*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.2017.000113*

# ỨNG DỤNG PHƯƠNG PHÁP THỦY VÂN ĐỂ BẢO VỆ CƠ SỞ DỮ LIỆU ĐIỂM HỌC TẬP TRONG TRƯỜNG ĐẠI HỌC

**Chung Nam Phong[1], Trần Cao Đệ [2]**

[1,2] Khoa Công nghệ thông tin, Trường đại học Cần Thơ

*chungnamphong@gmail.com, tcde@cit.ctu.edu.vn*

***TÓM TẮT:*** *Ngày nay, việc ứng dụng công nghệ thông tin vào trong quản lí ở các trường đại học, cao đẳng và các cấp học khác là rất phổ biến và đã chứng tỏ được hiệu quả tích cực. Một trong những cơ sở dữ liệu quan trọng trong các nhà trường là cơ sở dữ liệu lưu trữ kết quả học tập dạng điểm số, được quản lí, truy cập thông qua phần mềm quản lí của nhà trường. Có thể thấy rằng phần mềm quản lý và cơ sở dữ liệu phải được xây dựng cho phép nhiều người dùng, tức là nhiều giảng viên, cán bộ liên quan có thể truy cập, cập nhật. Chức năng bảo vệ cơ sở dữ liệu được xây dựng trong phần mềm và hệ quản trị cơ sở dữ liệu bằng cơ chế phân quyền, đăng nhập, ghi nhật kí (log) cập nhật. Tuy nhiên, phần mềm quản lí và quyền truy cập hệ thống thường được giao cho bên thứ 3, tức là người phát triển phần mềm với sự quản lí chưa thật chặt chẽ. Từ đó dẫn đến nguy cơ sẽ có những truy cập trực tiếp vào cơ sở dữ liệu từ hệ quản trị cơ sở dữ liệu. Vấn đề đặt ra là làm sao có thể phát hiện được các sửa đổi không hợp lệ, tức là sửa đổi không thông qua phần mềm quản lí? Làm sao một giảng viên hay cán bộ có thể xác nhận được điểm số đó là nguyên gốc như mình đã nhập vào từ nhiều năm trước và chưa hề bị sửa đổi một cách không hợp lệ?*

*Kỹ thuật thủy vân (watermarking) từ lâu nghiên cứu và có thể áp dụng để xác thực nguồn gốc, bảo vệ bản quyền các file ảnh. Kỹ thuật này kết hợp giữa việc mã hóa và giấu tin với mục đích bảo vệ bản quyền và toàn vẹn đối với sơ sở dữ liệu. Trong bài viết này chúng tôi trình bày các nghiên cứu sử dụng cách tiếp cận thủy vân để bảo vệ cơ sở dữ liệu quan hệ như điểm số nhằm để bảo vệ tốt hơn cơ sở dữ liệu điểm trong các nhà trường.*

***Từ khóa:*** *Điểm số, thuộc tính số, thủy vân, cơ sở dữ liệu.*

## I. GIỚI THIỆU

Thủy vân là phương pháp dùng các thông tin bí mật kết hợp với các đặc trưng của dữ liệu, sau đó áp dụng giải thuật chèn vào dữ liệu mà người sử dụng thông thường khó có thể phát hiện được sự có mặt của thủy vân. Thủy vân được trích xuất để xác nhận bản quyền hay toàn vẹn đối với dữ liệu. Ưu điểm của thủy vân so với các cơ chế bảo vệ khác là thuật toán tự xây dựng theo mục đích sử dụng, chỉ có chủ sở hữu dữ liệu giữ thông tin bí mật trong quá trình nhúng và trích xuất thủy vân, không ảnh hưởng đến tính khả dụng của dữ liệu.

Thủy vân trên cơ sở dữ liệu về cơ bản được phân loại dựa vào 2 yếu tố: tác động của thủy vân đối với cơ sở dữ liệu, kiểu của dữ liệu được thủy vân.

Đã có nhiều công trình nghiên cứu về thủy vân trên cơ sở dữ liệu quan hệ có các thuộc tính kiểu số [1], [2], [3], [4], [5]. Tuy nhiên, một phương pháp thủy vân trên cơ sở dữ liệu điểm cần đạt được những yêu cầu sau:

- Kiểm tra và phát hiện: để đảm bảo được chủ quyền của dữ liệu thì thủy vân nhúng vào phải được phát hiện bởi chủ sở hữu dữ liệu đó. Ngoài ra, khi có một thay đổi nào trên dữ liệu thì có thể kiểm tra được sự thay đổi đó dựa vào thủy vân nhúng vào.

- Tính mờ (blind) : khi áp dụng kỹ thuật thủy vân số thì không cần đến dữ liệu gốc ban đầu mà vẫn có thể xác định dữ liệu có bị thay đổi hay không, nếu có thì xác định vị trí dữ liệu bị thay đổi.

- Có cơ chế bảo vệ từng điểm số và từng bản ghi (record) điểm : thủy vân nhúng vào dữ liệu phải mang đặc trưng của những giá trị trong các thuộc tính của từng bộ dữ liệu, đảm bảo phát hiện được những thay đổi dù nhỏ đối với mỗi bộ dữ liệu.

- An toàn: thủy vân nhúng vào phải bền vững trước các kiểu tấn công thông thường và có chủ đích nhằm mục đích phá hủy thủy vân đã nhúng vào cơ sở dữ liệu.

- Dễ sử dụng: đảm bảo các người dùng có thể sử dụng chương trình thủy vân dễ dàng.

Chúng tôi đề xuất phương pháp thủy vân để bảo vệ cơ sở dữ liệu điểm số dựa vào việc chèn thêm một thuộc tính, mỗi giá trị của thuộc tính này được tính toán dựa vào hàm tổng hợp các giá trị thuộc tính khác trên cùng 1 bộ.

## II. CÁC NGHIÊN CỨU LIÊN QUAN

Trong bài viết này chúng tôi chỉ chú trọng trình bày các kỹ thuật thủy vân liên quan đến cơ sở dữ liệu quan hệ có các thuộc tính kiểu số.

Giải thuật đề xuất bởi R. Agarwal and J Kiernan [1] trên cơ sở dữ liệu quan hệ cơ bản dựa trên khóa chính và khóa bí mật. Phương pháp đã được đề xuất là nhúng 1 bit thủy vân đơn lẻ vào thuộc tính số của cơ sở dữ liệu và trích xuất nó dựa vào giải thuật trích xuất. Cơ sở dữ liệu quan tâm là cơ sở dữ liệu quan hệ có các thuộc tính số có thể chấp

nhận được những thay đổi nhỏ và không ảnh hưởng đến khả năng ứng dụng của nó. Giải thuật này được tạo nên từ nhiều tham số được giữ bí mật bởi chủ sở hữu dữ liệu nên tính an toàn cao, tuy nhiên cần xác định được ngưỡng tác động đối với dữ liệu, phương pháp này không thích hợp để áp dụng với cơ sở dữ liệu điểm số vì nó làm thay đổi giá trị của dữ liệu, trong khi đó điểm số không chấp nhận được giá trị sai lệch.

Trong [6] tác giả dùng phương pháp chèn 1 thuộc tính ảo vào dữ liệu gốc, thuộc tính này chứa giá trị được tính toán dựa vào hàm tổng hợp, giá trị tại đây được dùng làm giá trị kiểm tra giá trị của các thuộc tính khác trong cùng 1 bộ. Ở giai đoạn nhúng: tác giả phân chia các bộ dữ liệu vào các nhóm không chồng chéo lên nhau với số nhóm được giữ bí mật, sau đó chèn thêm thuộc tính ảo $V_A$ với các giá trị được tạo ra bằng cách sử dụng hàm tổng hợp, tính toán các giá trị trong các phân vùng đang xét. Do đó, thủy vân chèn vào không làm thay đổi dữ liệu mà thay vào đó là một thuộc tính với tính chất kiểm tra các giá trị còn lại. Ở giai đoạn trích xuất thủy vân: tác giả cũng thực hiện lại quá trình như lúc thủy vân để có được giá trị thuộc tính $V_{A'}$ và so sánh với giá trị cột $V_A$ , nếu giống nhau hoàn toàn thì dữ liệu an toàn, ngược lại dữ liệu đã bị thay đổi. Ưu điểm của phương pháp này là: không làm thay đổi giá trị nội dung dữ liệu, không phụ thuộc vào khóa chính, có nhiều cách chọn hàm tổng hợp cho giá trị thuộc tính ảo, khả năng phát hiện thay đổi dữ liệu cao. Tuy nhiên, phương pháp này có nhược điểm: tốn nhiều dung lượng lưu trữ thuộc tính chèn vào, giá trị của thuộc tính chèn vào chưa được mã hóa, thay đổi cấu trúc của cơ sở dữ liệu quan hệ được thủy vân. Phương pháp này tuy có thay đổi cấu trúc của cơ sở dữ liệu, tuy nhiên giá trị dữ liệu không bị thay đổi khi được sử dụng trong các hệ thống quản lý. Ngoài ra, nếu chọn hàm tổng hợp tốt thì giá trị chèn vào có thể kiểm soát chặt chẽ được các giá trị khác trên cùng một bộ. Với các tính chất trên cho thấy phương pháp được trình bày trong [6] có thể áp dụng tốt cho thủy vân cơ sở dữ liệu điểm số.

Trong nghiên cứu [7], ý tưởng chính cũng là áp dụng kỹ thuật thủy vân không làm thay đổi trên cơ sở dữ liệu được nhúng thủy vân, kỹ thuật thủy vân dựa trên phân vùng. Phân vùng có thể được xem như một nhóm ảo, không thay đổi giá trị của các phần tử của bảng cũng như các vị trí vật lý của chúng. Thay vì chèn thủy vân trực tiếp vào phân vùng cơ sở dữ liệu, chúng ta coi nó như là một biểu diễn trừu tượng của các phân vùng cụ thể, sao cho bất kỳ sự thay đổi nào trong các phân vùng cụ thể lại phản ánh trong đối tượng trừu tượng của nó. Giải pháp đã được đề ra là tạo một hình ảnh xám của phân vùng như là một thủy vân của phân vùng đó, nó phục vụ như là thủ tục phát hiện giả mạo, sau đó là cơ chế xác thực không tác động lên dữ liệu để xác minh quyền sở hữu.

Trong công trình [8], các tác giả cố gắng giấu các bit thủy vân vào những phần của thuộc tính thời gian mà không được chú ý đến trong những bộ dữ liệu. Trong thuộc tính thời gian bao gồm 2 phần ngày và giờ, trong phần giờ thì giây được sử dụng để ẩn thông tin (HH:MM:SS). Phương pháp này cơ bản dựa vào ảnh nhị phân dùng làm thủy vân, thuận lợi của việc sử dụng thuộc tính thời gian là số lượng lớn bit tiềm năng có thể ẩn thông tin thủy vân và vì vậy lượng lớn thủy vân có thể giấu dễ dàng nếu cần. Nhược điểm của phương pháp này là không thể áp dụng cho cơ sở dữ liệu không có thuộc tính thời gian, hoặc nếu thời gian bị làm tròn (bỏ giây) lúc này toàn bộ thủy vân sẽ bị mất. Phương pháp này tác động đến dữ liệu nhưng không làm mất khả năng sử dụng của dữ liệu, có thể áp dụng cho dữ liệu điểm số nếu cơ sở dữ liệu có tổ chức thêm thuộc tính thời gian.

Trong nghiên cứu của [9], [11] các tác giả đã sử dụng thuật toán mã hóa RSA để bảo vệ thông tin thủy vân. Trong phương pháp của [9] sử dụng chữ ký số để làm thông tin thủy vân, tác giả chia nhóm dữ liệu dựa vào số nhóm được giữ bí mật, sau đó sử dụng hàm băm SHA-512 tạo chữ ký số của nhóm. Chữ ký được mã hóa bằng RSA với Public_key và gửi tới người nhận. Chữ ký cũng được chèn vào dữ liệu bởi việc sinh thêm 1 cột ảo trong bảng dữ liệu, và coi đây là thủy vân. Trong quá trình trích xuất, dữ liệu nghi ngờ sẽ được phân vùng lại và tạo chữ ký số, so sánh với chữ ký được giải mã bằng Private_key để xác định dữ liệu có an toàn hay không. Với phương pháp này hỗ trợ tốt cho hệ thống nhiều người dùng mà vẫn đảm bảo tính bảo mật của thủy vân. Tuy nhiên, phương pháp này đòi hỏi phải có thủy vân gốc trong quá trình trích xuất thủy vân.

Ngoài ra, với phương pháp được trình bày trong [10], tác giả đã nhúng thủy vân ở 2 nơi dựa vào 2 khóa k1 và k2. Đầu tiên tác giả nhúng bit thủy vân tại LSB của thuộc tính số theo giải thuật của Agrawal et al [3], sau đó là nối kết bit thủy vân vào khóa k2 và nhúng vào trường giây (HH:MM:SS) trong thuộc tính thời gian. Trong phương pháp này thủy vân được nhúng ở 2 nơi nên độ an toàn cao hơn, gây khó khăn hơn cho kẻ tấn công muốn phá hủy thủy vân được nhúng. Tuy nhiên, phương pháp này cũng yêu cầu dữ liệu phải có thuộc tính thời gian và cũng có tác động thay đổi trên cơ sở dữ liệu vì thế phương pháp này không phù hợp với những cơ sở dữ liệu liệu có dạng như điểm số.Ý tưởng chính trong phương pháp thủy vân của chúng tôi là giữ nguyên giá trị của dữ liệu, đảm bảo an toàn và toàn vẹn với cơ sở dữ liệu điểm số. Ngoài ra, phương pháp áp dụng cần phải đơn giản, dễ sử dụng có thể áp dụng cho các hệ thống quản lý điểm tại các trường học.

## III. LƯỢC ĐỒ THỦY VÂN

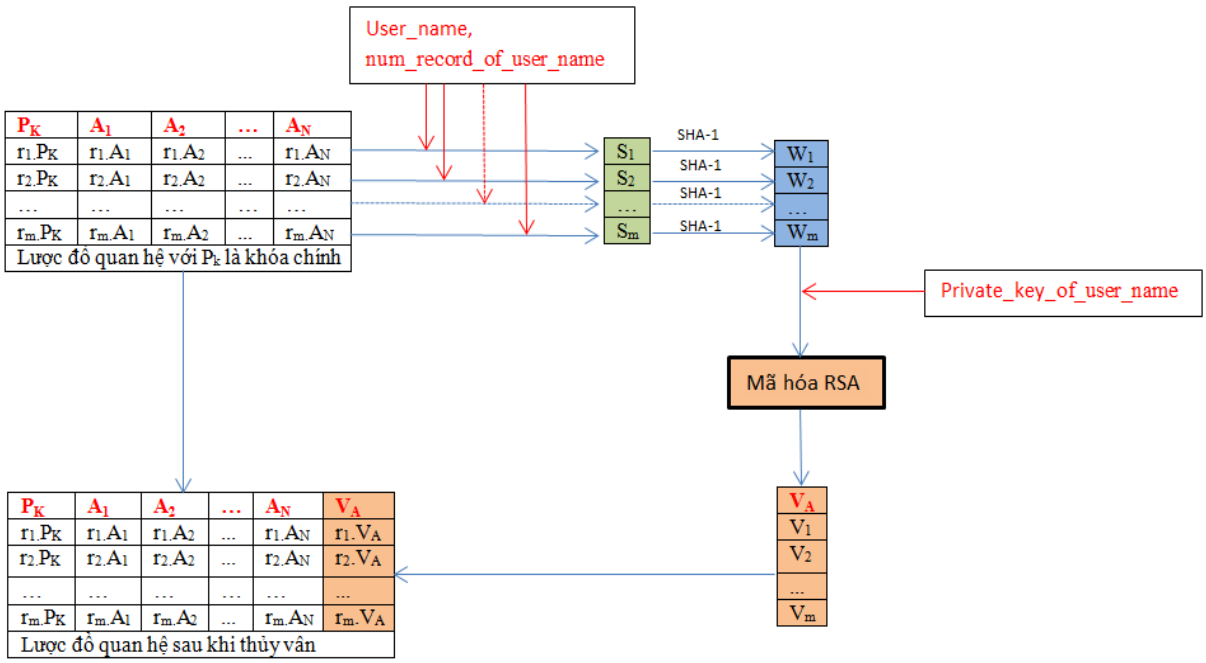
Cơ sở dữ liệu điểm số là cơ sở dữ liệu quan hệ có các thuộc tính số, phương pháp chèn thủy vân nếu có tác động lên dữ liệu thì việc xác định ngưỡng sẽ rất khó khăn. Vì thế, những phương pháp không làm thay đổi trên dữ liệu được ưu tiên lựa chọn để thủy vân cơ sở dữ liệu dạng điểm số.

Một đặc điểm cần được quan tâm đó là trong hệ thống quản lý điểm có nhiều giáo viên là user nhập điểm, khi cần kiểm tra các điểm số này thì ngoài user đó còn có người quản lý chung các điểm số này. Và khi đó, người quản lý

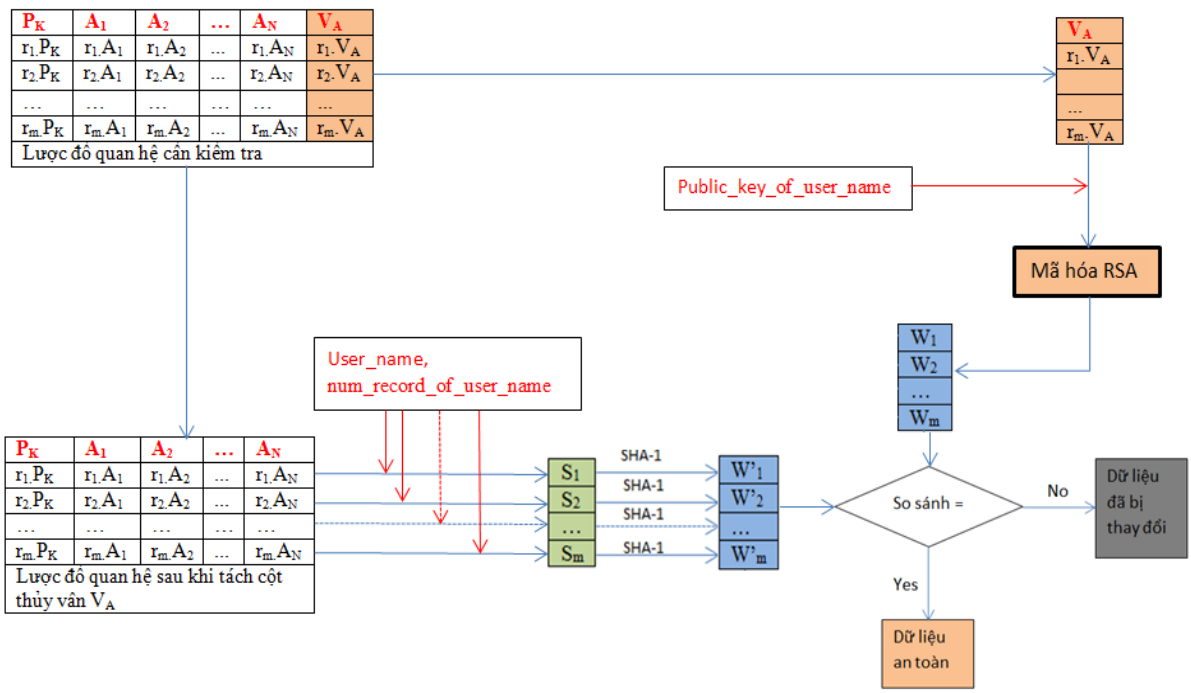
chung có thể kiểm tra được các điểm số là an toàn hay không mà không cần sự có mặt của user nhập điểm hay các thông tin bí mật của các user này. Tuy nhiên, đối với các phương pháp thông thường, chủ sở hữu dữ liệu cung cấp khóa bí mật để thủy vân dữ liệu và cũng sử dụng khóa này để trích xuất thủy vân xác định an toàn đối với dữ liệu.

Từ các vấn đề trên cho thấy cần 1 phương pháp đáp ứng được các yêu cầu đối với dữ liệu điểm số trong các hệ thống quản lý điểm. Chúng tôi đề xuất phương pháp thủy vân dựa trên lược đồ thủy vân của [6], tức là chèn vào 1 thuộc tính ảo $V_A$ dùng để kiểm tra các giá trị khác trên cùng một bộ, chúng tôi sử dụng thuật toán mã hóa RSA để mã hóa đặc trưng của dữ liệu. Thuật toán RSA với cặp khóa Private_key và Public_key có thể giải quyết được vấn đề kiểm tra dữ liệu mà không cần các thông tin bí mật của chủ sở hữu. Ngoài ra, với bản rõ có chiều dài đủ lớn thì độ an toàn của bản mã là khá cao khó có thể dịch ngược được. Sau đây chúng tôi sẽ trình bày cụ thể về lược đồ thủy vân thực hiện:

**Mô hình thủy vân:**




**Hình 1.** Quá trình thủy vân




**Hình 2.** Quá trình trích xuất thủy vân

Xét lược đồ quan hệ có cấu trúc N thuộc tính, m bộ dữ liệu, khóa chính là $P_K$.

**Giai đoạn nhúng thủy vân (hình 1):**

- Trích xuất đặc trưng dữ liệu của R

Mỗi bộ của quan hệ R được tạo ra bởi một user, đặc trưng dữ liệu chính là xác định user_name là chủ sở hữu của bộ $r_i$, giá trị của các thuộc tính trong bộ đó. Ngoài ra, để đảm bảo toàn vẹn dữ liệu thì đặc trưng dữ liệu sẽ tích hợp số bộ dữ liệu của từng user_name. Như vậy đặc trưng dữ liệu của bộ thứ i có dạng như sau:

$$S_i = \text{user\_name}\| r_i.P_K\| r_i.A_1\| r_i.A_2\|...\| r_i.A_N\|\text{num\_record\_of\_user\_name}$$

- Tạo giá trị băm từ đặc trưng dữ liệu của R

Mục đích của việc này là tăng độ an toàn đối với đặc trưng dữ liệu hay nói cách khác đó là độ an toàn của bản rõ trước khi đưa vào thuật toán mã hóa RSA. Ở đây chúng tôi sử dụng hàm băm SHA-1 để chuyển đổi $S_i$ thành một chuỗi ký tự có độ dài 160bit, là giá trị đầu vào của bước tiếp theo.

$$W_i = \text{SHA-1}(S_i)$$

- Tạo giá trị thủy vân bằng thuật toán mã hóa RSA với khóa Private_key

Giá trị băm của bản rõ Wi có được từ bước trên sẽ được mã hóa bằng thuật toán RSA với khóa bí mật Private_key, kết quả có được chính là giá trị thủy vân của bộ đang xét.

$$r_i.V_A = \text{RSA}(W_i, \text{Private\_key})$$

- Tạo thuộc tính ảo chứa giá trị thủy vân

Lược đồ dữ liệu đang xét sẽ được chèn thêm 1 thuộc tính ảo để chứa các giá trị thủy vân được tạo ra ở bước trên. Như vậy với mỗi bộ dữ liệu sẽ có 1 giá trị thủy vân tương ứng duy nhất.

**Ở giai đoạn trích xuất thủy vân (hình 2):**

- Giải mã giá trị thuộc tính ảo bằng thuật toán mã hóa RSA với khóa Public_key

Khi cần kiểm tra 1 lược đồ dữ liệu, chúng ta sẽ sử dụng thuật toán mã hóa RSA kết hợp với Public_key để giải mã từng giá trị của cột chứa giá trị thủy vân $V_A$. Giá trị nhận được chính là giá trị băm đặc trưng của từng bộ dữ liệu.

- Trích xuất đặc trưng dữ liệu từ R'

Với phần dữ liệu sau khi tách giá thuộc tính VA chúng ta tiến hành thực hiện trích xuất lại đặc trưng dữ liệu như quá trình thủy vân.

$$S'_i = \text{user\_name}\| r'_i.PK\| r'_i.A_1\| r'_i.A_2\|...\| r'_i.A_N\|\text{num\_record\_of\_user\_name}$$

- Tạo giá trị băm từ đặc trưng dữ liệu của R' từ hàm băm SHA-1

$$W'_i = \text{SHA-1}(S'_i)$$

- So sánh 2 giá trị băm, xác thực an toàn dữ liệu

Sau khi có giá trị $W'_i$ và $W_i$ của từng bộ dữ liệu, chúng ta tiến hành so sánh các cặp chuỗi tương ứng này với nhau, nếu chúng giống nhau hoàn toàn thì chúng ta có thể kết luận dữ liệu an toàn, ngược lại dữ liệu đã bị thay đổi và vị trí khác nhau chính là vị trí đã bị sửa đổi.

Trong phương pháp của chúng tôi thuộc tính ảo $V_A$ mang giá trị tổng hợp các giá trị khác trên cùng một bộ, nên khi có sự thay đổi bất kỳ giá trị nào đều làm thay đổi giá trị của cột $V_A$.

Khi áp dụng cụ thể lược đồ thủy vân ở trên vào hệ thống quản lý điểm, mỗi user sẽ được sinh một cặp khóa Private_key và Public_key, khóa Private_key được dùng mã hóa giá trị cho cột $V_A$, khóa Public_key dùng để giải mã giá trị cột $V_A$. Khóa Public_key sẽ được lưu lại trong hệ thống nhằm hỗ trợ việc kiểm tra dữ liệu sau này.

Một user khi nhập điểm vào hệ thống thì user đó sẽ dùng khóa Private_key để khóa giá trị thủy vân ở cột $V_A$ tương ứng với bộ dữ liệu nhập vào. Như vậy, trường hợp ngoài user là chủ sở hữu, có những user khác có quyền sửa đổi với dữ liệu đã thủy vân thì sao? Lúc này, tương ứng với bộ dữ liệu đang được chỉnh sửa sẽ được thủy vân lại với Private_key của user_name thực hiện thao tác đó và đây được coi là hợp lệ trong hệ thống. Lược đồ áp dụng cho thấy sự linh hoạt trong việc thủy vân với nhiều key khác nhau trên cùng 1 lược đồ quan hệ.

Để kiểm tra lại dữ liệu có an toàn hay không, với user có quyền kiểm tra chung (ví dụ: user admin) sẽ xem xét những user_name nào là được phép cập nhật dữ liệu thì xem đó là hợp lệ và dữ liệu vẫn an toàn, ngược lại thì dữ liệu không an toàn và chỉ ra vị trí bị thay đổi. Đối với user là chủ sở hữu của dữ liệu sẽ kiểm tra được dữ liệu mình nhập vào có an toàn hay không, nếu không thì xác định đã bị thay đổi ở vị trí nào và xác định được trường hợp đó là hợp lệ hay không hợp lệ.

Từ lược đồ áp dụng cho thấy để có được giá trị tại ô $V_A$ cần phải có Private_key tương ứng của user. Giả sử rằng dữ liệu bị can thiệp sửa đổi mà không thông qua phần mềm như: thêm, sửa, xóa giá trị của các bản ghi, lúc này giá trị của thuộc tính thủy vân tương ứng sẽ không thể giống với giá trị tính toán có được tại đây vì không có giá trị Private_key tương ứng.

## IV. ĐÁNH GIÁ KẾT QUẢ THỰC NGHIỆM

Trong phương pháp của chúng tôi thủy vân nhúng vào không làm thay đổi giá trị của dữ liệu, đảm bảo hoàn toàn phát hiện được việc sửa đổi, xác định chính xác đến vị trí bản ghi bị sửa đổi. Tuy nhiên, trong trường hợp xóa đi hoặc thêm mới các bộ dữ liệu thì phương pháp này chỉ có thể chỉ ra được nhóm dữ liệu bị chỉnh sửa theo user là chủ sở hữu tương ứng.

Dữ liệu minh họa là lược đồ quan hệ có 6 bộ, có 3 thuộc tính, trong đó có 2 thuộc tính id_sv, id_mon là khóa ngoại tham chiếu đến 2 bảng khác, các thuộc tính đều có kiểu số.

**Bảng 1.** Cơ sở dữ liệu minh họa

| id_sv | id_mon | diem_mh |
|---|---|---|
| 1 | 4 | 6 |
| 1 | 5 | 9 |
| 1 | 6 | 8 |
| 2 | 4 | 6 |
| 2 | 5 | 9 |
| 2 | 6 | 7 |

**Bảng 2.** Dữ liệu sau khi thủy vân

| id_sv | id_mon | diem_mh | watermark |
|---|---|---|---|
| 1 | 4 | 6 | e70b363b0af58a6036dddae5db4d02ff..... |
| 1 | 5 | 9 | 68680f39e6632413ed3463dc7a2028e0.... |
| 1 | 6 | 8 | 4a5888dfec69fd8ec053cab8bef40d33..... |
| 2 | 4 | 6 | 9a58328f1411803d74b0b0b037c8226b.... |
| 2 | 5 | 9 | 601c89722a9a4187aa369cf1eaf94b80..... |
| 2 | 6 | 7 | a8249f7b0840259917e9fd46ac5b6442..... |

Dữ liệu không bị thay đổi sau khi được thủy vân

**Bảng 3.** Trường hợp dữ liệu bị thay đổi

| id_sv | id_mon | diem_mh | watermark |
|---|---|---|---|
| 1 | 4 | 9 | e70b363b0af58a6036dddae5db4d02ff..... |
| 1 | 5 | 9 | 68680f39e6632413ed3463dc7a2028e0.... |
| 1 | 6 | 8 | 4a5888dfec69fd8ec053cab8bef40d33..... |
| 2 | 4 | 6 | 9a58328f1411803d74b0b0b037c8226b.... |
| 2 | 5 | 9 | 601c89722a9a4187aa369cf1eaf94b80..... |
| 2 | 6 | 7 | a8249f7b0840259917e9fd46ac5b6442..... |

Chỉ chính xác vị trí bị thay đổi

**Bảng 4.** Trường hợp dữ liệu bị thêm mới

| id_sv | id_mon | diem_mh | watermark |
|---|---|---|---|
| 1 | 4 | 6 | e70b363b0af58a6036dddae5db4d02ff..... |
| 1 | 5 | 9 | 68680f39e6632413ed3463dc7a2028e0.... |
| 1 | 6 | 8 | 4a5888dfec69fd8ec053cab8bef40d33..... |
| 2 | 4 | 6 | 9a58328f1411803d74b0b0b037c8226b.... |
| 2 | 5 | 9 | 601c89722a9a4187aa369cf1eaf94b80..... |
| 2 | 6 | 7 | a8249f7b0840259917e9fd46ac5b6442..... |
| 3 | 4 | 6 | c401b28f765a59aa98934841948ec9b9..... |

Chỉ ra nhóm dữ liệu bị thay đổi

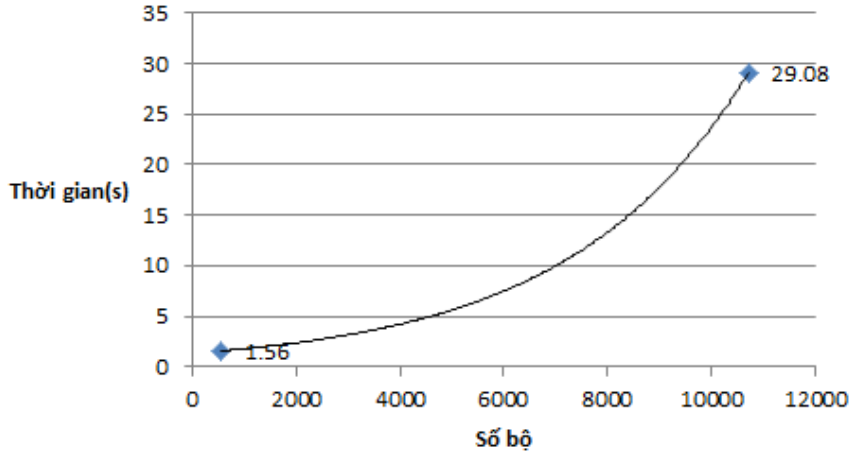
**Bảng 5.** Trường hợp dữ liệu bị xóa

| id_sv | id_mon | diem_mh | watermark |
|---|---|---|---|
| 1 | 5 | 9 | 68680f39e6632413ed3463dc7a2028e0.... |
| 1 | 6 | 8 | 4a5888dfec69fd8ec053cab8bef40d33..... |
| 2 | 4 | 6 | 9a58328f1411803d74b0b0b037c8226b.... |
| 2 | 5 | 9 | 601c89722a9a4187aa369cf1eaf94b80..... |
| 2 | 6 | 7 | a8249f7b0840259917e9fd46ac5b6442..... |

Chỉ ra nhóm dữ liệu bị thay đổi

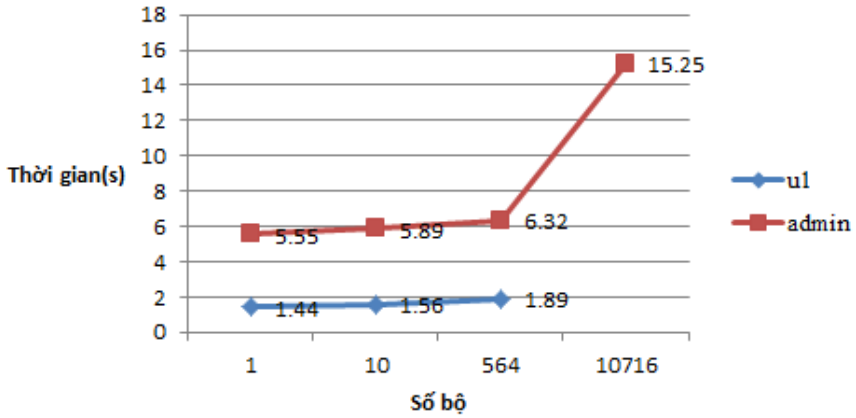
Thực nghiệm với dữ liệu là bảng điểm của một trường Cao đẳng với 10716 bộ dữ liệu, có 3 thuộc tính, trong đó 2 thuộc tính đầu là khóa ngoại tham chiếu đến khóa chính của 2 bảng khác, các thuộc tính đều có kiểu số. Chúng tôi đã thực nghiệm và thống kê được các kết quả sau:

- Thời gian thủy vân đối với từng user nhập điểm (có 564 bộ) là 1.56 giây và đối với admin khi thủy vân toàn bộ bảng điểm (có 10716 bộ) là 29.08 giây (hình 4).



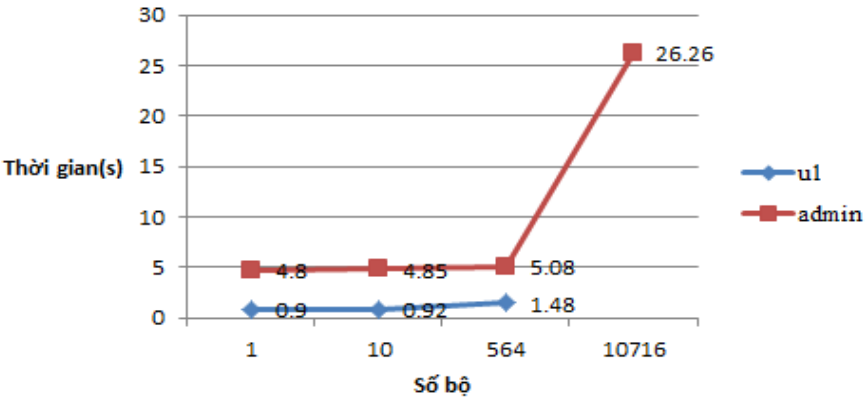

**Hình 4.** Thời gian thủy vân

- Thay đổi dữ liệu: chúng tôi tiến hành thay đổi giá trị các bộ với số bộ bị thay đổi lần lượt là: 1,10,188,10716 và kết quả thống kê được xác suất phát hiện thay đổi giá trị thuộc tính là 100%. Thời gian thực hiện kiểm tra dữ liệu đối với user nhập điểm và user admin được thể hiện trong (hình 5).



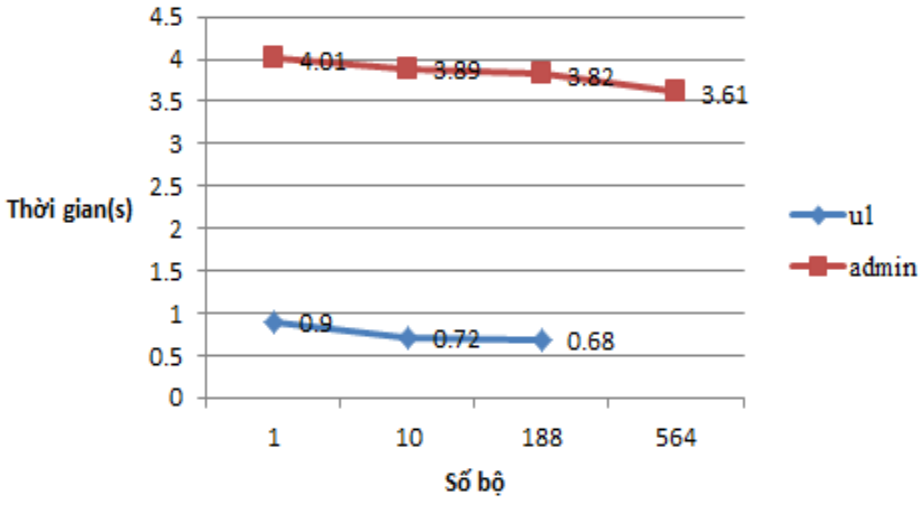

**Hình 5.** Thời gian kiểm tra khi dữ liệu bị sửa đổi đối với 2 user: **u1** và **admin**

- Thêm các bộ mới: chúng tôi tiến hành thêm các bộ mới với số lượng lần lượt là: 1,10,564,10716 và kết quả thống kê được xác suất phát hiện thay đổi là 100%. Thời gian thực hiện kiểm tra dữ liệu đối với user nhập điểm và user admin được thể hiện trong (hình 6).



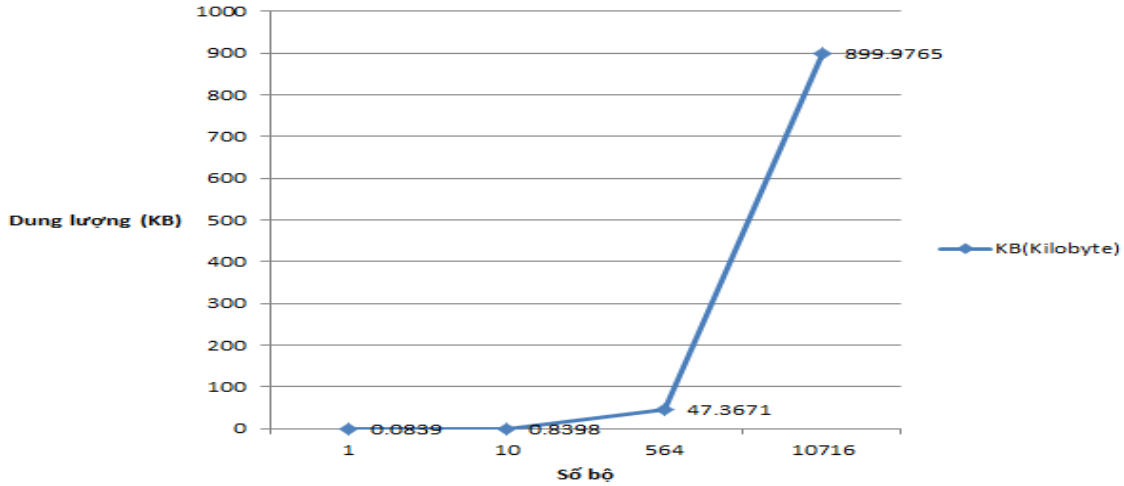

**Hình 6.** Thời gian kiểm tra khi dữ liệu bị thêm mới đối với 2 user: **u1** và **admin**

- Xóa các bộ: chúng tôi tiến hành xóa các bộ mới ngẫu nhiên với số lượng lần lượt là: 1,10,564 và kết quả thống kê được xác suất phát hiện thay đổi là 100%. Thời gian thực hiện kiểm tra dữ liệu đối với user nhập điểm và user admin được thể hiện trong (hình 7).




**Hình 7.** Thời gian kiểm tra khi dữ liệu xóa đối với 2 user: **u1** và **admin**

- Chi phí về dung lượng lưu trữ thủy vân: trong lược đồ thủy vân đề xuất chúng tôi sử dụng hàm băm SHA-1, sau đó mã hóa bằng RSA nên giá trị $V_A$ tốn dung lượng giống nhau 688bit. Trong khi thực nghiệm chúng tôi lần lượt chèn thủy vân vào với số bộ tương ứng là 1,10,564, 10716, kết quả thể hiện trong (hình 8).




**Hình 8.** Chi phí dung lượng lưu trữ thủy vân

Với kết quả thực nghiệm trong các trường hợp cho thấy khả năng phát hiện thay đổi dữ liệu cao, thời gian thủy vân đối với từng user là rất nhỏ, không cảm nhận được, thời gian kiểm tra thủy vân có thể chấp nhận được (10716 bộ, tốn 15.25s), chi phí lưu trữ thủy vân thấp, không đáng kể.

## V. KẾT LUẬN

Kỹ thuật thủy vân trên sơ sở dữ liệu quan hệ đã được nghiên cứu trong thời gian dài với nhiều phương pháp khác nhau và áp dụng đối với các loại dữ liệu khác nhau. Tuy nhiên, để áp dụng vào lĩnh vực cụ thể cần xem xét phương pháp phù hợp với đặc trưng của dữ liệu trong lĩnh vực đó. Bài báo cơ bản khái quát được kỹ thuật thủy vân trên cơ sở dữ liệu quan hệ có các thuộc tính số, từ đó đề xuất mô hình thủy vân có thể áp dụng cho dữ liệu điểm số tại các trường học.

Mô hình đề xuất áp dụng thuật toán đơn giản, không làm thay đổi giá trị các thuộc tính trên dữ liệu được nhúng, hỗ trợ kiểm tra thủy vân dễ dàng trong hệ thống quản lý nhiều người dùng, quá trình phát hiện thủy vân không cần đến dữ liệu gốc và thủy vân gốc. Qua thực nghiệm cho thấy mô hình này có xác suất phát hiện được các tấn công thay đổi rất cao. Từ đó cho thấy mô hình có thể áp dụng cho các hệ thống quản lý điểm, đảm bảo an toàn và xác thực tính toàn vẹn đối với dữ liệu điểm số tại các trường học.

Hướng phát triển của chúng tôi là kết hợp xử lý song song để tối ưu thời gian tính toán, nghiên cứu các phương pháp bảo mật thủy vân mà không phụ thuộc vào các thuật toán mã hóa.

## TÀI LIỆU THAM KHẢO

[1] Agrawal, R. and Kiernan, J. (2002). "Watermarking relational databases". In Proceedings of the 28th international conference on Very Large Data Bases (VLDB '02), pages 155–166, Hong Kong, China. VLDB Endowment.

[2] Agrawal, R., Haas, P. J., and Kiernan, J. (2003), "A system for watermarking relational databases". In Proceedings of the 2003 ACM SIGMOD international conference on Management of data (SIGMOD '03), pages 674–674, San Diego, California. ACM Press.

[3] Agrawal, R., Haas, P. J., and Kiernan, J. (2003), "Watermarking relational data: framework, algorithms and analysis". The VLDB Journal, Volume 12, Pages 157–169.

[4] Li, Y. and Deng, R. H. (2006), "Publicly verifiable ownership protection for relational databases". In Proceedings of the 2006 ACM Symposium on Information, computer and communications security (ASIACCS '06), pages 78–89, Taipei, Taiwan. ACM Press.

[5] Guo, H., Li, Y., Liua, A., and Jajodia, S. (2006), "A fragile watermarking scheme for detecting malicious modifications of database relations". Information Sciences Vol.176, No.10, pp.1350-1378.

[6] Prasannakumari, V., A robust tamperproof watermarking for data integrity in relational databases. Research Journal of Information Technology, 2009. 1(3): 115-121.

[7] Sukriti Bhattacharya and A. Cortesi, Database authentication by distortion-free watermarking, In Proceedings of the 5th International Conference on Software and Data Technologies (ICSOFT '10),pages 219– 226,2010

[8] A. Odeh and A. Al-Haj "Watermarking Relational Database Systems", IEEE, pp. 270-274, 2008.

[9] Gore, Ranjana Waman, and Rucha Tare. "Database Watermarking Using SHA 512 Signature Generation Technique.", 2014.

[10] Brijesh B. Mehta, Udai Pratap Rao," A Novel approach as Multi-place Watermarking For Security in Database"Dept. of Computer Engineering, S. V. National Institute of Technology, Surat, Gujarat.

[11] Sonupriya, P. S., and M. R. Rani. "THE DIGITAL WATERMARKING TECHNIQUE FOR NUMERICAL RELATIONAL DATABASES." (2014).

# APPLY WATERMARKING TECHNIQUE TO PROTECT SCORE STORAGE DATABASE IN UNIVERSITY

**Chung Nam Phong, Tran Cao De**

***ABSTRACT:*** *In modern world, it is generally agreed that applying information technology in universities, colleges and other lower education seems to produce desired effect. Score storage database is one of the most important database that is managed by school management software. Studies have shown that these database as well as management software would have to allow many kind of users, for instance, lecturer or school officer, to access and update. In addition, database protection function is built into the software and database management by permission and role based access, log in or update log. However, it is unsafe to give the management software and system access authority to the third party, who are software developers and there is no strict control on them. As a result, it run the risk of being directly accessed into the database from the database management system. Consequently, a great deal is being written and said about how we can detect invalid modified data which means modifying without management software. Similarly, how a lecturer or officer can confirm a transcript is original and is not invalid modified.*

*To answer these questions, we begin by taking a closer look at watermarking. It is a well-known fact that watermarking is applied for authentic origin and copyright protection on image files for a long time. This technique is a combination of encoding and data hiding aim to protect the copyright of the database. Against this background, the central question that motivates this paper is to present the researches on applying watermarking approach to protect relational database such as marks to provide safety to score storage database in universities, colleges as well as other lower education.*