Kỷ yếu Hội nghị Quốc gia lần thứ X về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR), Đà Nẵng, ngày 17-18/08/2017
DOI: 10.15625/vap.2017.00013

ĐẢM BẢO SỰ TOÀN VẸN CHO CƠ SỞ DỮ LIỆU QUAN HỆ CÓ CÁC THUỘC TÍNH VĂN BẢN BẰNG LƯỢC ĐỒ THỦY VÂN

Lưu Thị Bích Hương [1], Bùi Thế Hồng [2]
[1]Khoa Công nghệ thông tin, Trường ĐHSP Hà Nội 2
[2]Khoa Công nghệ thông tin, Trường ĐHSPKT Hưng Yên
luuthibichhuong@hpu2.edu.vn, hongbuithe@gmail.com

TÓM TẮT: *Bài báo đưa ra một lược đồ thủy vân cải tiến đảm bảo sự toàn vẹn cho cơ sở dữ liệu quan hệ có các thuộc tính văn bản. Mọi thay đổi trên quan hệ bất kỳ của lược đồ quan hệ đều ảnh hưởng đến ký tự thủy vân được sinh ra, điều này có nghĩa có thể phát hiện được sự thay đổi trên cả các quan hệ không nhúng thủy vân.*

Từ khóa: *thủy vân; thuộc tính văn bản; toàn vẹn.*

I. GIỚI THIỆU

Gần đây, đã có một số lược đồ thủy vân bền vững được công bố nhằm bảo vệ bản quyền cho cơ sở dữ liệu quan hệ. Tuy nhiên, do nhu cầu của chủ sở hữu không phải lúc nào các dữ liệu cũng cần phải chứng thực bản quyền mà có thể chỉ cần xác minh dữ liệu vẫn còn toàn vẹn là đủ [2]. Nhưng cho đến nay chưa có nhiều công trình nghiên cứu về khía cạnh này.

Bedi R. và cộng sự [1] đã đưa ra một lược đồ thủy vân cho dữ liệu kiểu văn bản. Tư tưởng của lược đồ này là sử dụng một khóa thủy vân được xây dựng từ các bộ trong quan hệ và nhúng khóa này vào các thuộc tính có ảnh hưởng không lớn đến giá trị sử dụng của dữ liệu. Tuy nhiên, nếu như kẻ tấn công biết được thuật toán tìm khoá thủy vân thì có thể dễ dàng tìm được khóa thủy vân cũng như ma trận thủy vân hay lược đồ không còn an toàn.

Dựa vào tư tưởng xây dựng thủy vân từ các bộ dữ liệu trong quan hệ của Bedi R. và cộng sự, chúng tôi đề xuất một lược đồ thủy vân cơ sở dữ liệu quan hệ với dữ liệu văn bản. Trong lược đồ này, việc tính ma trận thủy vân dựa vào tư tưởng của [1]. Điểm khác biệt giữa hai lược đồ là lược đồ đề xuất của chúng tôi đưa vào tham số khóa thủy vân và cách xác định các bộ được nhúng. Lược đồ đề xuất này, an toàn hơn lược đồ thủy vân của Bedi R. và cộng sự, ngoài ra còn có thể khoanh vùng được các giả mạo nếu có. Lược đồ thủy vân này đã được chúng tôi công bố trong bài báo [3].

Điểm hạn chế của lược đồ thủy vân trong bài báo [3] là chỉ có thể thủy vân trên một quan hệ. Khắc phục nhược điểm này chúng tôi đề xuất một lược đồ thủy vân cơ sở dữ liệu quan hệ. Trong lược đồ này, mọi thay đổi trên quan hệ bất kỳ của lược đồ thủy vân đều ảnh hưởng đến ký tự thủy vân được sinh ra điều này có nghĩa có thể phát hiện được sự thay đổi trên cả các quan hệ không nhúng thủy vân.

Trong phần tiếp theo chúng tôi trình bày một số định nghĩa. Phần III là lược đồ thủy vân cải tiến, trong phần này sẽ trình bày tư tưởng của lược đồ thủy vân cải tiến, thuật toán nhúng thủy vân, thuật toán phát hiện thủy vân và xác minh sự toàn vẹn của cơ sở dữ liệu quan hệ. Phần IV là chứng minh tính đúng đắn của lược đồ thủy vân cải tiến. Phần cuối cùng là kết luận.

II. MỘT SỐ ĐỊNH NGHĨA

Trong một lược đồ quan hệ, có một số thuộc tính có ý nghĩa quan trọng và một số thuộc tính khác có ảnh hưởng không lớn đến giá trị sử dụng cũng như ý nghĩa thực tế. Sau đây chúng tôi sẽ đưa ra hai định nghĩa để làm rõ hơn về các loại thuộc tính này [3].

Định nghĩa 1: Thuộc tính có tác động cao

Một thuộc tính được gọi là thuộc tính tác động cao hoặc thuộc tính có tác động cao hoặc thuộc tính có ảnh hưởng cao hay còn gọi là thuộc tính có ý nghĩa quan trọng khi bất kỳ một thay đổi nào đối với giá trị của thuộc tính thì giá trị đặc trưng của chúng cũng bị thay đổi theo.

Định nghĩa 2: Thuộc tính có tác động thấp

Một thuộc tính được gọi là thuộc tính tác động thấp hoặc thuộc tính có tác động thấp hoặc thuộc tính có ảnh hưởng thấp hay còn gọi là thuộc tính không có ý nghĩa quan trọng nếu có một thay đổi nhỏ đối với giá trị của thuộc tính sẽ có ảnh hưởng không lớn đến giá trị sử dụng cũng như giá trị ý nghĩa thực tế của thuộc tính đó.

Ví dụ, trong một lược đồ quan hệ nhân sự thì các thuộc tính ghi số chứng minh thư, họ tên, năm sinh, giới tính, ngày tăng lương là những thuộc tính quan trọng, có ảnh hưởng lớn đối với đương sự. Các thuộc tính như quê quán, nơi sinh có ảnh hưởng không lớn đối với đương sự.

III. LƯỢC ĐỒ THỦY VÂN

Ý tưởng của lược đồ thủy vân cải tiến là tính giá trị trên tất cả các thuộc tính (bao gồm các thuộc tính văn bản và thuộc tính số) sau đó lấy giá trị này kết hợp với khóa thủy vân cho trước để xác định thủy vân rồi nhúng vào các thuộc tính văn bản tác động thấp.

Lược đồ thủy vân cải tiến được thiết kế để đảm bảo sự toàn vẹn cho các quan hệ thuộc lược đồ quan hệ có dạng:

$$R(H_1, H_2, ..., H_m, L_1, L_2, ..., L_n)$$

Trong đó, m thuộc tính $H_1, H_2, \ldots, H_m$ là thuộc tính tác động cao còn $L_1, L_2, ..., L_n$ là n thuộc tính tác động thấp. Không mất tính tổng quát, giả sử trong lược đồ quan hệ trên mỗi quan hệ có ω bộ. K là khóa thủy vân.

Bảng 1. *Các ký hiệu được sử dụng trong lược đồ thủy vân*

Ký hiệu	**Ý nghĩa**
R	Lược đồ quan hệ
r	Quan hệ r thuộc lược đồ R
r_i	Bộ thứ i của quan hệ r
$r_i.B_j$	Giá trị của thuộc tính B_j thuộc bộ r_i
ω	Số bộ trong quan hệ r
K	Khóa nhúng thủy vân
n	Số thuộc tính kiểu văn bản có tác động thấp trong quan hệ
m	Số thuộc tính kiểu văn bản có tác động cao trong quan hệ
H_i	Thuộc tính kiểu văn bản có tác động cao thứ i trong quan hệ hoặc trong nhóm
L_i	Thuộc tính kiểu văn bản có tác động thấp thứ i trong quan hệ hoặc trong nhóm

Lược đồ thủy vân dùng để đảm bảo sự toàn vẹn cho các cơ sở dữ liệu quan hệ có các thuộc tính kiểu văn bản được thực hiện dựa vào hai thuật toán:

- Thuật toán nhúng thủy vân.

- Thuật toán phát hiện thủy vân và xác minh sự toàn vẹn.

A. Thuật toán nhúng thủy vân

Ý tưởng của thuật toán nhúng thủy vân cải tiến bao gồm các bước sau:

- Sinh thủy vân từ các bộ của lược đồ cơ sở dữ liệu quan hệ R

- Với mỗi quan hệ $r \in R$. Chia ω bộ của r thành g nhóm

- Mỗi một nhóm G_i được sinh ra xác định 1 ký tự thủy vân nhúng vào thuộc tính tác động thấp tại vị trí bất kì trong nhóm.

1) Sinh thủy vân của các tác giả [1], [3]

Đầu tiên sẽ tìm thủy vân từ các bộ của quan hệ bằng cách dựa vào khóa thủy vân, các thuộc tính tác động cao và các thuộc tính tác động thấp.

Đối với m thuộc tính tác động cao, tính tổng mã Unicode của tất cả các nguyên âm, phụ âm và ký tự đặc biệt trong toàn bộ các thuộc tính và ký hiệu cụ thể là:

$$V^H = \sum_{i\,j}\{\sum \text{Unicode của các nguyên âm thuộc } r_i.H_j,\ 1 \le i \le \omega;\ 1 \le j \le m\}$$

$$C^H = \sum_{i\,j}\{\sum \text{Unicode của các phụ âm thuộc } r_i.H_j,\ 1 \le i \le \omega;1 \le j \le m\}$$

$$P^H = \sum_{i\,j}\{\sum \text{Unicode của các ký tự đặc biệt của } r_i.H_j,\ 1 \le i \le \omega;1 \le j \le m\}$$

Đối với n thuộc tính tác động thấp, tính tổng mã Unicode của các nguyên âm, phụ âm, ký tự đặc biệt và tổng mã Unicode của tất cả các ký tự theo từng thuộc tính.

$$V^L_j = \sum_i \{\sum \text{Unicode của các nguyên âm của } r_i.L_j,\ 1 \le i \le \omega\}$$

$$C^L_j = \sum_i \{\sum \text{Unicode của các phụ âm của } r_i.L_j,\ 1 \le i \le \omega\}$$

$$P^L_j = \sum_i \{\sum \text{Unicode của các ký tự đặc biệt của } r_i.L_j,\ 1 \le i \le \omega\}$$

$$A^L_j = \sum_i \{\sum \text{Unicode của tất cả ký tự của } r_i.L_j,\ 1 \le i \le \omega\}$$

- Xây dựng một ma trận đặt tên là *D* bao gồm 4 hàng và *n* cột với các thành phần là D_{i1}, D_{i2}, D_{i3}, D_{i4} (với i = 1, 2, …, n) được tính như sau:

$$D_{i1} = V^H + V^L_i,\ D_{i2} = C^H + C^L_i,\ D_{i3} = P^H + P^L_i,\ D_{i4} = A^L_i$$

- Tiến hành chuẩn hóa ma trận *D* để thu được ma trận chuẩn hóa *N* theo công thức:

$$N_{ij} = \frac{D_{ij}}{\sqrt{\sum D^2_{ij}}}$$

- Xây dựng ma trận thủy vân *W* bằng cách nhân ma trận chuẩn hóa *N* với ma trận chuyển vị N^T của nó. Ma trận *W* thu được là một ma trận vuông kích thước 4 × 4 với các giá trị trên đường chéo chính là e_1, e_2, e_3, e_4. Đây chính là các giá trị đặc trưng của ma trận này.

- Điểm khác của lược đồ thủy vân [3] và [1] là dùng hàm băm các giá trị e_j sau khi ghép với khóa thủy vân *K*. Chuyển các giá trị băm thành các ký tự thủy vân W_j theo công thức:

$$W_j = \text{UNICHAR}\ (H(e_j \| K) \bmod 224 + 32);\quad j = 1, 2, 3, 4$$

Trong đó UNICHAR() là hàm chuyển mã Unicode thành ký tự tương ứng. Sở dĩ phải cộng thêm 32 là vì 31 ký tự đầu tiên trong bảng mã Unicode là các ký tự không in ra được. Khóa *K* là bí mật và đối xứng, chỉ người chủ cơ sở dữ liệu được biết và được dùng ở cả quá trình nhúng thủy vân và phát hiện thủy vân. Hàm băm được sử dụng ở đây để đảm bảo rằng khi có bất cứ một thay đổi nào xảy ra trong cơ sở dữ liệu thì các ký tự thủy vân W_j cũng thay đổi theo. Đây chính là điều mong muốn đối với các lược đồ thủy vân dùng để đảm bảo sự toàn vẹn của các cơ sở dữ liệu quan hệ.

Các ký tự thủy vân W_j sau khi được sinh ra được nhúng vào cuối các thuộc tính văn bản tác động thấp của quan hệ.

Quá trình sinh thủy vân ở trên có một số nhược điểm sau:

- Ký tự thủy vân sinh ra là ngẫu nhiên trong khoảng mã ký tự 32 đến 255 do đó nó là các ký tự dễ phát hiện và chiếm chỗ.

- Các ký tự thủy vân được nhúng vào tất cả các thuộc tính tác động thấp khiến cho cơ sở dữ liệu tăng dung lượng lên đáng kể sau khi nhúng.

- Do thủy vân chỉ thực hiện trên các quan hệ được chọn của lược đồ, nên nếu có bất kỳ một quan hệ khác trong lược đồ quan hệ bị sửa đổi thì không đảm bảo sự toàn vẹn cho CSDL này.

2) Sinh thủy vân cải tiến

Để tính tổng của V, *C*, *P*, *A* ở trên chuyển các giá trị của các thuộc tính tác động cao H_i không phải là văn bản về kiểu văn bản bằng hàm *toString*(*S*) rồi tính tổng các mã ký tự của các thuộc tính thuộc tính như sau:

Với mỗi bộ t_i thuộc *R*, nếu $t_i.H_j$ có kiểu khác với kiểu văn bản thì *sValue* = toString($t_i.H_j$), ngược lại thì *sValue* = $t_i.H_j$

$$V^H = V^H + \sum \text{Mã Unicode của các nguyên âm thuộc } sValue$$

$$C^H = C^H + \sum \text{Mã Unicode của các phụ âm thuộc } sValue$$

$$P^H = P^H + \sum \text{Unicode của các ký tự đặc biệt thuộc } sValue$$

$$V^L_j = V^L_j + \sum \text{Mã Unicode của các nguyên âm thuộc } t_i.L_j$$

$$C^L_j = C^L_j + \sum \text{Mã Unicode của các phụ âm thuộc } t_i.L_j$$

$$P^L_j = P^L_j + \sum \text{Mã Unicode của các ký tự đặc biệt thuộc } t_i.L_j$$

$$A^L_j = A^L_j + \sum \text{Mã Unicode của tất cả các ký tự thuộc } t_i.L_j$$

Điểm khác biệt thứ 2 là dùng một mảng *W* với *4* phần tử là các ký tự Unicode, mã này có đặc tính là không hình dạng (rỗng) và không chiếm chỗ, mảng này được tạo ra bằng cách sử dụng hàm băm các giá trị e_j ghép với khóa thủy vân *K*. Để xác định ký tự thủy vân trong *W*, tính W_j theo công thức sau:

W_j = UNICHAR (H(e_j || *K*) mod *32*) (*j* = 1, 2, 3, 4)

Với:

- *K* là khóa bí mật và đối xứng, chỉ người chủ cơ sở dữ liệu được biết và được dùng ở cả quá trình nhúng thủy vân và phát hiện thủy vân.

- Hàm băm H được sử dụng ở đây để đảm bảo rằng khi có bất cứ một thay đổi nào xảy ra trong cơ sở dữ liệu thì các ký tự thủy vân W_j cũng thay đổi theo. Đây chính là điều mong muốn đối với các lược đồ thủy vân dùng để đảm bảo sự toàn vẹn của các cơ sở dữ liệu quan hệ.

- || là phép toán nối chuỗi.

- Hàm UNICHAR là hàm chuyển mã Unicode thành ký tự tương ứng.

Lược đồ nhúng, phát hiện thủy vân có sử dụng thủ tục sinh thủy vân từ lược đồ cơ sở dữ liệu quan hệ R có *m* thuộc tính tác động cao, *n* thuộc tính tác động thấp và khóa thủy vân *K*, thủ tục đó được xây dựng như sau:

```
Procedure SinhTV(R, K)
Input:  R, K
Output: W
 1.V^H = C^H = P^H = 0
 2.For r_i ∈ R  Do
 3.  for i = 1 to ω do
 4.    for j =1 to m do
 5.          if    t_i.H_j  khác  với  kiểu  văn  bản  then  sValue  =
t_i.H_j.toString()
 6.       else   sValue = t_i.H_j
 7.       end if
   8.         V^H = V^H +  ∑  Mã Unicode của các nguyên âm thuộc sValue
   9.         C^H = C^H + ∑ Mã Unicode của các phụ âm thuộc sValue
   10.       P^H = P^H + ∑ Unicode của các ký tự đặc biệt thuộc sValue
   11.       end for
   12.      end for
   13.      for i = 1 to ω do
   14.       V^L_j  = C^L_j  =  P^L_j  = A^L_j = 0
   15.        for j = 1 to  n  do
   16.          sValue = r_i.L_j
     17.      V^L_j = V^L_j +  ∑ Unicode của các nguyên âm thuộc sValue
     18.      C^L_j = C^L_j +  ∑ Unicode của các phụ âm thuộc sValue
     19.      P^L_j = P^L_j +  ∑ Unicode của các ký tự đặc biệt thuộc sValue
     20.      A^L_j = A^L_j + ∑ Unicode của tất cả các ký tự thuộc sValue
     21.     end for
     22.    end for
     23.  End For
     24.   for i = 1 to n do
     25.        D_i1 = V^H + V^L_i
     26.        D_i2 = C^H + C^L_i
     27.        D_i3 = P^H + P^L_i
     28.        D_i4 =A^L_i
     29.    end for
     30.  for i = 1 to n do
     31.     for  j = 1 to  4  do
     32.        N_ij = D_ij /  sqrt(D^2_i1 + D^2_i2 + D^2_i3 + D^2_i4)
     33.        N^T_ji = N_ij
     34.    end for
     35. end for
```

```
36. W = N^T*N
37. for j = 1 to 4 do
38.   W_j  = UNICHAR (H(e_j || K) mod 32 )
39.  end for
40. Sắp thứ tự của W
41. return W
```

3) Chia nhóm các bộ của quan hệ

Cho quan hệ r thuộc R với ω bộ dữ liệu, khi đấy việc phân chia ω bộ của quan hệ r thành các g nhóm dựa vào khóa chính của bộ và khóa thủy vân K. Cách phân chia này sẽ làm tăng tính ngẫu nhiên khi chọn các bộ và phân các bộ vào các nhóm riêng rẽ. Tính ngẫu nhiên này có độ bảo mật cao được đảm bảo bằng hàm băm mật mã H(). Mục đích của việc phân chia này nhằm làm tăng khả năng bền vững của thuỷ vân trước các tấn công và phát hiện giả mạo nếu có.

Với số lượng nhóm g, khóa thủy vân K chỉ chủ sở hữu cơ sở dữ liệu quan hệ biết, việc phân chia các bộ của quan hệ vào nhóm G_k (k= 0, 1, …, g - 1) sẽ được thực hiện bằng công thức sau:

Với mỗi bộ $t_i \in r$, $t_i \in G_k$ với k = H(K || $t_i.P$) mod g ($\forall$ i = 1, 2, ..., ω), trong đó, H(K || $t_i.P$) là hàm băm khóa thủy vân K cùng với giá trị thuộc tính khóa chính P của bộ t_i và || là phép ghép nối.

Sau đây là thủ tục chia nhóm các bộ dữ liệu.

Procedure ChiaNhom(*r, K, g*)

```
1.  for k = 0 to g-1 do // khởi tạo các chỉ số và các nhóm
2.        q_k = 0
3.        G_k = ∅
4.  end for
5.  for i = 1 to ω do
6.        k = H(K || t_i.P) mod g
7.        G_k = G_k ∪ {t_i}
8.        q_k = q_k + 1 // q_k là số bộ trong nhóm G_k
9.  end for
```

4) Nhúng thủy vân vào thuộc tính tác động thấp

Đầu tiên, tiến hành lựa chọn bộ để nhúng thủy vân trong nhóm G_j của quan hệ r:

$$t = \text{CODE}(W_{(j \bmod 4)+1}) \bmod q_j$$

Trong đó q_j là số bộ của nhóm G_j, hàm CODE là hàm chuyển ký tự thành mã Unicode tương ứng.

Sau đó, tiến hành xác định thuộc tính và vị trí nhúng để nhúng ký tự thủy vân trong bộ đã chọn.

Trong lược đồ thủy vân có sử dụng thủ tục nhúng thủy vân vào các thuộc tính tác động thấp trong một nhóm, thủ tục đó được xây dựng như sau:

Procedure NhungTV((*G_j, W*))

```
1.  // Chọn bộ thứ t trong G_j để nhúng
2.  t  = CODE(W_(j mod 4)+1) mod q_j
3.  //Xác định  thuộc tính nhúng
4.  e = t mod n
5.  vtrinhung = Converter(H(K))%length(r_t.L_e)
6.   Chèn W_(j mod 4)+1 vào r_t.L_e tại vtrinhung
```

5) Nhúng thủy vân vào cơ sở dữ liệu quan hệ

Thuật toán 1: Nhúng thủy vân

Input: **-** Lược đồ $R(H_1, H_2,…,H_m, L_1, L_2, ..., L_n)$. Trong đó, $H_1, H_2,…, H_m$ là các thuộc tính kiểu văn bản có tác động cao, còn $L_1, L_2, ..., L_n$ là các thuộc tính kiểu văn bản có tác động thấp.

- Khóa nhúng thủy vân K.
- Số nhóm g

Output: **-** Lược đồ *R* đã nhúng thủy vân.

```
1.      W = SinhTV (R,K)
2.      For r_i ∈ R  Do
3.          ChiaNhom(r_i,K,g)
4.          for j = 0 to g-1 do
5.             NhungTV(G_j, W)
6.          end for
7.      end For
```

6) Nhận xét

Với việc tính tổng các giá trị nguyên âm, phụ âm và các ký tự đặc biệt trên toàn bộ lược đồ R thay vì tính trên quan hệ *r* riêng lẻ khiến ký tự thủy vân sinh ra mang tính bao quát hơn. Mọi thay đổi trên quan hệ bất kỳ của lược đồ đều ảnh hưởng đến ký tự thủy vân được sinh ra điều này có nghĩa ta có thể phát hiện được sự thay đổi trên cả các quan hệ không nhúng thủy vân lên trên.

Việc sử dụng các ký tự không chiếm chỗ không hiện hình và chèn các ký tự này vào vị trí bất kỳ trên các thuộc tính văn bản tác động thấp khiến cho việc phát hiện thủy vân trở nên khó khăn hơn.

Với việc chia nhóm các bộ trên quan hệ nhúng thủy vân, với mỗi nhóm G_i nhúng một ký tự thủy vân như trình bày ở trên khiến dung lượng tăng thêm của cơ sở dữ liệu là không lớn (một quan hệ nhúng tối đa *g* ký tự thủy vân).

B. ***Thuật toán phát hiện thuỷ vân***

Quan hệ *r* thuộc *R* sau khi nhúng thủy vân có thể lưu thông bình thường trong môi trường công cộng. Khi có nghi ngờ về một sự xuyên tạc hay giả mạo nào đó đối với quan hệ này, người chủ sở hữu quan hệ có thể tiến hành xác minh bằng thuật toán phát hiện thủy vân.

Giả sử *R'* là lược đồ cơ sở dữ liệu cần kiểm tra xem có phải là giả mạo của lược đồ *R* đã thủy vân hay không. Thuật toán phát hiện thủy vân và xác minh sự toàn vẹn được chia làm 3 phần cơ bản:

- Thực hiện quy trình sinh thủy vân trên *R'* giống như đã thực hiện trên *R* trong thuật toán nhúng thủy vân. Vị trí các mã ký tự nhúng được sinh ra từ *R'* là W_j với j = 1, 2, 3, 4.

- Theo các quy tắc nhúng thủy vân vào các thuộc tính tác động thấp, trích ra các mã ký tự đã nhúng và gọi các ký tự này là W'_j với j = 1, 2, 3, 4.

- So sánh các ký tự của *W'* với các ký tự của *W* với nhau. Nếu chúng trùng khớp thì *R'* chính là *R* và không bị sửa đổi. Nếu ngược lại thì kết luận là *R* đã bị sửa đổi.

Thuật toán 2: Phát hiện thủy vân

Input: **-** Lược đồ $R'(H_1, H_2,\ldots,H_m, L_1, L_2, \ldots., L_n)$. Trong đó, $H_1, H_2,\ldots, H_m$ là các thuộc tính kiểu văn bản có tác động cao, còn $L_1, L_2, \ldots., L_n$ là các thuộc tính kiểu văn bản có tác động thấp.

- Khóa nhúng thủy vân K.
- Số nhóm *g*

Output: **-** Lược đồ *R* toàn vẹn hay đã bị sửa đổi.

```
1.   W = SinhTV (R',K)
2.   For r_i ∈ R'  Do
3.       ChiaNhom(r_i,K,g)
4.       for j = 0 to g-1 do
5.         NhungTV(G_j, W)
6.            w'_j = substring(r'_t.L_e, vtrinhung,1)
7.       End for
8.     Sắp thứ tự W'
9.      if W <> W' then
10.              return "Lược đồ bị sửa đổi"
11.     end if
12.   End For
```

Giả sử có trung bình p ký tự trong mỗi thuộc tính của một bộ và mỗi bộ có $m + n$ thuộc tính. Khi đó, việc tính tổng các ký tự có trong một bộ tốn khoảng $p(m + n)$ đơn vị thời gian. Vậy thời gian tính toán cho lược đồ thủy vân là $\omega p(m + n)$ đơn vị thời gian hay độ phức tạp là $O(\omega p(m + n))$.

IV. TÍNH ĐÚNG ĐẮN

Định lý: Thuật toán phát hiện thủy vân xác định sự toàn vẹn của lược đồ quan hệ R nếu lược đồ quan hệ R không bị sửa đổi.

Chứng minh. Để chứng minh định lý sẽ chứng minh W_j và W'_j là trùng khớp. Thật vậy:

- Dựa vào m thuộc tính tác động cao và n thuộc tính tác động thấp tính được ma trận D bao gồm 4 hàng và n cột với các thành phần là D_{i1}, D_{i2}, D_{i3}, D_{i4} (với i = 1, 2, …, n) với:

$D_{i1} = V^H + V^L_i$

$D_{i2} = C^H + C^L_i$

$D_{i3} = P^H + P^L_i$

$D_{i4} = A^L_i$

Trong đó V^H, V^L_i, C^H, C^L_i, P^H, P^L_i, A^L_i là tổng mã Unicode của các nguyên âm, phụ âm, ký tự đặc biệt và tổng mã Unicode của tất cả các ký tự theo từng thuộc tính.

Chuẩn hóa ma trận D để thu được ma trận chuẩn hóa N theo công thức:

$$N_{ij} = \frac{D_{ij}}{\sqrt{\sum D_{ik}^2}}$$

$\Rightarrow W_j$ (j = 1, 2, 3, 4) (1)

- Theo thuật toán phát hiện thủy vân, ta có:

+ Phân chia các bộ t_i của quan hệ $r \in R$ vào nhóm G_k (k = 0, 1, …, g-1) sẽ được thực hiện theo các bước sau: Với mỗi $t_i \in r$, $t_i \in G_k$ với $k = H(K \parallel t_i.P) \bmod g$

+ Các bộ của r được chia nhóm trong G. Để thủy vân nhóm G_i gồm q_i bộ, đầu tiên chọn bộ thứ $t \in G_i$ để nhúng thủy vân như sau:

$$t = CODE(W_{(i \bmod 4)+1}) \bmod q_i$$

+ Sau khi xác định được bộ thứ t cần nhúng thủy vân, vị trí nhúng thủy vân trong thuộc tính L_e của bộ thứ t được thực hiện như sau:

$$e = t \bmod n$$

$$vtrinhung = Converter(H(K))\%length(r_t.L_e)$$

$\Rightarrow$ bit thủy vân W'_j (j = 1, 2, 3, 4) (2)

- Theo giả thiết lược đồ quan hệ R không bị sửa đổi nên: $t'.L_i = t.L_i$ (3)

Từ (1), (2) và (3) $\Rightarrow W$ trùng với $W' \Rightarrow$ điều phải chứng minh.

V. KẾT LUẬN

Lược đồ thủy vân cải tiến nhằm đảm bảo sự toàn vẹn của cơ sở dữ liệu quan hệ có những ưu điểm sau:

- Bền vững: Các giá trị đặc trưng của lược đồ có thể được nhúng trong thuộc tính có tác động thấp ở khắp nơi trong quan hệ. Vì vậy rất khó để có thể gỡ bỏ hết các ký tự thủy vân đã nhúng.

- Nhạy cảm: Mọi thay đổi trên quan hệ bất kỳ của lược đồ đều ảnh hưởng đến ký tự thủy vân được sinh ra điều này có nghĩa ta có thể phát hiện được sự thay đổi trên cả các quan hệ không nhúng thủy vân lên trên.

- Phát hiện mù: Quá trình xác minh sự toàn vẹn của lược đồ không đòi hỏi lược đồ gốc và thủy vân gốc.

- Không hiện: Thủy vân được nhúng là các ký tự có đặc tính là không hình dạng (rỗng) và không chiếm chỗ.

Nhược điểm của lược đồ là việc sử dụng các ký tự không chiếm chỗ không hiện hình và chèn các ký tự này vào vị trí bất kỳ trên các thuộc tính văn bản tác động thấp khiến cho việc khôi phục dữ liệu gốc trở nên khó khăn hơn.

TÀI LIỆU THAM KHẢO

[1] Bedi R., Thengade A., Wadhai V. (2011), "A New Watermarking Approach for Non Numeric Relational Database". International Journal of Computer Applications (0975 - 8887), Vol 13, No 7, pages 37-40.

[2] Hamadou A., Sun X., Shah S. A. and Gao L. (2011), "A Weight-based Semi-Fragile Watermarking Scheme for Integrity Verification of Relational Data", International Journal of Digital Content Technology and its Applications, Volume 5, Number 8, pages 148-157.

[3] Lưu Thị Bích Hương, Bùi Thế Hồng (2014), "Đảm bảo sự toàn vẹn của cơ sở dữ liệu quan hệ với các dữ liệu kiểu văn bản bằng kỹ thuật thủy vân", Tạp chí Tin học và Điều khiển học, T.30, S.1, tr. 52-62.

ENSURING INTEGRITY FOR RELATIONAL DATABASES WITH ATTRIBUTE TEXT BY WATERMARK SCHEME

Luu Thi Bich Huong, Bui The Hong

***ABSTRACT**: This paper proposed an improved scheme for ensuring the integrity relational databases with attribute text. Any change in any relation of relational scheme affect the generated character watermarking, which means that can be detected changes in non-embedded watermark relationships.*