Đảm bảo tính riêng tư và chống thông đồng trong khai thác luật kết hợp trên dữ liệu phân tán tán ngang

Collusion-Resistant Privacy-Preserving Association Rules Mining on Horizontally Distributed Data

Trần Quốc Việt, Cao Tùng Anh, Lê Hoài Bắc

Abstract: *In this paper, we use the encryption technology to build a new protocol, compute the global support of itemsets in the horizontal distributed database, ensure the privacy in semi - honest environment and have anti - collusion capability, have running time in linear base on the number of parties in the system. We also improved the mining algorithm based on dynamic bit string structure, and combined with the protocol of computing global support built to use on horizontal distributed data, ensure privacy and have high level of anti-collusion.*

Keywords*: Privacy - preserving, collusion, frequent itemset, horizontal distributed.*

I. GIỚI THIỆU

Những tri thức tiềm ẩn được rút trích từ quá trình khai thác dữ liệu có ý nghĩa quan trọng đối với hệ thống quyết định của các tổ chức. Tuy nhiên, quá trình khai thác dữ liệu cũng có thể làm tiết lộ những thông tin nhạy cảm, bất lợi cho tổ chức. Lo ngại này sẽ ngăn cản việc cung cấp dữ liệu của những người sở hữu, vì vậy cần phải giải quyết vấn đề đảm bảo riêng tư một cách hiệu quả.

Tuỳ thuộc vào kiểu cấu trúc dữ liệu mà có những kỹ thuật đảm bảo tính riêng tư khác nhau tương ứng. Hiện tại có hai kiểu bố trí dữ liệu đã và đang được nghiên cứu: CSDL tập trung và CSDL phân tán.

- Với kiểu dữ liệu tập trung, các CSDL được tập hợp về một CSDL duy nhất. Lúc đó phải đảm bảo tính riêng tư của dữ liệu trước khi nó được công bố. Kỹ thuật thường dùng trong trường hợp này là sửa đổi dữ liệu, CSDL phải được sửa đổi sao cho không ai có thể biết nội dung thực sự của dữ liệu, tuy nhiên các thuật toán khai thác có thể rút ra những kết quả gần đúng trên trên dữ liệu đã thay đổi này.
- Với kiểu dữ liệu phân tán, CSDL được xem như gồm nhiều CSDL con, mỗi CSDL con được sở hữu riêng tư bởi mỗi thành viên trong hệ thống, các thành viên hợp tác xử lý để đạt được kết quả giống như khi thực hiện trên một CSDL hợp nhất, trong khi đảm bảo tính riêng tư cho từng CSDL con. Kỹ thuật thường dùng trong tình huống này là tính toán đa bên an toàn, một giao thức tính toán an toàn giữa m bên cho phép tính toán một hàm với m giá trị đầu vào $f(x_1, x_2, \ldots, x_m)$, trong đó mỗi x_i thuộc sở hữu riêng tư của một bên S_i và mỗi S_i không có bất kỳ thông tin nào của các bên ngoài x_i và kết quả cuối cùng của giao thức.

Về cơ bản có hai kiểu phân tán dữ liệu:

- *Phân tán ngang*: Các CSDL con có cùng lược đồ và có tập các giao tác độc lập.
- *Phân tán dọc*: Các CSDL con có cùng tập giao tác nhưng khác nhau tập các thuộc tính.

Hầu hết các thuật toán khai thác luật kết hợp, đảm bảo riêng tư trên dữ liệu phân tán ngang hiện có thường giả định trong môi trường Semi-Honest (SH), nghĩa là tất cả các bên trong hệ thống phải thực hiện theo đúng những giao thức đã được định trước, nhưng

có thể sử dụng các kết quả trung gian và kết quả cuối củng để suy luận thông tin riêng tư [5], [8], [11], [12]. Tuy nhiên, những thuật toán này chưa thực sự ngăn chặn khả năng thông đồng có thể xảy ra.

Trong bài báo này, chúng tôi nghiên cứu giải pháp cho vấn đề đảm bảo tính riêng tư trong khai thác luật kết hợp trên dữ liệu phân tán ngang với kỹ thuật tính toán đa bên an toàn. Cụ thể, chúng tôi vận dụng giao thức tính tích của hai tổng an toàn (SPoS: Secure Product of Summations) của Bin Yang trong [13] (2010) để xây dựng một giao thức mới, cho phép tính độ hỗ trợ toàn cục của các itemset, đảm bảo riêng tư và có khả năng chống thông đồng hoàn toàn. Chúng tôi cũng áp dụng giao thức mới vào thuật toán khai thác tập phổ biến dựa trên chuỗi bit động [1], đảm bảo tính riêng tư trong môi trường SH, trên CSDL phân tán ngang.

II. CÁC NGHIÊN CỨU LIÊN QUAN

II.1. Khai thác luật kết hợp phân tán.

Giả sử có m bên S_1, S_2, …, S_m, mỗi bên sở hữu một CSDL giao tác ${}^iDB$ riêng, các CSDL ${}^iDB$ được xem như phân mảnh ngang, nghĩa là có cùng một lược đồ và có dữ liệu độc lập. Tập các items: $I = \{i_1, i_2, \ldots, i_n\}$ giống nhau giữa tất cả các bên. Mỗi ${}^iDB$ chứa tập các giao tác ${}^iT = \{{}^it_1, {}^it_2, ..., {}^it_{k_i}\}$, trong đó mỗi giao tác ${}^it_j$ là một tập con khác rỗng của I. Mỗi tập con X khác rỗng của I được gọi là một Itemset. Kí hiệu $|{}^iX|$ và $|X|$ lần lượt là số lượng giao tác trong CSDL ${}^iDB$ và CSDL $DB = \{{}^1DB \cup {}^2DB \cup \ldots \cup {}^nDB\}$ có chứa X.

Độ phổ biến cục bộ của X trong S_i, kí hiệu $\sigma({}^iX)$, là tỷ lệ số giao tác trong CSDL ${}^iDB$ có chứa *X* so với tổng số giao tác hiện có CSDL ${}^iDB$.

$$\sigma({}^iX) = \frac{|{}^iX|}{|{}^iDB|}$$

Độ phổ biến toàn cục của X, kí hiệu $\sigma(X)$ là tỷ lệ số giao tác có trong CSDL $DB = {}^1DB \cup {}^2DB \cup \ldots \cup {}^nDB$ chứa X so với tổng số giao tác trong DB.

$$\sigma(X) = \frac{\sum_{i=1}^{m} |{}^iX|}{|\bigcup_{i=1}^{m} {}^iDB|}$$

X - được gọi là tập phổ biến cục bộ tại S_i nếu $\sigma({}^iX) \geq$ *minsupport* và được gọi là phổ biến toàn cục nếu $\sigma(X) \geq$ *minsupport* (*minsupport* là ngưỡng độ phổ biến tối thiểu được định trước bởi người dùng).

Tìm ra tất cả các tập phổ biến là bước quan trọng nhất của quá trình khai thác luật kết hợp, vì vậy vấn đề được giải quyết là tính độ phổ biến toàn bộ $\sigma(X)$ của itemsets X trong khi bảo mật nội dung của các CSDL con cũng như bảo mật độ phổ biến cục bộ của X tại mỗi S_i. Cheung [4] (1996) đã đề xuất thuật toán cho phép khai thác nhanh luật kết hợp trên dữ liệu phân tán ngang gọi là FDM. Tuy chưa thực sự quan tâm đến vấn đề đảm bảo riêng tư nhưng nó có ảnh hưởng nhiều đến các thuật toán sau này

II.2. Một số công cụ tính toán đa bên an toàn

- ***Định nghĩa:*** Một giao thức được cho là giảm mức độ riêng tư g đến f nếu tồn tại một tính toán riêng tư g khi sử dụng f. Khi đó, ta nói rằng g có thể giảm mức độ riêng tư đến f [13].
- ***Định lý (tổng hợp):*** Giả sử g có thể giảm riêng tư đến f và tồn tại một giao thức tính toán riêng tư f thì cũng tồn tại một giao thức tính toán riêng tư g [13].
- ***Hệ mã hóa đồng cấu (Homomorphic encryption)***

Hệ mã hóa có tính chất đồng cấu được sử dụng nhiều trong các giao thức tính toán đa bên an toàn. Một hệ mã hóa công khai với hàm mã hóa $E_{pk}(.)$ có tính chất đồng cấu nếu với mọi thông điệp (bản rõ) m_1, m_2, ta luôn có:

$$E_{pk}(m_1 +_M m_2) = E_{pk}(m_1) +_C E_{pk}(m_2)$$

Trong đó: $+_M$ là phép toán hai ngôi định nghĩa trên không gian bản rõ (*plaintext space*) và $+_C$ là phép toán

hai ngôi định nghĩa trên không gian bản mã (*ciphertext space*)

Các hệ mã RSA, EL Gamal, Paillier,.. đều có tính chất đồng cấu. Dựa trên tính chất đồng cấu, ta có thể thực hiện những tính toán trên các bản rõ mà không cần giải mã chúng.

- **Hệ mã hóa giao hoán**

Một hệ mã hóa khóa công khai **E** với không gian bản rõ M, không gian bản mã C và không gian khóa K được gọi là có tính giao hoán nếu với mọi bản rõ m, mọi bộ n khóa $k_1,k_2, ..,k_n$ ($k_i \in K$) và một hoán vị bất kỳ của i, j ta luôn có:

$$E_{k_{i_1}}(...E_{k_n}(m)...) = E_{k_{j_1}}(...E_{k_{j_n}})...)$$

Nghĩa là thứ tự mã hóa và giải mã là không quan trọng.

Một ứng dụng của tính chất hoán vị của hệ mã hóa là thực hiện phép hợp đảm bảo riêng tư [5], [6], [15], [16].

- ***Giao thức tính tích của hai tổng an toàn***

Ngoài các giao thức tính toán đa bên an toàn như: tính tổng, so sánh, phép hợp, tính lực lượng của phần giao,…, được trình bày trong [5], [6], [7], vận dụng tính chất đồng cấu của hệ mã hóa, Bin Yang cùng các đồng sự đã đề xuất giao thức tính tích của hai tổng an toàn SPoS [13] (2010).

Giả sử có m bên $S_1, S_2,…, S_m$, mỗi S_i sở hữu hai số thực ${}^{i}x_1$ và ${}^{i}x_2$ ($0<{}^{i}x_1, {}^{i}x_2<1$), giao thức SPoS cho phép mỗi bên tính được tích:

$$P = \sum_{i=1}^{m} {}^{i}x_1 \times \sum_{i=1}^{m} {}^{i}x_2$$

Trong khi vẫn giữ bí mật các giá trị riêng tư mỗi bên. Giao thức này đã được tác giả chứng minh là đảm bảo riêng tư và có khả năng chống thông đồng hoàn toàn. Tuy tác giả chưa đề cập đến việc sử dụng giao thức này trong khai thác luật kết hợp, nhưng trong phần sau, chúng tôi sẽ vận dụng giao thức này để xây dựng giao thức tính độ phổ biến toàn cục cho các itemsets trên dữ liệu phân tán ngang, đảm bảo riêng tư.

II.3. Một số thuật toán đã có

Kỹ thuật sửa đổi dữ liệu gốc thường cho kết quả gần đúng và áp dụng chủ yếu trên CSDL tập trung. Đối với CSDL phân tán, tính riêng tư thường được đảm bảo thông qua kỹ thuật tính toán đa bên an toàn.

- Thuật toán SFDM [5]: Được đề xuất bởi Murat Kantarcioglu và Chris Clifton (2004), là sự cải tiến của thuật toán FDM trong [4] nhằm đảm bảo tính riêng tư. Tác giả đã áp dụng tính chất giao hoán của hệ mã hóa RSA để ẩn danh nguồn gốc của các itemsets trong quá trình trao đổi. Để bảo vệ độ phổ biến cục bộ của itemset X, mỗi S_i sử dụng độ hỗ trợ vượt ngưỡng ${}^{i}X_{excess}=|{}^{i}X| - s\%\times|{}^{i}DB|$ thay cho độ hỗ trợ cục bộ. S_1 phát sinh số bí mật R_1, tính $v_1 = {}^{1}X_{excess} + R_1$ và gởi v_1 đến S_2, S_2 tính $v_2 = v_1 + {}^{2}X_{excess}$ và gởi kết quả đến S_3,…, sau khi tính $v_m = v_{m-1} + {}^{m}X_{excess}$, S_m thực hiện phép so sánh riêng tư v_m với R_1 trong S_1. Nếu $v_m \geq R_1$ thì X là phổ biến toàn cục. Thuật toán SFDM chỉ an toàn khi không có sự thông đồng trong hệ thống, cụ thể nếu S_{i-1} và S_{i+1} thông đồng với nhau thì có thể suy ra giá trị riêng của S_i.

- Thuật toán CRDM [8]: Được đề xuất bởi Urabe (2007). Ý tưởng chính của thuật toán là thực hiện phép tính tổng an toàn, bằng cách chia nhỏ mỗi giá trị riêng tư đến một số bên khác nhau trong hệ thống, để từ đó nhận được kết quả cuối cùng. Tác giả đã chứng minh rằng, tính riêng tư trong thuật toán CRDM có thể bị vi phạm khi có ít nhất m-2 bên thông đồng với nhau.

- Thuật toán của Vladimir Estivill-Castro [11] (2007): Tác giả chỉ sử dụng kỹ thuật mã hóa khóa công khai để chia sẻ dữ liệu ẩn danh mà không cần sử dụng đến tính chất giao hoán của hệ mã. Thuật toán sử dụng một bên đặc biệt (S_1) để khởi tạo, phát sinh và phân phối khóa mã hóa cho tất cả các bên còn lại trong hệ thống. S_1 mã hóa dữ liệu của mình và gởi đến S_2, lần lượt các

S_i (i = 2, 3 ,…, m) mã hóa dữ liệu hiện có và trộn với dữ liệu nhận được từ S_{i-1}, sau khi loại bỏ các bộ thừa, gởi kết quả đến $S_{(i \bmod m)+1}$. Kết thúc thuật toán, S_1 có tập dữ liệu đầy đủ nhưng đã được làm mờ nguồn gốc sở hữu. Do quá trình truyền thông chỉ sử thực hiện trên một vòng lặp duy nhất (để mã hóa), việc giải mã chỉ thực hiện cục bộ tại S_1 nên thời gian chạy của thuật toán nhanh hơn các thuật toán sử dụng tính chất giao hoán của hệ mã hóa khóa công khai đã có trước đó. Tuy nhiên, mức độ đảm bảo riêng tư của thuật toán này phụ thuộc hoàn toàn vào bên khởi tạo.

▪ Thuật toán của Mahmoud Hussein [12] (2008): Tác giả đã cải tiến thuật toán trong [11] bằng cách tái sắp xếp vai trò của các bên nhằm tăng mức độ an toàn cũng như thời gian chạy. Thuật toán đã sử dụng hai bên đặc biệt gọi là **Initiator** và **Combiner**, Initiator có nhiệm vụ khởi tạo khóa, tính toán kết quả cuối cùng, Combiner có nhiệm vụ sưu tập dữ liệu từ các client, xáo trộn và trao đổi với đối với **Initiator**. Mức độ đảm bảo riêng tư của thuật toán này phụ thuộc vào khả năng thông đồng có thể xảy ra giữa Initator và Combiner.

Hầu hết những thuật toán hiện có đều chưa thực sự an toàn khi có sự thông đồng xảy ra trong hệ thống. Vấn đề này sẽ được giải quyết trong thuật toán mới của chúng tôi được trình bày trong phần tiếp theo.

III. GIẢI THUẬT KHAI THÁC TẬP PHỔ BIẾN ĐẢM BẢO RIÊNG TƯ VÀ CHỐNG THÔNG ĐỒNG TRÊN DỮ LIỆU PHÂN TÁN NGANG

III.1. Giao thức đảm bảo tính riêng tư trong tính độ phổ biến toàn cục

Để xây dựng giao thức tính độ phổ biến toàn cục, trước hết, chúng tôi giả định rằng tất cả m bên đều biết một số nguyên A thỏa điều kiện A ≥ max {$|^1DB|$, $|^2DB|$, …, $|^mDB|$}, việc tiết lộ giá trị A như vậy không làm ảnh hưởng lớn đến tính riêng tư, tuy nhiên trong phần sau, chúng tôi sẽ đề xuất một giao thức chọn A an toàn.

Đặt: $^ix = \dfrac{|^iDB|}{A+\varepsilon}$ và $^iy = \dfrac{(|^iX|+\varepsilon)}{A+\varepsilon}$

Với ε là số thực rất bé được biết trước bởi tất cả các bên, (ε≤ 1, trong thực tế ta có thể chọn ε = 1). Khi đó ta có:

$$0 < {}^ix < 1 \text{ và } 0 < {}^iy < 1$$

Mỗi bên S_i phát sinh một số thực ngẫu nhiên $^ic \in (0,1)$. Áp dụng giao thức tính tích của hai tổng đảm bảo riêng tư SPoS trong [13], ta tính được:

$$p_1 = ({}^1x + {}^2x + ... + {}^mx)({}^1c + {}^2c + ... + {}^mc) \quad (3.1)$$

$$p_2 = ({}^1y + {}^2y + ... + {}^my)({}^1c + {}^2c + ... + {}^mc) \quad (3.2)$$

Chia (3.2) cho (3.1) ta được:

$$\frac{P_2}{P_1} = \frac{\sum_{i=1}^{m} {}^ix}{\sum_{i=1}^{m} {}^iy} = \frac{\sum_{i=1}^{m} |^iX| + m\varepsilon}{|\sum_{i=1}^{m} |^iDB|} \quad (3.3)$$

Trong khai thác dữ liệu, m (số lượng các bên) nhỏ hơn rất nhiều so với số lượng giao tác trong CSDL, hơn nữa ε rất bé (ε≤ 1), nên thành phần mε có thể bỏ qua trong công thức (3.3). Do vậy ta có:

$$\frac{P_1}{P_2} = \frac{\sum_{i=1}^{m} |^iX| + m\varepsilon}{|\sum_{i=1}^{m} |^iDB|} \approx \frac{\sum_{i=1}^{m} |^iX|}{|\sum_{i=1}^{m} |^iDB|} = \sigma(X) \quad (3.4)$$

Công thức (3.4) cho ta độ phổ biến toàn cục của itemset X.

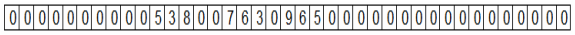
III.2. Cấu trúc chuỗi bit động

Theo tác giả trong [1], dữ liệu liên quan đến mỗi tập mục dữ liệu được lưu trữ bởi một chuỗi bit động (DBS: Dynamic Bit String). Mỗi DBS cho một tập mục dữ liệu được biểu diễn bởi 2 thành phần:

- Ppos: lưu vị trí của byte khác không đầu tiên trong chuỗi bit .

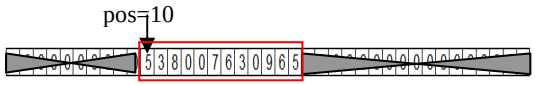
- Bitlist: dãy các byte của chuỗi bit sau khi đã loại bỏ các byte bằng không ở đầu và ở cuối chuỗi bit.

Ví dụ 1: Giả sử có chuỗi bit (được tính ra giá trị thập phân) như sau:



Khi đó: pos = 10

Bitlist = {5,3,8,0,0,7,6,3,0,9,6,5}



DBS

Hình 1. Minh họa về DBS

Số lượng bit 1 trong DBS chính là độ hỗ trợ của tập mục dữ liệu tương ứng. Việc loại bỏ những bit 0 ở đầu và cuối chuỗi bit sẽ tiết kiệm đáng kể bộ nhớ để lưu trữ CSDL trên bộ nhớ trong trong quá trình xử lý..

Tác giả cũng đã đề xuất kỹ thuật tính nhanh độ phổ biến của tập mục XY dựa trên hai DBS biểu diễn tương ứng cho hai tập mục X, Y. Tập phổ biến tìm được được lưu trữ bởi cấu trúc cây gọi là FITree, mỗi nút trong cây gồm hai thành phần:

- Tập mục dữ liệu X

- Chuỗi bit động (DBS) biểu diễn cho tập mục dữ liệu X

III.3. Giải thuật khai thác tập phổ biến

Để tiết kiệm bộ nhớ và tăng tốc độ xử lý cục bộ tại mỗi S_i, chúng tôi chọn thuật toán khai thác tập phổ biến dựa trên cấu trúc chuỗi bit động [1], kết hợp với giao thức tính độ phổ biến toàn cục được xây dựng trong phần III.1 để có thể áp dụng trên CSDL phân tán ngang, đảm bảo riêng tư.

Mỗi S_i sử dụng một FITree để lưu trữ tập phổ biến toàn cục, mỗi mỗi nút trong FITree gồm các thông tin: Itemset X, DBS(X), $\sigma(^iX)$ và $\sigma(X)$.

Kí hiệu:

^iLL_k:Tập itemset phổ biến cục bộ tại S_i trong lần duyệt thứ k.

C_k: Tập ứng viên toàn cục ở lần duyệt thứ k.

^iBT: CSDL của S_i được nén, mỗi phần tử của ^iBT gồm 4 phần: Itemsret X, DBS(X) và độ phổ biến cục bộ $\sigma(^iX)$ ở S_i và độ phổ biến toàn cục $\sigma(X)$.

Các bên S_i (i=1,2,.., m) cùng thực hiện song song thủ tục CREATE_FITREE được mô tả sau đây để có được FITree toàn cục.

CREATE_FITREE(^iDB, {S_j , j = 1 ,2,…, m})

1. *Áp dụng cấu trúc chuỗi bit động, nén CSDL ^iDB của S_i vào ^iBT ;*
2. *A = **UPPER_BOUND**($|^iDB|$});*
3. *Phát sinh ngẫu nhiên số thực $^ic \in (0, 1)$;*
4. $P_1 = SPoS(^ic, \frac{|^iDB|}{A+\varepsilon})$;
5. *L= ∅ // L là tập các Item phổ biến;*
6. k=1;
7. *^iLL_k =Tập phổ biến cục bộ ở S_i ;*
8. *C_k=SECURE_UNION(^iLL_k);*
9. ***For Each** $X \in C_k$.*
10. $\sigma(X) = \text{SECURE\_SUPPORT}(X)$;
11. ***If** $\sigma(X) \geq$ minsup **then***
12. $L = L \cup \{X\}$;
13. ***End for***
14. *Khởi tạo FITree = Empty;*
15. *FITree.Children=L.;*
16. *EXTEND _FITREE(FITree, minsup, 0);*
17. ***Return** FITree;*

SECURE_SUPPORT(X)

18. $P_2 = SPoS(\frac{|X|+\varepsilon}{A+\varepsilon}, ^ic)$;.
19. $\sigma(X) = \frac{P_2}{P_2}$;
20. ***Return** $\sigma(X)$;*

EXTEND _FITREE (FITree, minsup, k).

21. *k=k+1;*
22. ***For** l=1 **To** FITree.Children.Count-1*
23. X_i = *To FITree.Children[i];*

24. *${}^iC_k$=Tập phổ biến cục bộ ở S_i phát sinh từ FITree;*
25. *C_k= SECURE_UNION (${}^iC_k$);*
26. ***End for***
27. ***For*** *j = l+1* ***to*** *FITree.Children.Count*
28. *X_j= FITree.Children[j];*
29. ***If*** *(${}^l(X_i \cup X_j) \in C_k$)* ***then***
30. $P_2 = SPoS(\frac{|^i(X_i \cup X_j)|+\varepsilon}{A+\varepsilon}, {}^ic)$;
31. $\sigma(X_i \cup X_j)$=SECURE_SUPPORT($X_i \cup X_j$);
32. ***If*** $\sigma(X_i \cup X_j) \geq$ *minsup* ***then***
33. X_i*.children.Add($X_i \cup X_j$);*
34. *FITree.DBS=Empty;*
35. ***End if***
36. ***End if***
37. ***End for***
38. ***EXTEND _FITREE*** *(X_i, minsup, k);*

UPPER_BOUND(^{i}DB)

39. *Phát sinh số nguyên ngẫu nhiên r_i;*
40. ***If*** *(i=1)* ***then*** *//**S_1** là master*
41. *Gởi $v_1 = r_1 + |{}^1DB|$ đến S_2.;*
42. *Nhận v_m từ S_m.*
43. *Gởi v_m đến tất cả các S_j ($j \neq i$);*
44. ***Else*** *//S_i không phải là master*
45. *Nhận v_{i-1} từ S_{i-1};*
46. *Gởi v_i=max{v_{i-1},r_i + $|{}^iDB|$} đến $S_{(i \bmod m)+1}$;*
47. *Nhận v_m từ S_1;*
48. ***End if***
49. ***Return*** *v_m;*

Thủ tục SECURE_UNION thực hiện phép hợp an toàn để tỉa tập ứng viên toàn cục, chúng ta có thể sử dụng giải thuật trong [16] để có mức độ đảm bảo riêng tư cao, chống khả năng thông đồng.

Thủ tục SECURE_SUPPORT(X) (dòng 18, 19, 20) là sự cài đặt của giao thức tính độ phổ biến toàn cục của itemset X được xây dựng trong phần III.1. Giao thức SPoS(${}^ix_1$, ${}^ix_2$) trong [13] được vận dụng vào thủ tục để tính giá trị trung gian $P = \sum_{k=1}^{m} {}^kx_1 \sum_{k=1}^{m} {}^kx_2$ trong khi bảo vệ riêng tư các giá trị ${}^ix_1$, ${}^ix_2$.

Sau khi các nút con của nút gốc trong FITree (gồm các 1-itemset phổ biến toàn cục) được tạo (từ dòng 1 đến dòng 15). Thủ tục EXTEND_FITREE được gọi một cách đệ quy để mở rộng và hoàn thiện FITree chứa tập đầy đủ các itemset phổ biến toàn cục (từ dòng 20 đến dòng 38).

Từ tính chất "*Một itemset là phổ biến toàn cục thì phải phổ biến cục bộ ít nhất tại một bên nào đó*" [4], chúng tôi sử dụng phép hợp an toàn (SecureUnion) trong [16] để tìm tập itemset ứng viên trong mỗi bước xử lý (dòng 8 và 25).

Ví dụ: Hình 2 minh họa cho thuật toán với trường hợp cụ thể gồm hai bên S_1, S_2 như sau:

S1

Trans	Items
1	A, B
2	A, C
3	A, B, C
4	B, C
5	A, C, D

S2

Trans	Items
6	C, D
7	A, B, D
8	A, B, C
9	A, B

minsupport=40%

FITree={}

FITree={}

Hình 2. Minh họa hệ thống gồm 2 bên S_1, S_2

- Kết quả của bước nén các CSDL cục bộ (dòng 1) để đưa vào bộ nhớ trong:
  - ✓ Nén CSDL ^{1}DB:

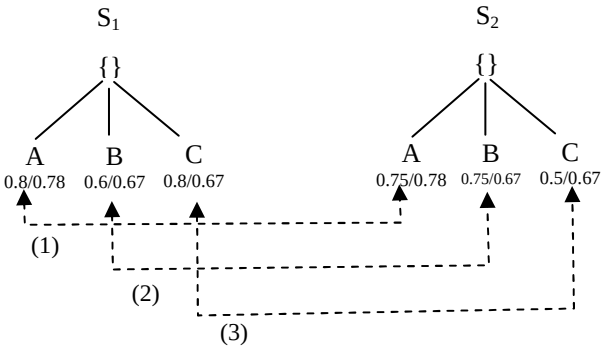
    ^{1}BT={(A,29,?), (B,22,?), (C,15,?),(D,1,?)}
  - ✓ Nén CSDL ^{2}DB:

    ^{2}BT={(A,7,?), (B,7,?), (C,10,?),(D,12,?)}

(Sử dụng kí hiệu ? để biểu diễn cho độ phổ biến toàn cục chưa biết của các itemsets).

- Tạo các nút con của nút gốc của FITree (từ dòng 3 đến dòng 15):
  - ✓ Tập ứng viên toàn cục C_1 = {A, B, C}.

✓ Lần lượt tính độ phổ biến toàn cục các itemset A, B và C (dòng 9, 15), tất cả đều có độ phổ biến toàn cục lớn hơn *minsupport* nên L={A, B, C} là con của nút gốc FITree ở mỗi S_i (kết quả như Hình 3).



Hình 3. Kết quả FITree sau khi xử lý nút gốc

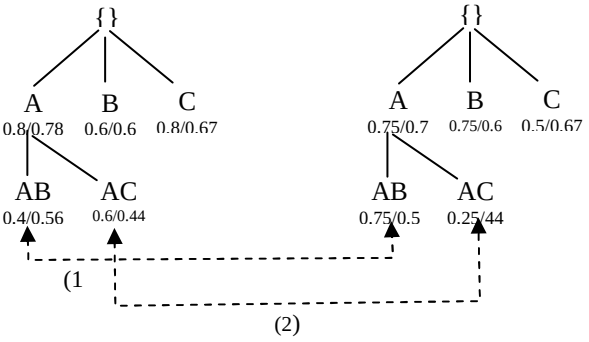
Thủ tục **EXTEND_FITREE để mở rộng và hoàn thiện FITree (từ dòng 20 đến dòng 33)**

- Tạo các nút con cho nút A:

✓ Tập ứng viên toàn cục: C_2 = {AB, AC}

✓ Lần lượt tính độ phổ biến toàn cục cho AB, AC. Cả hai có độ phổ biến toàn cục lớn hơn *minsupport* nên đều là nút con của nút A trong từng FITree trong mỗi S_i (kết quả như Hình 4).

Để tiết kiệm bộ nhớ, huỷ DBS của nút A sau khi xử lý (dòng 32).



Hình 4. Kết quả FITree sau khi xử lý nút A

- Tạo các nút con cho nút AB:

Độ phổ biến toàn cục của itemset ABC trên cả S_1 và S_2 lần lượt là 0.2 và 0.25, cả hai đều nhỏ hơn *minsupport* = 40% nên tập ứng viên toàn cục tương ứng $C_3 = \varnothing$, không có nút con của nút AB.

- Tạo các nút con cho nút B:

✓ Tập ứng viên toàn cục ứng với nút B là C_4={BC}.

✓ Độ phổ biến toàn cục của BC là 0.33 < *minsuppor*t nên không tạo nên nút con của B.

Kết thúc thuật toán, Hình 4 cũng là tình trạng cuối cùng của FITree, tập phổ biến tìm được là tập các itemsets tương ứng với các nút trong FITree.

III.4. Đánh giá thuật toán

a. Đánh giá mức độ bảo vệ riêng tư

Với giả định các hệ mã hóa sử dụng là an toàn về mặt ngữ nghĩa như hệ mã Paillier [9], [13].

- **Mức độ đảm bảo riêng tư của thủ tục SUPPER_BOUND**: Trong thủ tục này, mỗi S_i đều cộng thêm vào $|^iDB|$ của mình một số nguyên ngẫu nhiên r_i trước khi trao đổi với bên khác, do vậy $|^iDB|$ của S_i được đảm bảo riêng tư và cũng chống lại khả năng thông đồng.
- ***Mức độ đảm bảo riêng tư của giao thức tìm tập ứng viên toàn cục (*SECURE_UNION*)***: Chúng tôi sử dụng phép hợp đảm bảo riêng tư trong [16] để tìm tập ứng viên toàn cục. Trong [16], tác giả đã chứng minh giao thức này là an toàn trong cả môi trường malicious và có khả năng chống thông đồng.
- ***Mức độ đảm bảo riêng tư của giao thức tính độ phổ biến toàn cục cho các itemset:*** Tác giả trong [13] đã chứng minh giao thức SPoS là an toàn theo mức độ an toàn của hệ mã hóa sử dụng. Hơn nữa, giao thức SPoS có mức độ bảo vệ riêng tư và chống thông đồng hoàn toàn (full - private). Tuy

nhiên chúng ta cần phải xem xét cụ thể khi áp dụng giao thức này để tính độ phổ biến toàn cục.

Độ phổ biến toàn cục của itemset X được xác định thông qua công thức 3.4. Trong đó mức độ đảm bảo riêng tư trong tính toán giá trị P_1, P_2 được xác định theo hai giao thức SPoS và SUPPER_BOUND (full - private)**.**

Theo định lý tổng hợp [13], để đánh giá mức độ duy trì tính riêng tư của toàn bộ thuật toán, ta xem các giao thức con đảm bảo riêng tư đã dùng như các hộp đen và xem xét mức độ riêng tư của giao thức tính độ phổ biến toàn cục của itemset X. Độ phổ biến toàn cục của X được tính theo công thức .

$$\sigma(X) = \frac{\sum_{i=1}^{m} |^i X|}{\sum_{i=1}^{m} |^i DB|} \qquad (3.5)$$

Xem xét các trường hợp có thể xảy ra sau đây:

- *Trường hợp có ít hơn m - 1 bên thông đồng*: phương trình (3.5) luôn có nhiều hơn 4 biến, do vậy độ phổ biến cục bộ của X trong các bên còn lại vẫn được giữ bí mật.
- *Trường hợp có m - 1 bên* $S_{i1}, S_{i_2}, \ldots, S_{i_{m-1}}$ *thông đồng với nhau*:

- Nếu tất cả $S_{i1}, S_{i_2}, \ldots, S_{i_{m-1}}$ tham gia trong giao thức tính độ phổ biến toàn cục của X với kích cỡ CSDL cũng như độ phổ biến cục bộ bằng 0, ta có:

$$\sigma(X) = \frac{\sum_{i=1}^{m} |^i X|}{\sum_{i=1}^{m} |^i DB|} = \frac{|^{i_m} X|}{|^{i_m} DB|}$$

Khi đó, các $S_{i1}, S_{i_2}, \ldots, S_{i_{m-1}}$ sẽ biết được độ phổ biến cục bộ của X tại S_{i_m}. Tuy nhiên, trong mô hình SH, các bên thực hiện theo đúng giao thức đã được định sẵn, độ phổ biến cục bộ của itemset X có thể bằng 0 nhưng số lượng giao tác trong mỗi CSDL phải lớn hơn (rất nhiều) so với 0. Do vậy, với giả định hệ mã hóa sử dụng là an toàn về mặt ngữ nghĩa [9][13], việc suy luận chính xác độ phổ biến cục bộ của X trong S_{i_m} là không thể.

- Nếu độ phổ biến của X lớn hơn 0 tại ít nhất một bên trong m-1 bên $S_{i1}, S_{i_2}, \ldots, S_{i_{m-1}}$,giao thức tính độ phổ biến toàn cục được xây dựng đảm bảo tính riêng tư hoàn toàn như giao thức SPoS.

Tóm lại, giao thức tính độ phổ biến toàn cục của chúng tôi đảm bảo riêng tư hoàn toàn với môi trường semi-honest.

b. Đánh giá chi phí truyền thông

Trọng tâm của bài báo là xây dựng giao thức tính độ phổ biến toàn cục của tập mục dữ liệu, đảm bảo riêng tư hoàn toàn, thuật toán khai thác tập phổ biến là một áp dụng của giao thức. Để đánh giá sự phụ thuộc của chi phí truyền thông trong giao thức, chúng tôi bỏ qua ảnh hưởng của phép hợp an toàn.

Theo thuật toán được xây dựng, độ phổ biến toàn cục của mỗi itemset X được tính bởi công thức (3.5)

P_1 và giá trị A sử dụng trong thuật toán chỉ được tính một lần trong giai đoạn khởi tạo, do vậy ta chỉ cần đánh giá chi phí truyền thông phát sinh từ việc tính P_2. Mỗi giá trị P_2 được tính tương ứng với một lần thực hiện giao thức SPoS, số lượng thông điệp truyền đi trong hệ thống là: 4m(m-1) với m là số lượng các bên. Tuy nhiên, quá trình trao đổi thông điệp xảy ra song song trên từng cặp hai thành viên độc lập và trong [13] đã chứng minh rằng thời gian chạy của giao thức SPoS chỉ là tuyến tính theo m (O(m)), như vậy thời gian để tính P_2 cũng là O(m), nếu thời gian thực hiện các xử lý cục bộ là như nhau giữa các bên và không xét đến phép hợp an toàn để tỉa tập ứng viên, tổng thời gian chạy của toàn bộ thuật toán tìm tập phổ biến được xây dựng cũng là O(m).

IV. THỰC NGHIỆM

Khai thác luật kết hợp gồm 2 giai đoạn: i) *Tìm tập phổ biến.* ii) *Phát sinh luật kết hợp*. Vấn đề tính riêng tư chỉ tập trung ở giai đoạn i), khi đã có tập phổ biến thì giai đoạn ii) phát sinh luật, giống như các thuật toán truyền thống trước đây. Do đó, thuật toán được xây dựng chỉ khác với các thuật toán truyền thống

trước đây ở giai đoạn tìm tập phổ biến, vì vậy chúng tôi chỉ tiến hành thực nghiệm trên giai đoạn này.

Chúng tôi sử dụng ngôn ngữ lập trình C# trong Visual Studio 2010, sử dụng hệ mã hóa đồng cấu Paillier, kích cỡ khóa 1024 bit để đảm bảo tính riêng tư cho giao thức tính độ phổ biến toàn cục. Mục đích của thực nghiệm là đánh giá thời gian chạy của thuật toán, sự phụ thuộc của thời gian chạy vào số lượng các bên. Thực nghiệm được tiến hành trên CSDL Accidents (http://fimi.cs.helsinki.fi/data/) với 340.183 giao tác, 468 item (35MB) và CSDL bảo hiểm nhân thọ với 393.301 giao tác, 476 item (8 MB).

Để tiến hành thực nghiệm, chúng tôi phân mảnh (ngang) mỗi CSDL tương ứng thành 2, 3, 4 và 5 phần bằng nhau tương ứng với thực nghiệm trên 2, 3, 4 và 5 máy. Cấu hình các máy sử dụng trong các thực nghiệm như sau:

Máy 1:
- CPU: Intel Core I3 M380, 2.53 GHz
- RAM: 3GB
- Card mạng: Atheros AR8152 PCI-E Fast Ethernet Controller

Máy 2:
- CPU: Intel Core 2 Duo T5470, 1.6GHz
- RAM: 2GB
- Card mạng: Intel 82566MM Gigabit Network Connection

Máy 3:
- CPU: Intel Core I3 M370 2.4GHz
- RAM: 4GB
- Card mạng Atheros AR8152 PCI-E Fast Ethernet Controller

Máy 4:
- CPU: Intel Core 2 Duo P8600, 2.4GHz
- RAM: 3GB
- Card mạng: Realtek RTL8168D Family-PCI-E Gigabit Ethernet

Máy 5:
-CPU: Intel Core I3 M370 2.4GHz
- RAM: 2GB
- Card mạng: Atheros AR8152 PCI-E Fast Ethernet Controller

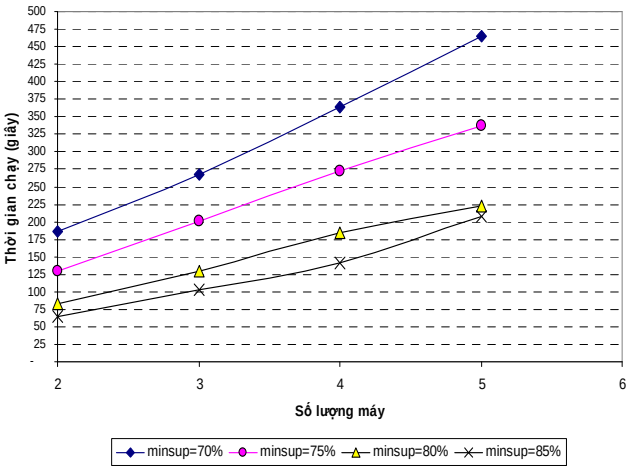
Thời gian chạy của thuật toán phụ thuộc vào máy có cấu hình thấp nhất, do đó trong tất cả các thực nghiệm chúng tôi luôn sử dụng máy 2.

Bảng 1. Thời gian chạy trên CSDL Accidents

Minsupp (%)	Thời gian chạy (giây) theo số lượng máy			
	TN1 (2m)	TN2 (3m)	TN3 (4m)	TN4 (5m)
70	186	268	363	465
75	129	201	273	338
80	83	129	184	223
85	64	103	142	207

Bảng 2. Thời gian chạy trên CSDL Bảo hiểm

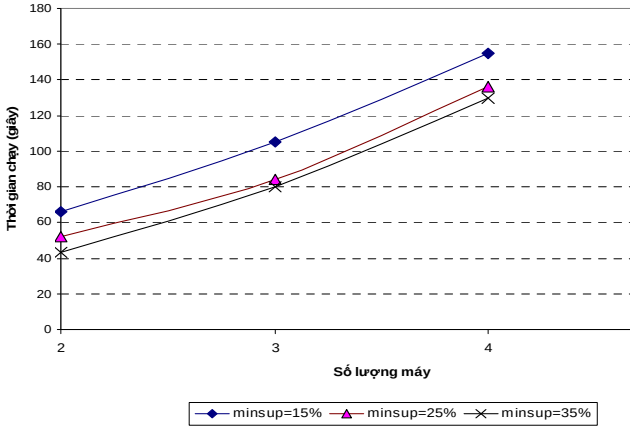
Minsupport (%)	Thời gian chạy (giây) theo số lượng máy			
	TN1 (2m)	TN2 (3m)	TN3 (4m)	TN4 (5m)
70	186	268	363	465
75	129	201	273	338
80	83	129	184	223
85	64	103	142	207



Hình 5. Sự phụ thuộc thời gian chạy vào số lượng máy trên CSDL Accident

Bảng 1 ghi nhận thời gian chạy của thuật toán tìm trên CSDL Accidents và Bảng 2 ghi nhận thời gian chạy trên CSDL bảo hiểm, các đồ thị biểu diễn tương ứng như trong Hình 5 và Hình 6. Thực nghiệm đã kiểm chứng lại lý thuyết cho rằng thời gian chạy của

thuật toán là tuyến tính theo số lượng các bên tham gia vào hệ thống.



Hình 6. Sự phụ thuộc thời gian chạy theo số lượng máy trên CSDL bảo hiểm

V. KẾT LUẬN

Đảm bảo riêng tư là một trong những yêu cầu quan trọng trong quá trình khai khai thác dữ liệu. Với dữ liệu phân tán, kỹ thuật thường dùng là tính toán đa bên an toàn. Tính riêng tư trong nhiều thuật toán trước đây chưa thật sự đảm bảo khi xảy ra thông đồng giữa một nhóm các bên. Dựa trên giao thức của Bin Yang trong việc tính tích hai tổng [13], chúng tôi đã xây dựng một giao thức mới, cho phép tính độ phổ biến toàn cục của itemset, đảm bảo riêng tư trong môi trường SH, có khả năng chống thông đồng trên dữ liệu phân tán ngang. Trong giao thức này, độ phổ biến cục bộ của itemset cũng được bảo vệ ngay cả trong trường hợp hệ thống chỉ gồm 2 bên. Chúng tôi cũng đã cải tiến thuật toán trong [1], kết hợp với giao thức tính độ hỗ trợ được xây dựng để có thể áp dụng được trên CSDL phân tán ngang, đảm bảo riêng tư và có khả năng chống thông đồng.

Chúng tôi đã tiến hành thực nghiệm trên CSDL Accidents và CSDL bảo hiểm trong thực tế. Kết quả thực nghiệm khẳng định lại rằng, dù số lượng thông điệp truyền thông trong quá trình tính độ hỗ trợ của một itemset là $O(m^2)$, nhưng thời gian chạy chỉ là O(m), nghĩa là tuyến tính theo số lượng các bên trong hệ thống

Tương lai, chúng tôi tiếp tục nghiên cứu, cải tiến để thuật toán có thể thực hiện hiệu quả hơn về mặt tốc độ cũng như mức độ đảm bảo riêng tư. Cụ thể:

- Làm việc trên các số nguyên thay cho số thực.
- Tìm hiểu các hệ mã hóa hiệu quả hơn về mức độ an toàn cũng như tốc độ thực hiện.
- Nghiên cứu những giao thức hiệu quả về tốc độ xử lý cũng như an toàn trong thực hiện phép hợp để tỉa tập ứng viên.
- Kết hợp với các phương pháp khác như ẩn luật kết hợp để hạn chế tiết lộ những luật nhạy cảm từ CSDL, nâng cao mức độ đảm bảo riêng tư cho thuật toán.

TÀI LIỆU THAM KHẢO

[1]. VÕ ĐÌNH BẢY, LÊ HOÀI BẮC, *Chuỗi Bit Động: Cách Tiếp Cận Mới để Khai Thác Tập Phổ Biến*, ICTFIT' 2010, Nhà xuất bản Khoa học Kỹ thuật.

[2]. R. AGRAWAL, T. IMIELINSKI AND A. SWAMI, *Mining association rules between sets of items in large databases*, In proceedings of ACM SIGMOD Intl. Conference on Management of Data (SIGMOD), 1993.

[3]. R. AGRAWAL AND R. SRIKANT, *Fast algorithms for mining association rules*, In proceedings of 20th Intl. Conf. on Very Large Data Bases (VLDB), 1994.

[4]. D. W.-L. CHEUNG, J. HAN, V. NG, A. W.C. FU,AND Y. FU, *A fast distributed algorithm for mining association rules*. In Proceedings of the 1996 International Conference on Parallel and Distributed Information Systems (PDIS'96), Miami Beach, Florida, USA.

[5]. MURAT KANTARCIOGLU AND CHRIS CLIFTON, *Privacy preserving distributed mining of association rules on horizontally partitioned data*. IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 9, pp. 1026-1037, 2004.

[6]. CLIFTON C., KANTARCIOGLOU M., LIN X., ZHU M., *Tools for privacy-preserving distributed data mining*. ACM SIGKDD Explorations, 4(2), 2002.

[7]. FLORIAN KERSCHBAUM, DEBMALYA BISWAS, AND SEBASTIAAN DE HOOGH. *Performance Comparison of Secure Comparison Protocols*. In

Proceedings of the 1st International Workshop on Business Processes Security, 2009.

[8]. URABE, S., WANG, J., KODAMA, E., AND TAKATA, T., *A high collusion-resistant approach to distributed privacy-preserving data mining*. In H. Burkhart (Ed.), Parallel and distributed computing and networks (pp. 326-331), 2007, Austria: ACTA Press.

[9]. P. PALLIER, *Public-Key Cryptosystems based on Composite Degree Residue Classes*, In Proceedings of EuroCrypt 99, Springer Verlag LNCS series, 1998, pp. 223-238.

[10]. ALEXANDRE EVFIMIEVSKI, RAMAKRISHNAN SRIKANT, RAKESH AGRAWAL, AND JOHANNES GEHRKE. *Privacy Preserving Mining of Association Rules.* Information Systems, 29(4), pages 343-364, 2004, Elsevier.

[11]. ESTIVILL-CASTRO, V., HAJYASIEN, A., *Fast Private Association Rule Mining by a Protocol Securely Sharing Distributed Data*. In Proceedings of the 2007 IEEE Intelligence and Security Informatics, New Brunswick, New Jersey, USA, May 23-24, pp. 324–330.

[12]. M. HUSSEIN, A. EL-SISI, N. ISMAIL. *Fast Cryptographic Privacy Preserving Association Rules Mining on Distributed Homogenous Data Base,* In Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II, 2008.

[13]. BIN YANG, HIROSHI NAKAGAWA, ISSEI SATO AND JUN SAKUMA, *Collusion-resistant privacy-preserving data mining,* In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, 2010**.**

[14]. MURAT KANTARCIOGLU, *"A Survey of Privacy-Preserving Methods Across Horizontally Partitioned Data,"* In Privacy-Preserving Data Mining: Models and Algorithms, pp. 313-336, 2008.

[15]. MOEZ WADDEY, PASCAL PONCELET, SADOK BEN YAHIA, *A novel approach for privacy mining of generic basic association rules*, In Proceedings of CIKM-PAVLAD, pp. 45-52, 2009.

[16]. STEFAN BöTTCHER AND SEBASTIAN OBERMEIER. *Secure set union and bag union computation for guaranteeing anonymity of distrustful participants,* Journal of Software, Vol 3(1), pp. 9-17, Academy, 2008.

Ngày nhận bài: 06/03/2011

SƠ LƯỢC VỀ TÁC GIẢ

TRẦN QUỐC VIỆT

Thạc sỹ, Phó giám đốc Trung tâm Tin học Ứng dụng, Trường Đại học Nông lâm Tp. HCM.

Hướng nghiên cứu: Đảm bảo tính riêng tư trong Khai thác dữ liệu.

Email: tqv@hcmuaf.edu.vn

CAO TÙNG ANH

Thạc sỹ, Phó Trưởng Khoa CNTT, Trường Đại học Kỹ thuật Công nghệ Tp. HCM.

Hướng nghiên cứu: Cơ sở Dữ liệu. Đảm bảo tính riêng tư trong Khai thác dữ liệu.

Email: tunganh@hcmhutech.edu.vn

LÊ HOÀI BẮC

PGS.TS. Phó Trưởng Khoa, Trưởng Bộ môn Khoa học Máy tính, Khoa CNTT, Trường Đại học Khoa học Tự nhiên Tp. HCM.

Hướng nghiên cứu: Trí tuệ Nhân tạo, Tính toán mềm và Khám phá tri thức và Khai thác dữ liệu.

Email: lhbac@fit.hcmus.edu.vn