# Đề xuất một số lược đồ nhúng tin và thủy vân dễ vỡ khóa công khai trên ảnh JPEG

## New Data Hiding and Public Key Fragile Watermarking Schemes for JPEG Images

**Cao Thị Luyên, Tiêu Thị Ngọc Dung, Đỗ Văn Tuấn, Phạm Văn Ất**

***Abstract:*** *This paper proposes some new schemes for data hiding and watermarking in JPEG images by using quantized discrete Cosine transform (DCT) blocks. The DCT coefficients are segmented into groups of* $2^r - 1$ *continuous elements. For each group, r bits can be embedded by changing at most one element of the group. In comparison to some widely used data hiding schemes for JPEG images, our data hiding schemes achieve higher capability and better stego-image quality. The proposed public key watermarking method based on an embedded algorithm is capable to authenticate digital images.*

***Keywords****: Public key fragile watermarking, data hiding in JPEC image, quantiized DCT.*

### I. GIỚI THIỆU

Với sự phát triển của khoa học và công nghệ, việc trao đổi dữ liệu qua mạng hiện đã trở nên phổ biến. Vì vậy, việc bảo mật thông tin nhằm chống lại sự truy cập bất hợp pháp đang là hướng nghiên cứu thời sự của Tin học.

Thủy vân là kỹ thuật nhúng một lượng thông tin (dấu thủy vân) vào dữ liệu đa phương tiện như ảnh số, âm thanh hay video... Dấu thủy vân được sử dụng làm cơ sở để chứng minh bản quyền hoặc xác thực tính toàn vẹn của ảnh thủy vân (ảnh chứa dấu thủy vân). Có thể phân loại thủy vân theo ứng dụng hoặc theo khóa. Nếu dựa vào ứng dụng thì các lược đồ thủy vân được chia thành thủy vân bền vững và thủy vân dễ vỡ, còn dựa vào khóa thì có thủy vân khóa bí mật và thủy vân khóa công khai. Đối với thủy vân bền vững, dấu thủy vân ít bị biến đổi trước các phép tấn công, loại thủy vân này được dùng trong bài toán bảo vệ bản quyền tác giả [13,14,20]. Trái lại, thủy vân dễ vỡ có khả năng phát hiện được sự thay đổi dù là rất nhỏ của ảnh thủy vân, nó được ứng dụng trong bài toán xác thực tính toàn vẹn của ảnh [5,9,10,18]. Đối với thủy vân khóa bí mật [5,9,13], cả hai quá trình nhúng và kiểm tra dấu thủy vân đều dùng chung một khóa bí mật. Trong khi đó, thủy vân khóa công khai [8,10,14,20] dùng khóa bí mật cho quá trình nhúng và dùng khóa công khai để kiểm tra dấu thủy vân. So với thủy vân khóa bí mật, thủy vân khóa công khai có ưu điểm hơn ở chỗ không phải trao đổi khóa giữa các bên nhúng và kiểm tra dấu thủy vân.

Các lược đồ giấu tin và thủy vân được nghiên cứu phổ biến trên ảnh nén bảo toàn như BMP, TIF, PNG [3,4,8-10,14,19]. Đối với các ảnh loại này, việc nhúng tinđược thực hiện trực tiếp trên giá trị điểm ảnh. Tuy nhiên, các kỹ thuật nhúng tin đó khó có thể áp dụng được trên ảnh nén không bảo toàn. Gần đây hướng nghiên cứu tập trung vào ảnh nén JPEG [1,2,6,7,11,12,15,16,18], tiêu biểu là các lược đồ J-Steg [6], Iwata cùng các đồng sự [7] và lược đồ của Liu cùng các đồng sự[12]. Các lược đồ trên đều sử dụng các khối hệ số cosine rời rạc lượng tử (DCTLT) để nhúng tin theo các cách khác nhau. Lược đồ [7] tiến hành nhúng chín bít trên chín đường chéo song song với đường chéo chính của khối DCTLT và làm thay đổi tối đa chín phần tử. Trong khi đó, các lược đồ còn lại thì quét các hệ số DCTLT theo đường zigzag rồi thực hiện nhúng một bít trên một hệ số DCTLT có giá trị khác 0 và ±1. Lược đồ [6] nhúng theo phương

pháp chèn bít thấp, lược đồ [12] thì nhúng tin bằng kỹ thuật bù nhau (complementary). Nhìn chung trong các lược đồ trên, tỷ lệ thay đổi là 0.5 phần tử /1 bít, tức là để nhúng $n$ bít thì cần thay đổi khoảng $n/2$ phần tử (một phần tử thay đổi một đơn vị). Khả năng nhúng tin của các phương pháp này vào khoảng 15.6 bít/1 khối DCTLT (xem Mục V.1).

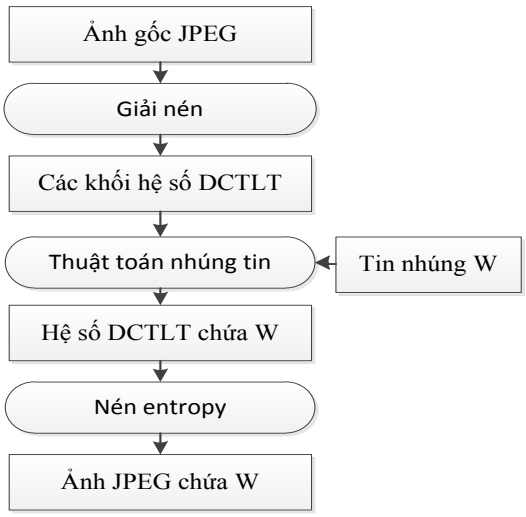
Bài báo đề xuất các phương án nhúng: 9, 10, 12 hay 16 bít trên một khối DCTLT mà chỉ thay đổi tối đa từ 2 đến 4 phần tử. Như vậy, các lược đồ đề xuất không những có khả năng nhúng cao mà còn cho chất lượng ảnh tốt hơn so với các lược đồ trên. Từ các lược đồ đề xuất, chúng tôi xây dựng lược đồ thủy vân dễ vỡ khóa công khai ứng dụng trong bài toán xác thực tính toàn vẹn của ảnh.

Nội dung tiếp theo của bài báo được bố cục như sau: Mục 2 giới thiệu những công trình liên quan. Các lược đồ nhúng tin đề xuất được trình bày trong Mục 3. Mục 4 sẽ đề xuất lược đồ thủy vân dễ vỡ khóa công khai trên ảnh JPEG. Việc so sánh khả năng nhúng tin và chất lượng ảnh giữa lược đồ đề xuất với các phương pháp liên quan được trình bày ở Mục 5. Mục 6 là một số kết luận của bài báo.

## II. CÁC CÔNG TRÌNH LIÊN QUAN

### II.1. Quy trình nhúng tin trên ảnh JPEG

Theo [7,18], qui trình nhúng tin trên ảnh JPEG được mô tả như Hình 1:

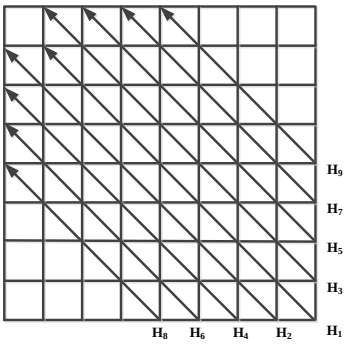



**Hình 1.** *Quy trình nhúng tin trên ảnh JPEG.*

Các lược đồ nhúng tin chỉ khác nhau chủ yếu ở phương pháp nhúng dãy bít $W$ vào các khối DCTLT sao cho đạt được khả năng nhúng cao và cải thiện chất lượng ảnh. Những phần tiếp theo sẽ trình bày một số lược đồ nhúng trên ảnh JPEG.

### *II.2.* Lược đồ Iwata

Lược đồ Iwata [7] nhúng 9 bít dữ liệu trên 9 đường chéo của một khối DCTLT như Hình 2:

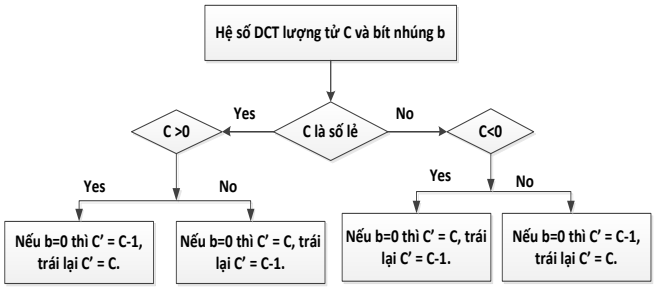



**Hình 2**. *Các đường chéo của khối DCTLT.*

Gọi $D_i = (d_1, d_2, \dots, d_{k_i})$ với $1 \le i \le 9$ là dãy hệ số trên đường chéo $H_i$ theo thứ tự từ dưới lên trên như Hình 2, $t_i$ là số phần tử 0 liên tiếp tối đa của $D_i$tính từ hệ số $d_1$ $(0 \le t_i \le k_i)$. Iwata nhúng một bít $b_i$ vào $D_i$ bằng cách thay đổi tối đa một phần tử của $D_i$ để $b_i$ có cùng tính chẵn lẻ với $t_i$. Như vậy, lược đồ [7] nhúng được chín bít và thay đổi tối đa chín phần tử của khối DCTLT. Trong mục 3.2 chúng tôi sẽ đề xuất các lược đồ có khả năng nhúng cao hơn mà số phần tử cần biến đổi cũng ít hơn lược đồ Iwata.

### II.3 Lược đồ Liu

Lược đồ Liu [12] nhúng một bít b trên mỗi hệ số DCTLT C có giá trị khác 0 và khác $\pm 1$. Nội dung thuật toán được mô tả như Hình 3.




***Hình 3**: Thuật toán nhúng tin Liu.*

## III. CÁC LƯỢC ĐỒ NHÚNG TIN ĐỀ XUẤT

### III.1. Lược đồ nhúng r bít trên một dãy $2^r$-1 phần tử nguyên

Để tiện cho việc trình bày, chúng tôi sử dụng ký hiệu $\oplus$ là phép *xor* trên từng cặp bít tương ứng của hai số nguyên không âm $D_i$, $i = 1 \dots n$ và kí hiệu:

$$\sum_{i=1\dots n}^{\oplus} D_i = D_1 \oplus D_2 \oplus \dots \oplus D_n$$

#### *III.1.1. Thuật toán nhúng*

Cho một dãy số nguyên $D = (d_1, d_2, \dots, d_R)$ và dãy bít cần nhúng $b = (b_1, b_2, \dots, b_r)$, trong đó $R = 2^r - 1$. Thuật toán nhúng $r$ bít của $b$ vào dãy nguyên $D$ để nhận được dãy $D'$ chỉ khác $D$ tối đa một phần tử như sau:

*Bước 1.* Tính:

$$s = \sum_{i=1\dots R}^{\oplus} mod(|d_i|, 2) \times i \qquad (1)$$

*Bước 2.* So sánh $s$ và $b$:

Nếu $s = b$ thì $D' = D$ và kết thúc thuật toán, ngược lại thì chuyển sang Bước 3.

*Bước 3.* Tính:

$$t = s \oplus b$$

Khi đó $t \in \{1, 2, \dots, R\}$. Tăng $d_t$ lên một đơn vị:

$$d_t = d_t + 1$$

Đặt $D' = D$ và kết thúc thuật toán.

#### *III.1.2. Thuật toán trích tin*

Giả sử có dãy $D' = (d'_1, d'_2, \dots, d'_R)$. Khi đó, dãy bít $b$ được trích từ $D'$ theo công thức sau:

$$b = \sum_{i=1\dots R}^{\oplus} mod(|d'_i|, 2) \times i \qquad (2)$$

*Chứng minh tính đúng đắn*

Đặt:

$$s' = \sum_{i=1\dots R}^{\oplus} mod(|d'_i|, 2) \times i$$

Ta cần chứng minh $s' = b$. Thật vậy:

Trường hợp 1: nếu $D'$ nhận được từ $D$ ở Bước 2 thì (2) hiển nhiên là đúng.

Trường hợp 2: nếu $D'$ nhận được từ $D$ ở Bước 3 thì:

$$d'_i = \begin{cases} d_i, i \neq t \\ d_i + 1, i = t \end{cases}$$

Nếu $d_t$ chẵn, tức là $mod(|d_t|, 2) = 0$ thì $mod(|d'_t|, 2) = 1$, vậy ta có:

$$s' = s \oplus t$$

Do $t = s \oplus b$ suy ra:

$$s' = s \oplus (s \oplus b) = (s \oplus s) \oplus b = b$$

- Nếu $d_t$ lẻ hay $mod(|d_t|, 2) = 1$ thì $mod(|d'_t|, 2) = 0$. Khi đó, bằng cách lập luận tương tự suy ra:

$$s = s' \oplus t$$

Thêm $t$ vào cả hai vế ta được:

$$s \oplus t = s' \oplus t \oplus t$$

Từ đó suy ra:

$$s' = s \oplus t = s \oplus (s \oplus b) = b$$

Vậy $s' = b$.

Như vậy công thức (2) đúng trong mọi trường hợp. Đó là điều cần chứng minh.

### III.2. Các lược đồ nhúng tin trên ảnh JPEG

Phần này đề xuất các lược đồ nhúng tin trên một khối DCTLT của ảnh, sử dụng lược đồ nhúng $r$ bít như ở Mục 3.1. Các lược đồ dưới đây có thuật toán trích tin giống nhau chỉ khác nhau ở thuật toán nhúng tin như sau:

#### *III.2.1. Lược đồ T2*

Lược đồ *T2* nhúng được 10 bít $W = (w_1, w_2, \dots, w_{10})$ trên một khối DCTLT $D$ kích thước 8×8, để nhận được khối $D'$ chỉ khác $D$ tối đa hai giá trị như sau:

*Bước 1.* Biến đổi ma trận $D$ thành một dãy $\mathfrak{D}$ gồm 64 phần tử theo đường zigzag:

$$\mathfrak{D} = (d_1, d_2, \dots, d_{64}).$$

*Bước 2.* Từ dãy $\mathfrak{D}$ ta lập hai dãy con $U = (d_2, d_3, \dots, d_{32})$ và $V = (d_{33}, d_{34}, \dots, d_{63})$.

*Bước 3.* Nhúng dãy $(w_1, w_2, \dots, w_5)$ vào $U$ và dãy $(w_6, w_7, \dots, w_{10})$ vào $V$ theo thuật toán đã trình bày ở Mục 3.1 để nhận được $U', V'$. Ghép $(d_1, U', V', d_{64})$ để có dãy $\mathfrak{D}'$.

*Bước 4.* Biến đổi ngược dãy $\mathfrak{D}'$ để được ma trận $D'$chứa dấu thủy vân *W*.

**III.*2.2 Lược đồ T3***

Thuật toán nhúng của Lược đồ T3 được thực hiện tương tự như ở Lược đồ T2 chỉ khác ở Bước 2 và Bước 3, nhúng được 12 bít thủy vân $W=(w_1, w_2, \dots, w_{12})$ trên một khối DCTLT mà *D* và $D'$chỉ khác nhau tối đa 3 giá trị như sau:

*Bước 2.* Từ dãy *D* tạo ra ba dãy con:

$U=(d_2, d_3, \dots, d_{16}), V = (d_{17}, d_{18}, \dots, d_{31})$ và

$T = (d_{32}, d_{33}, \dots, d_{48})$.

*Bước 3.* Mỗi dãy *U*, *V*, *T* được nhúng bốn bít của dấu thủy vân *W* theo thuật toán ở Mục 3.1 để nhận được các dãy $U', V', T'$ tương ứng.

**Nhận xét:** Nếu chọn các dãy con *U*, *V*, *T* như sau:

$U=(d_2, d_3, \dots, d_{32})$, $V = (d_{33}, d_{34}, \dots, d_{47})$ và $T = (d_{48}, d_{49}, \dots, d_{62})$

Tiến hành nhúng 5 bít trên dãy *U*, các dãy *V* và *T* mỗi dãy nhúng 4 bít thì Lược đồ T3 nhúng được 13 bít mà *D* và $D'$chỉ khác nhau tối đa 3 giá trị.

***III.2.3 Lược đồ T4***

Lược đồ này nhúng được 16 bít thủy vân $W=(w_1, w_2, \dots, w_{16})$ trên một khối DCTLT mà chỉ thay đổi tối đa bốn giá trị của khối. Thuật toán nhúng được thực hiện tương tự như Lược đồ T2 chỉ khác ở Bước 2 và Bước 3 như sau:

*Bước 2.* Từ dãy *D* tạo ra bốn dãy con:

$U=(d_2, d_3, \dots, d_{16}), V = (d_{17}, d_{18}, .., d_{31})$

$T = (d_{32}, d_{33}, \dots, d_{48})$ và $S= (d_{49}, d_{50}, \dots, d_{63})$

*Bước 3.* Thực hiện nhúng vào mỗi dãy *U*, *V*, *T* và *S* bốn bít thủy vân của *W* theo Mục 3.1 để nhận được $U', V', T'$và $S'$. Ghép $(d_1, U', V', T', S', d_{64})$ để có dãy $\mathfrak{D}'$.

## III. Đề xuất lược đồ thủy vân dễ vỡ khóa công trên ảnh JPEG

Lược đồ thủy vân đề xuất dựa theo [8,10] có sử dụng hệ mật mã RSA, hàm băm *SHA1* và thuật toán nhúng T4. Lược đồ gồm thuật toán nhúng dấu thủy vân và thuật toán xác thực tính toàn vẹn như Hình 4.

Để tiện cho việc trình bày chúng tôi ký hiệu:

$P_1(I)$: 16 khối DCTLT của thành phần *Y* của ảnh JPEG *I*.

$P_2(I)$: các khối DCTLT còn lại của thành phần *Y* và toàn bộ các khối DCTLT của $C_b$ và $C_r$ của ảnh *I*.

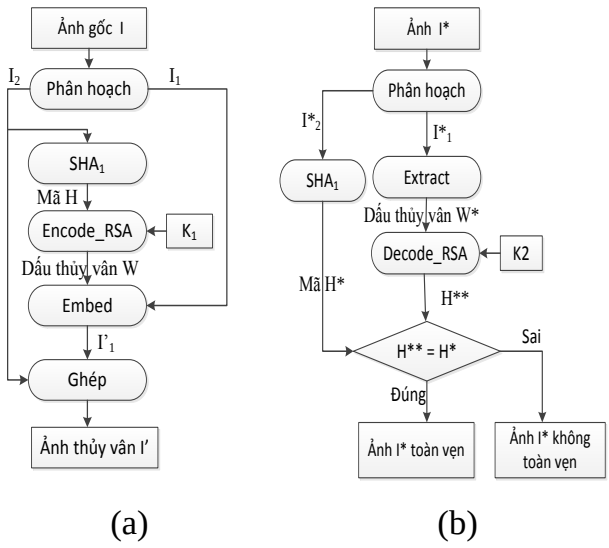
$SHA1(I_2)$: bản mã hàm băm *SHA1* của $I_2$.

$Encode_RSA(K_1, H)$: bản mã RSA của *H* theo khóa bí mật $K_1$.

$Decode_RSA(W, K_2)$: bản giải mã RSA của *W* theo khóa công khai $K_2$

$Embed(W, I_1)$: khối ảnh nhận được sau khi nhúng *W* vào $I_1$ theo lược đồ nhúng T4 (Mục III.2).

$Extract(I_1^*)$: dãy bít trích từ $I_1^*$ theo thuật toán trích (Mục 3.1) tương ứng với lược đồ nhúng T4.




**Hình 4.** *Mô hình thủy vân đề xuất, (a) Quá trình nhúng dấu thủy vân, (b) Quá trình xác thực.*

### IV.1. Thuật toán nhúng thủy vân

Đầu vào là ảnh JPEG *I*, đầu ra là ảnh thủy vân $I'$ ở định dạng JPEG. Các bước chính của thuật toán như sau:

*Bước 1.* Phân hoạch ảnh *I*:

$$I_1 = P_1(I), \qquad I_2 = P_2(I)$$

*Bước 2.* Tạo dấu thủy vân W từ $I_2$:

$$H = SHA1(I_2) \quad (3)$$

$$W = Encode_RSA(K_1, H) \quad (4)$$

*Bước 3.* Nhúng dấu thủy vân W vào $I_1$:

$$I_1' = Embed(W, I_1) \quad (5)$$

*Bước 4.* Tạo ảnh thủy vân $I'$: Ghép $I_1'$ và $I_2$ để nhận được ảnh thủy vân $I'$.

**Nhận xét:** Theo Bước 4, ta có:

$$I_2^* = I_2' \quad (6)$$

**IV.2. Thuật toán xác thực tính toàn vẹn của ảnh**

Trong quá trình truyền tải, ảnh thủy vân $I'$ có thể bị tấn công thành ảnh $I^*$. Thuật toán dưới đây có thể kiểm tra tính toàn vẹn của vùng ảnh $I_2'$ (giả định $I_2'$ chứa dữ liệu mật cần được bảo vệ):

*Bước 1.* Phân hoạch ảnh $I^*$:

$$I_1^* = P_1(I^*), I_2^* = P_2(I^*)$$

*Bước 2.* Tính $H^*$ và $H^{**}$:

$$H^* = SHA1(I_2^*) \quad (7)$$

$$W^* = Extract(I_1^*) \quad (8)$$

$$H^{**} = Decode_RSA(K_2, W^*) \quad (9)$$

*Bước 3.* Kiểm tra tính toàn vẹn:

Nếu $H^{**}$ =$H^*$ thì kết luận $I_2^* = I_2'$.

Nếu $H^{**} \neq H^*$ thì kết luận $I_2^* \neq I_2'$.

**IV.3. Phân tích tính dễ vỡ của lược đồ đề xuất**

Ta chứng minh nếu $I_2'$ bị tấn công (tức là$I_2^* \neq I_2'$) thì thuật toán xác thực sẽ phát hiện được.

**Trường hợp 1:** Nếu ảnh $I'$chỉ bị tấn công ở vùng $I_2'$, tức là:

$$I_1^* = I_1' \text{ và } I_2^* \neq I_2'$$

Nên từ (3), (6) và (7) suy ra:

$$H^* \neq H \quad (10)$$

Mặt khác do $I_1^* = I_1'$ nên từ (5) và (8) suy ra:

$$W^* = W$$

Nên từ (4) và (9) ta có:

$$H^{**} = H$$

Do đó, từ (10) suy ra $H^* \neq H^{**}$.

Vậy thuật toán xác thực phát hiện được sự thay đổi của ảnh $I_2'$.

**Trường hợp 2:** Ảnh $I'$ bị tấn công ở cả hai vùng $I_2'$ và $I_1'$ nghĩa là:

$$I_1^* \neq I_1' \text{ và } I_2^* \neq I_2' \quad (11)$$

Để không bị phát hiện sự biến đổi của $I_2'$, người thám tin cần thay đổi $I_1'$ và $I_2'$ (tức là xác định $I_1^*$ và $I_2^*$) sao cho:

$$H^* = H^{**} \quad (12)$$

Ta sẽ chỉ ra điều này là không thể. Thật vậy:

Người thám tin có thể tấn công theo các cách sau:

***Cách 1:*** Biến đổi đồng thời $I_1'$ và $I_2'$, tức là xác định đồng thời $I_1^*$ và $I_2^*$ thỏa mãn (11) và (12). Trên thực tế, việc này không làm được vì số phương án cần duyệt quá lớn.

***Cách 2:*** Tấn công $I_2'$ trước sau đó biến đổi $I_1'$, nghĩa là biết trước $I_2^*$ rồi xác định $I_1^*$ theo $I_2^*$.

Từ (9) suy ra để $H^* = H^{**}$thì $W^*$ cần thỏa mãn:

$$W^* = Encode_RSA(K_1, H^*) \quad (13)$$

Trong đó, $H^*$ xác định theo (7) (do $I_2^*$ đã biết). Như vậy, người thám tin có thể đạt được (12) bằng cách tính $W^*$ theo (13), rồi xác định $I_1^*$ theo công thức:

$$I_1^* = Embed(W^*, I_1') \quad (14)$$

Tuy nhiên, do không biết $K_1$($K_1$là khóa bí mật) nên người thám tin không thể thực hiện được điều này.

***Cách 3:*** Tấn công $I_1'$ trước, sau đó biến đổi $I_2'$, tức là biết trước $I_1^*$ rồi xác định $I_2^*$ theo $I_1^*$.

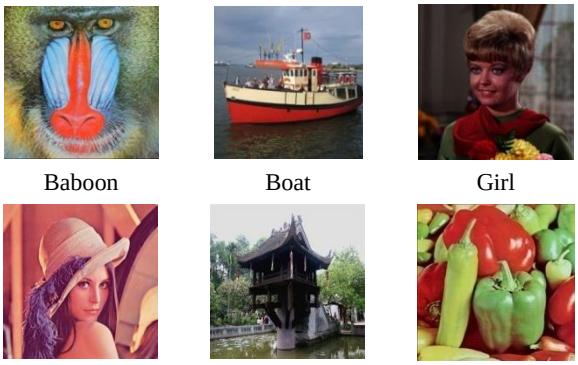
Từ (7) có thể thấy để $H^* = H^{**}$ thì $I_2^*$ cần thỏa mãn:

*SHA1*($I_2^*$)=$H^{**}$(15). Trong đó $H^{**}$ tính theo (8) và (9) (do $I_1^*$ đã biết). Như vậy, thám tin có thể đạt được (12) bằng cách xác định $I_2^*$ từ $H^{**}$ theo (15). Tuy nhiên do *SHA1* là hàm một chiều, nên điều này không thể thực hiện được.

Vậy, tính dễ vỡ của thuật toán xác thực được chứng minh.

## V. SO SÁNH, ĐÁNH GIÁ KHẢ NĂNG NHÚNG TIN VÀ CHẤT LƯỢNG ẢNH GIỮA CÁC LƯỢC ĐỒ

Chúng tôi sử dụng ngôn ngữ VC++ để viết phần mềm khảo sát về chất lượng ảnh của các lược đồ [6,7,12] và lược đồ đề xuất. Các ảnh JPEG thử nghiệm như Hình 5.




***Hình 5.*** *Các ảnh thử nghiệm kích thước 256×256*

### V.1. Đánh giá về khả năng nhúng

Đối với lược đồ Iwata[7], mỗi khối DCTLT nhúng được 9 bít thay đổi tối đa 9 phần tử.

Các lược đồ [6] và [12] thực hiện nhúng trên các hệ số khác 0 và khác $\pm 1$. Qua thống kê với hơn 100 ảnh kích thước 256×256 đơn vị cho thấy, trung bình mỗi khối DCTLT nhúng được 15.6 bít và thay đổi trung bình từ 7 đến 8 phần tử.

Đối với các lược đồ đề xuất, lược đồ T4 nhúng được 16 bít trên mỗi khối DCTLT thay đổi tối đa 4 phần tử. Nếu thực hiện nhúng 2 bít trên mỗi phân đoạn 3 phần tử của dãy thu được từ việc quét zigzag khối DCTLT thì lược đồ đề xuất nhúng tối đa được 42 bít và thay đổi tối đa 21 phần tử. Như vậy lược đồ đề xuất có khả năng nhúng cao hơn các lược đồ [6], [7] và [12].

### V.2. Đánh giá về chất lượng ảnh

#### *V.2.1. Khảo sát sự thay đổi của khối DCTLT*

Để đánh giá độ thay đổi của khối DCTLT chúng tôi dùng công thức:

$$\sum_{t=1}^{k} |A_t - A'_t| \qquad (16)$$

Trong đó, k là tổng số khối DCTLT thu được ở bước lượng tử của ảnh JPEG. $A_t$ là một khối DCTLT, $A'_t$ là khối DCTLT sau khi đã nhúng tin.

$$|A_t - A'_t| = \sum_{i=1}^{8}\sum_{j=1}^{8} |A_t(i,j) - A'_t(i,j)| \qquad (17)$$

***Bảng 1.*** *Sự thay đổi của khối DCTLT.*

| Ảnh | Iwata[7] | J_Steg[6] | Liu[12] | T2 |
|---|---|---|---|---|
| Baboon | 3331 | 4120 | 4099 | **1096** |
| Boat | 2780 | 3194 | 2954 | **611** |
| Girl | 2224 | 2544 | 2536 | **359** |
| Lena | 2214 | 2879 | 2689 | **455** |
| Pagoda | 2925 | 3279 | 3021 | **517** |
| Pepper | 2358 | 3585 | 3334 | **695** |

Bảng 1 cho thấy khối DCTLT của lược đồ nhúng đề xuất thay đổi ít hơn so với các lược đồ còn lại.

#### *V.2.2. Khảo sát hệ số PSNR*

Người ta thường dùng hệ số $PSNR$ để đánh giá chất lượng của ảnh thủy vân $I'$ so với ảnh gốc $I$ kích thước $k \times l$. Theo [5], hệ số $PSNR$ được tính theo công thức sau:

$$PSNR = 20.log_{10}\frac{MAX}{\sqrt{MSE}}$$

Trong đó, $MAX$ là giá trị cực đại của điểm ảnh và $MSE$ được xác định theo công thức:

$$MSE = \frac{1}{k \times l}\sum_{i=1}^{k}\sum_{j=1}^{l}|I(i,j) - I'(i,j)|$$

Bảng 2 khảo sát hệ số *PSNR* của lược đồ đề xuất (T2) và với các lược đồ [6], [7] và [12]. Lược đồ nào có PSNR cao hơn được đánh giá là cho chất lượng ảnh tốt hơn. Các số liệu ở Bảng 2 sẽ cho thấy PSNR của lược đồ đề xuất cao hơn các lược đồ [6], [7] và [12] chứng tỏ lược đồ đề xuất có chất lượng ảnh tốt hơn.

***Bảng 2.*** *PSNR của lược đồ[6,7,12] và lược đồ đề xuất*

| Ảnh | Iwata[7] | J_Steg[6] | Liu[12] | T2 |
|---|---|---|---|---|
| Baboon | 36.9725 | 42.6061 | 43.3055 | **50.5264** |
| Boat | 41.3311 | 46.9287 | 47.4533 | **50.4830** |
| Girl | 39.2523 | 40.0313 | 40.3198 | **43.3800** |
| Lena | 37.5772 | 41.7128 | 41.4035 | **43.1859** |
| Pagoda | 37.0597 | 45.6192 | 45.9878 | **53.5600** |
| Pepper | 40.0331 | 41.9051 | 42.3796 | **46.7503** |

Như vậy, lược đồ đề xuất không chỉ có khả năng nhúng tốt hơn mà còn nâng cao được chất lượng ảnh.

## VI. KẾT LUẬN

Bài báo đã đề xuất các lược đồ nhúng tin mà kỹ thuật nhúng được thực hiện trên các khối DCT lượng tử của ảnh JPEG. Mỗi khối DCT lượng tử được phân hoạch thành các đoạn con có độ dài $2^r-1$ và tiến hành nhúng $r$ bít trên mỗi dãy con đó. Các lược đồ nhúng tin đề xuất không chỉ có khả năng nhúng cao hơn từ 1.2 đến 2.4 lần mà còn cho chất lượng ảnh tốt hơn các lược đồ [6], [7] và [12]. Thực nghiệm cũng cho thấy, các lược đồ nhúng đề xuất có khối DCT lượng tử ít thay đổi hơn, hệ số PSNR cao hơn chứng tỏ chất lượng ảnh của các lược đồ nhúng đề xuất là tốt hơn. Lược đồ thủy vân khóa công khai được xây dựng dựa trên lược đồ nhúng đề xuất có khả năng xác thực tính toàn vẹn của vùng ảnh chứa dữ liệu mật.

## TÀI LIỆU THAM KHẢO


[1] A. ALMOHAMMAD, G. GHINEA AND R.M.HIERONS *"JPEG steganography: a performance evaluation of quantization tables*", Advanced Information Networking and Applications conference, IEEE, pp471–478, 2009.

[2] C.C.CHANG, C.C. LIN, C.S.TSENG AND W.L.TAI, "*Reversible hiding in DCT-based compressed images"*, Information Sciences, Vol.177, pp.2768-2786, 2007.

[3] C.C.CHANG, C.S.TSENG AND C.C.LIN. *"Hiding Data in Binary Images"*, ISPEC 2005, LNCS 3439, pp.338-349, 2005.

[4] Y.CHEN, H.PAN, AND C.S. TSENG, *"A secure Data Hiding Scheme for Two color images"*, IEEE Symposium on Computer and Communication, 2000.

[5] C.FEI, R.KWONG AND D.KUNDAR, "*Secure Semi-Fragile Watermarking for Image Authentication*", IEEE, 2009.

[6] J_Steg_http://www.ussrback.com/crypto/steganography/DOS/jsteg.txt , 2008.

[7] M.IWATA, K.MIYAKE AND A.SHIOZAKI, "*Digital Steganography Utilizing features of JPEG images*", IEICE, 2004.

[8] H.Y.KIM AND R.L.DE. QUEIROZ, "*A New Public Key Authentication Watermarking for Binary Document Images*", IEEE, 2004.

[9] H.Y. KIM, R.L.DE. QUEIROZ, "*Alteration – Locating Authentication Watermarking for Binary and Halftone Images*", IEEE Transactions on Signal Processing, 2004.

[10] H.Y. KIM, "*A New Public Key Authentication Watermarking for Binary Document Images resistant to parity* attacks", IEEE, 2005.

[11] C.C. LIN AND F.F. SHIU, *"DCT-Base Reversible Data Hiding Scheme"*, Journal of software, 2010.

[12] C.L. LIU, C.R.LIAO "*High-performance JPEG ste-ganography using complementary embedding strategy*", Pattern recogn 41, pp.2945–2955, 2009.

[13] B.C.MOHAN AND S.KUMAR, *"A Robust Digital Image Watermarking Scheme Using Singular Value Decomposition"*, Journal of Multimedia 3(1), pp.7-15, 2008.

[14] R.MUNIR, B.RIYANTO, S.SUTIKNO AND W.P.AGUNG, "*A public key watermarking method for still image based on random permutation*",in Proc Iternational Joint Conference in Engineering, 2008.

[15] N.PROVOS, "*Defening against statical steganalysis*". In 10th USENIX Security Symposium, 2001 (323336).

[16] S. SACHDEVA AND A. SHARMA AND GILL "*Colour Image Steganography Using Modified JPEG Quantization Technique*", Int J Latest Res Sci tech(1), pp.1-5, 2012.

[17] K.WANG, Y.J.HE AND Z.M. LU, "*A high capacity lossless data hiding scheme for jpeg images*", Journal of Systems and Software (86), 2013.

[18] C.C. WANG AND Y.C.HSU, "*New watermarking algorithm with data authentication and reduction for JPEG image*", journal of electronic image, 2008.

[19] M.Y.WU AND J.H.LEE, "*A Novel Data Embedding Method for Two-color Facsimile Images*", in Proc. Int. Symp, on Multimedia Information Processing, Chung-Li, Taiwan,R.O.C, 1998.

[20] ĐỖ VĂN TUẤN, TRẦN ĐĂNG HIÊN, CAO THỊ LUYÊN VÀ PHẠM VĂN ẤT, *"Một thuật toán thủy vân bền vững khóa công khai cho ảnh màu dựa trên sự hoán vị ngẫu nhiên trong các tập con"*, Tạp chí CNTT&TT, Tập V-1, Số 9 (29), tháng 6 -2013.

**Ngày nhận bài**: 22/04/2015

**SƠ LƯỢC VỀ TÁC GIẢ**

**CAO THỊ LUYÊN**

Sinh ngày 28/4/1979 tại Hưng Yên.

Tốt nghiệp ĐH năm 2001 và Thạc sĩ năm 2005 tại Trường ĐH Khoa học Tự nhiên, ĐH Quốc Gia Hà Nội.

Hiện giảng dạy tại Khoa CNTT – Trường ĐH Giao thông Vận tải.

Lĩnh vực nghiên cứu: Giấu tin, thủy vân số, an toàn thông tin.

Email: luyenct28042000@yahoo.com

**TIÊU THỊ NGỌC DUNG**

Sinh ngày 25/11/1980 tại Thanh Hóa.

Tốt nghiệp ĐH năm 2003 tại Trường ĐH Bách khoa Hà Nội; Thạc sĩ năm 2006 tại Trường ĐH Tổng hợp Hàn Quốc.

Hiện giảng dạy tại Khoa CNTT – Trường ĐH Giao thông Vận.

Lĩnh vực nghiên cứu: an toàn thông tin, xử lý ảnh, nhận dạng

Email: tieungocdung@yahoo.com

**ĐỖ VĂN TUẤN**

Sinh ngày 9/6/1975 tại Hải Dương.

Tốt nghiệp ĐH tại Học viện Kỹ thuật Quân sự năm 2002 ; Thạc sĩ năm 2007 tại Trường ĐH Công nghệ – ĐH Quốc gia Hà Nội.

Hiện giảng dạy tại Khoa CNTT – Trường Cao đẳng Thương mại và Du lịch Hà Nội.

Email: dvtuanest@gmail.com

**PHẠM VĂN ẤT**

Sinh ngày 12/6/1945 tại Hà Nội.

Tốt nghiệp ĐH năm 1967 và tiến sĩ năm 1980 tại Trường ĐH Tổng hợp Hà Nội. Nhận học hàm Phó Giáo sư Năm 1984.

Hiện giảng dạy tại Khoa CNTT – Trường ĐHGTVT.

Lĩnh vực quan tâm: Lý thuyết ma trận, xử lý ảnh, an toàn thông tin, phân tích dữ liệu.

Email: phamvanat83@vnn.vn