectively. He completed his PhD in computer science and engineering from Manonmaniam Sundaranar University in the year 2004. In September 1992, he joined in Manonmaniam Sundaranar University where he is currently working as associate professor in the Computer Science and Engineering Department. His research interests include digital image processing, wireless networks, pervasive computing, and parallel computing.



Ramasamy Balasubramanian received the BSc degree in computer science and engineering, from Bharathidasan University, Tiruchi, India and MSc in computer science & engineering, from Regional Engineering College, Tiruchi. Crentlly, he is doing PhD degree at Manonmaniam Sundaranar University, India. Since 1994, he is working as associate professor in the Department of Computer Science and Engineering, Manonmaniam Sundaranar University. He has authored one computer networks book, more than 30 conference papers. His research interests include image segmentation, image compression, content-based image retrieval, and Data Mining.

Perumal Eswaran received the MSc degree in computer science and information technology from Madurai Kamaraj University, India in 2003, and the MSc degree in computer and information technology from Manonmaniam Sundaranar University, India in 2005. He is currently pursuing the PhD in the Department of Computer Science and Engineering of Manonmaniam Sundaranar University. His research interests include digital image processing, focusing on color image edge detection, data mining, and computer vision.