(12) **BẢN MÔ TẢ GIẢI PHÁP HỮU ÍCH THUỘC BẰNG ĐỘC QUYỀN**
**GIẢI PHÁP HỮU ÍCH**

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **2-0003167**

(51) **H04L 9/26**
2020.01

(13) **Y**

(21) 2-2022-00143 (22) 07/01/2014
(67) 1-2021-00444
(45) 26/06/2023 423 (43)
(73) Đại học Quốc gia Thành phố Hồ Chí Minh (VN)
Phường Linh Trung, quận Thủ Đức, thành phố Hồ Chí Minh
(72) Nguyễn Đình Thúc (VN); Đặng Hải Vân (VN); Trương Thị Mỹ Trang (VN).

(54) PHƯƠNG PHÁP TÌM KIẾM TRÊN DỮ LIỆU MÃ HÓA THEO TỪ KHÓA DỰA TRÊN MA TRẬN GIẢ-NGHỊCH ĐẢO

(57) Sáng chế đề cập đến Phương pháp tìm kiếm trên dữ liệu mã hóa theo từ khóa dựa trên ma trận giả-nghịch đảo (quasi-inverse) nhằm cải thiện tốc độ tính toán so với các phương pháp đại số khác. Phương pháp bao gồm: xây dựng hàm keyGen(λ) để nhận vào tham số biểu diễn khóa công khai Kpub và khóa riêng Kpriv; xây dựng hàm PEKS(m,Kpub) để nhận vào từ khóa m và khóa công khai Kpub; xây dựng hàm trapdoor(m,Kpriv) để nhận vào từ khóa m và khóa riêng Kpriv; và xây dựng hàm test(P,T) để nhận bản mã P và trapdoor T=(T1,T2) nếu H2(P.T2)=T1 thì trả về 1 (hay true), ngược lại trả về 0 (hay false).

**Lĩnh vực kỹ thuật được đề cập**

Giải pháp hữu ích thuộc lĩnh vực công nghệ thông tin, đề cập đến phương pháp tìm kiếm trên dữ liệu mã hóa theo từ khóa dựa trên ma trận giả-nghịch đảo (quasi-inverse), và một cài đặt hiệu quả dựa trên ma trận giả - nghịch đảo để bảo vệ tính riêng tư, bí mật của dữ liệu trong cơ sở dữ liệu được gửi trên máy chủ nhà cung cấp dịch vụ. Bằng phương pháp này, ngay cả nhà cung cấp dịch vụ cũng không thể khai thác được dữ liệu. Giải pháp đặc biệt phù hợp cho các ứng dụng trong điện toán đám mây.

**Tình trạng kỹ thuật của sáng chế**

Hiện nay, các nghiên cứu trên thế giới dựa chính vào giải pháp dùng giả thiết về độ khó bài toán lô-ga-rit rời rạc trên các cấu trúc đại số nhóm. Mô hình tìm kiếm dữ liệu có mã hóa bằng từ khóa hiện nay chủ yếu được xây dựng dựa trên giả thiết về độ khó bài toán lô-ga-rit rời rạc, hàm song tuyến tính. Ưu điểm của các phương pháp đó là tính an toàn được xác định dựa trên bài toán lô-ga-rit rời rạc. Khuyết điểm chính là thời gian tìm kiếm, đặc biệt khi mở rộng tìm kiếm liên từ khóa (từ khóa "and", từ khóa "or",…). Ngoài ra, các phương pháp tìm kiếm hiện nay trên thế giới dựa trên tính toán số lớn. Vì thế, để an toàn (kích thước khóa/dữ liệu dài (1024 bit)), tốc độ tính toán trên các số lớn của các phương pháp đó chậm hơn nhiều so với phương pháp đại số (phương pháp đại số ở đây bao gồm nhưng không giới hạn ở các phép toán ma trận hay còn gọi là các phép tuyến tính.

**Bản chất kỹ thuật của sáng chế**

Do đó, mục đích của sáng chế là cải thiện tốc độ tính toán và có thể mở rộng cho phép tìm kiếm liên từ khóa theo phép "and" và "or" nhằm khắc phục các nhược điểm của các giải pháp kỹ thuật đã nêu ở trên.

Để đạt được mục đích, giải pháp hữu ích đề xuất phương pháp tìm kiếm trên dữ liệu mã hóa theo từ khóa dựa trên ma trận giả-nghịch đảo (quasi-inverse) sử dụng các tính toán tuyến tính trên ma trận nhằm cải thiện tốc độ tính toán.

**Mô tả vắn tắt hình vẽ**

Hình 1 là hình vẽ thể hiện quá trình tìm kiếm theo từ khóa theo sáng chế.

**Mô tả chi tiết sáng chế**

Dựa trên mô hình hệ thống khóa công khai cho phép tìm kiếm theo từ khóa của nhóm tác giả Boneh và cộng sự, sử dụng khái niệm giả - nghịch đảo (quasi-inverse), chúng tôi phát triển một mô hình để xây dựng cụ thể mô hình khóa công khai cho k tìm kiếm theo từ khóa (PEKS) theo Hình 1.Trong đó:

h là một phần tử chọn ngẫu nhiên trong tập M, và f là giả nghịch đảo (quasi-inverse) của h, nghĩa là f thỏa: f.h.f = f, và h.f.h = h với M là tập các phần tử giả nghịch đảo với phép toán ".";

H1, H2, H3 là các hàm được định nghĩa cụ thể khi thực thi. H1, H3 là hàm biến đổi từ khóa thành một phần tử trong M, H2 là hàm biến đổi một phần tử trong M thành một chuỗi.

Theo Hình 1, phương pháp tìm kiếm trên dữ liệu mã hóa được thực hiện như sau:

Bước 1: Alice □ người gửi sinh cặp khóa gồm khóa công khai Kpub và khóa riêng Kpriv. Khóa công khai Kpub được công bố cho Bob□người nhận biết, thông qua xây dựng hàm KeyGen(λ) để nhận vào tham số biểu diễn khóa công khai Kpub và khóa riêng Kpriv bao gồm:

chọn ngẫu nhiên phần tử h trong tập M các phần tử giả-nghịch đảo;

tính giả-nghịch đảo (quasi-inverse) f của h, nghĩa là f thỏa: f.h.f = f, và h.f.h = h với M là tập các phần tử giả nghịch đảo với phép toán "."; và

tạo khoá công khai Kpub = f.h, khóa riêng Kpriv = f.

Bước 2: Bob –người nhận dùng khóa công khai của người gửi để tạo bản mã P cho từ khóa m và gửi bản mã đến lưu trữ ở máy chủ, thông qua xây dựng hàm PEKS(m,Kpub) nhằm nhận vào từ khóa m và khóa công khai Kpub, bao gồm các bước sau:

biến đổi từ khóa m thành một phần tử z trong M bằng một hàm băm H3: z = H3(m);

tính giá trị bản mã P = z.Kpub; (H3 là hàm được định nghĩa cụ thể khi thực thi).

Bước 3: khi Alice cần tìm kiếm dữ liệu theo từ khóa m, Alice sử dụng khóa riêng Kpriv để tạo thông tin cho phép tìm ra tài liệu có từ khóa m thông qua hàm Trapdoor(m,Kpriv), kí hiệu là T, và gửi trapdoor đến cho máy chủ. Trapdoor là là một dạng mã cho phép Alice chứng minh mình là chủ của khóa công khai Kpub mà Bob đã dung để mã hóa tài liệu cũng như từ khóa m, nhưng không tiết lộ bất cứ thông tin gì về Kpriv. Bước này bao gồm:

biến đổi từ khóa m thành một phần tử q trong tập M bằng hàm băm H3: q = H3(m);

dùng hàm băm H1, H2 để tính T1 = H2(H1(m).Kpriv.q), T2 = Kpriv.q; và

trả về trapdoor T = (T1, T2) (H1, H2, H3 là các hàm được định nghĩa cụ thể khi thực thi).

Bước 4 : khi máy chủ nhận trapdoor T từ Alice gửi đến, máy chủ sẽ sử dụng trapdoor T=(T1, T2) để chạy hàm Test(P,T) nhằm kiểm tra từu khóa trong trapdoor T=(T1, T2) và từ khóa trong bản mã P lưu ở máy chủ có khớp nhau không. Như vậy mặc dù máy chủ không biết rõ từ khóa trong trapdoor T và từ khóa trong bản mã P, thông qua hàm Test, máy chủ vẫn trả lời khớp ( true) hay không khớp (false), cụ thể nếu H2(P.T2)=T1 thì trả về 1 (hay true), ngược lại trả về 0 (hay false).

Cách thực hiện hiệu quả của phương pháp trình bày trên là xây dựng M có cấu trúc đặc biệt là tập các ma-trận. Cụ thể, cho trước tham số an toàn $\lambda$= (n,p), n là số nguyên dương và p là số nguyên tố, hệ thống xây dựng một ma trận A kích thước n×1 và giả-nghịch đảo (pseudo-inverse matrix) của A trong trường Zp (với gcd(n,p)=1). Hay với $\lambda$=(m,n,p), xây dựng ma trận A kích thước m×n (m<n) và giả-nghịch đảo (pseudo-inverse matrix) của A trong trường $Z_p$ (với gcd(n,p) = 1) với các bước cụ thể như trình bày dưới. Giải pháp đề xuất 2 cách cho phép xây dựng các ma-trận giả-nghịch đảo khác nhau: (a) cho phép xây dựng ma-trận kích thước (n×1) và (b) cho phép xây dựng ma-trận tổng quát. Cụ thể:

(a) Xây dựng ma trận A kích thước n×1 và giả-nghịch đảo (pseudo-inverse matrix) của A trong trường $Z_p$ (với gcd(n,p) = 1) như sau:

Bước 1. Sinh ngẫu nhiên giá trị cho (n-1) phần tử đầu tiên của A: A[i,1]=random(p) với random(p) là hàm trả về giá trị ngẫu nhiên thuộc {1,2,…,p-1}, i=1,…, n-1

Bước 2. Chọn q sao cho ($\Sigma$ A[i,1]$^2$ + q$^2$) mod p $\neq$0 với i=1,…,n-1

Bước 3. Gán A[n,1]=q

Bước 4. Ma trận giả-nghịch đảo của A, kí hiệu $A^+$: $A^+ = (A^T.A)^{-1}.A^T$, trong đó $A^T$ là chuyển vị của A, $A^{-1}$ là phép nghịch đảo ma trận cả ma-trận A, $A^+$ là ma trận giả-nghịch đảo của A.

Bước 5. Trả về $(A,A^+)$

(b) Xây dựng ma trận A kích thước m×n (m<n) và giả-nghịch đảo (pseudo-inverse matrix) của A trong trường $Z_p$ (với gcd(n,p)=1) như sau:

Bước 1. Sinh ngẫu nhiên ma trận khả nghịch A' kích thước m×n trong trường $Z_p$ :

i. sinh ngẫu nhiên ma trận tam giác trên U sao cho tích các phần tử trên đường chéo mod p khác 0.

ii. sinh ngẫu nhiên ma trận tam giác dưới L sao cho tích các phần tử trên đường chéo mod p khác 0.

iii. A'=L.U

Bước 2. Sinh ngẫu nhiên ma trận A'' kích thước m×(n-m)

Bước 3. Ghép hai ma trận A' và A'' theo cột để tạo thành ma trận A: A=[A' || A'']

Bước 4. Nếu det($A.A^T$) mod p = 0 thì quay về bước (2); ngược lại, qua bước (5).

Bước 5. Tính $A^+=A^T.(A.A^T)^{-1}$.

Bước 6. Trả về $(A,A^+)$

**Ví dụ thực hiện sáng chế**

Ví dụ 1: xây dựng ma trận trong trường $Z_p$:

n=5 là số nguyên tốc.

Bước 1. Sinh ngẫu nhiên giả sử được A[1,1]=2, A[2,1]=6, A[3,1]=1, A[4,1]=2.

Bước 2. Sinh ngẫu nhiên q, giả sử q= 0,

Khi đó $\Sigma$ A[i,1]$^2$ + q$^2$= 0: không thỏa, qay lại bước 2.

Sinh ngẫu nhiên lại q, giả sử q= 3

Khi đó $\Sigma A[i,1]^2 + q^2 = 2$: thỏa, qua bước kế.

Bước 3. Gán A[5,1]=2

Bước 4. $A^+=(A^T.A)^{-1}.A^T$ = [2 4 5 2 5]

Bước 5. Trả về $(A,A^+)$

Ví dụ 2: xây dựng ma trận A kích thước m×n (m<n) và giả-nghịch đảo (pseudo-inverse matrix) của A trong trường Zp:

Giả sử, m=5,n=8, p=7.

Bước 1. Sinh ma trận khả nghịch A' kích thước 5× 5.

i. Sinh ma trận tam giác dưới ngẫu nhiên, giả sử ta được:

L=
[6 0 0 0 0]
[0 2 0 0 0]
[4 0 6 0 0]
[4 1 0 6 0]
[2 6 0 0 1]

ii. Sinh ma trận tam giác trên ngẫu nhiên, giả sử ta được:

U=
[6 5 2 4 3]
[0 3 5 3 2]
[0 0 3 0 4]
[0 0 0 4 4]
[0 0 0 0 6]

iii. Sinh ma trận khả nghịch 5×5:

A'=LU=
[1 2 5 3 4]
[0 6 3 6 4]
[3 6 5 2 1]
[3 2 6 1 3]
[5 0 6 5 3]

Bước 2. Sinh ngẫu nhiên ma trận 5 × 3, giả sử là:

$$A'' = \begin{bmatrix} 4 & 1 & 6 \\ 6 & 0 & 0 \\ 4 & 5 & 0 \\ 1 & 6 & 0 \\ 1 & 4 & 4 \end{bmatrix}$$

Bước 3. Tạo ma trận A”

$$A=[A'||A''] = \begin{bmatrix} 1 & 2 & 5 & 3 & 4 & 4 & 1 & 6 \\ 0 & 6 & 3 & 6 & 4 & 6 & 0 & 0 \\ 3 & 6 & 5 & 2 & 1 & 4 & 5 & 0 \\ 3 & 2 & 6 & 1 & 3 & 1 & 6 & 0 \\ 5 & 0 & 6 & 5 & 3 & 1 & 4 & 4 \end{bmatrix}$$

Bước 4. $\det(AA^T)= 1$

Bước 5. Sinh ma trận giả nghịch đảo

$$A^+=A^T.(A.A^T)^{-1} = \begin{bmatrix} 6 & 5 & 0 & 5 & 6 \\ 6 & 4 & 0 & 1 & 3 \\ 4 & 4 & 3 & 1 & 5 \\ 3 & 5 & 2 & 5 & 2 \\ 0 & 1 & 5 & 1 & 2 \\ 3 & 6 & 6 & 1 & 4 \\ 4 & 5 & 6 & 3 & 3 \\ 6 & 4 & 1 & 3 & 0 \end{bmatrix}$$

Ví dụ 3: minh họa hệ mã khóa công khai cho phép tìm kiếm theo từ khóa sử dụng ma trận giả nghịch đảo.

- KeyGen():

1. Sinh ma trận G và ma trận giả-nghịch đảo của G

$$G = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$G^+ = \begin{bmatrix} 3 & 6 \end{bmatrix}$$

2. Tạo khóa công khai Kpub và khóa riêng Kpriv

Kpub=G.F=

[3 6]

[6 5]

Kpriv=G=

[1]

[2]

- PEKS(m,Kpub):

1. Giả sử z=H1(m)=[2 5]

2. Tính giá trị bản mã P=z.Kpub=[1 2]

- Trapdoor(m,Kpriv):

1. Giả sử: q=H3(m)=[3]

2. Giả sử H1(m)=[2 5]

T1=H2(H1(m).Kpriv.q)=H2(1)

T2=Kpriv.q=

[3]

[6]

3. Trả về trapdoor T=(T1, T2)

- Test(P,T):

H2(P.T2)=H2(1)=T1 nên trả về 1 (hay true).

## YÊU CẦU BẢO HỘ

1. Phương pháp tìm kiếm trên dữ liệu mã hóa theo từ khóa dựa trên ma trận giả-nghịch đảo (quasi-inverse):

xây dựng hàm keyGen(λ) để nhận vào tham số biểu diễn khóa công khai Kpub và khóa riêng Kpriv bao gồm:

chọn ngẫu nhiên phần tử h trong tập M các phần tử giả-nghịch đảo;

tính giả-nghịch đảo (quasi-inverse) f của h, nghĩa là f thỏa: f.h.f = f, và h.f.h = h với M là tập các phần tử giả nghịch đảo với phép toán "."; và

tạo khoá công khai Kpub = f.h, khóa riêng Kpriv = f;

xây dựng hàm PEKS(m,Kpub) để nhận vào từ khóa m và khóa công khai Kpub bao gồm:

biến đổi từ khóa m thành một phần tử z trong M bằng một hàm băm H3: z=H3(m);

tính giá trị bản mã P=z.Kpub với H3 là hàm được định nghĩa cụ thể khi thực thi;

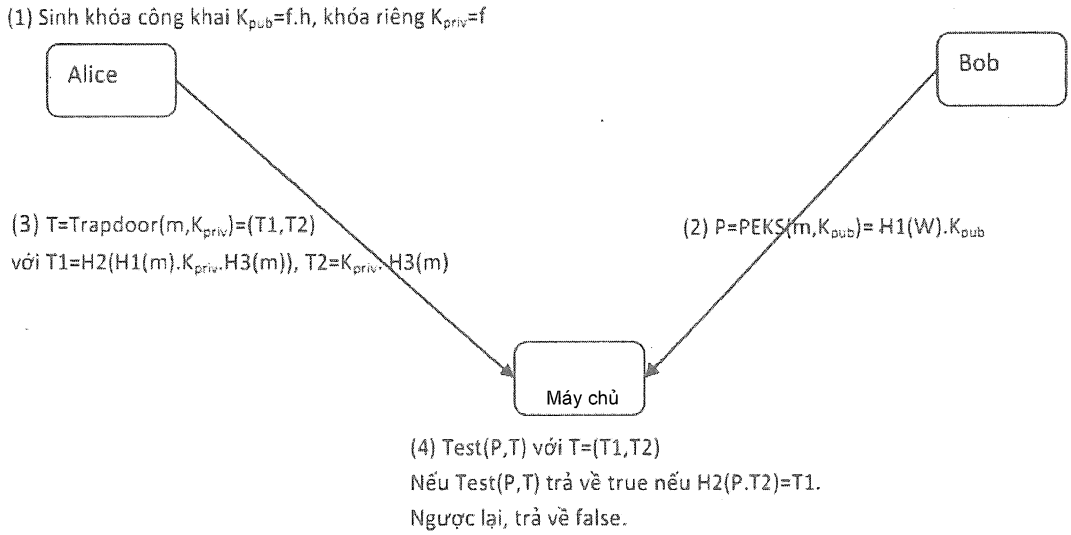
xây dựng hàm trapdoor(m,Kpriv) để nhận vào từ khóa m và khóa riêng Kpriv bao gồm:

biến đổi từ khóa m thành một phần tử q trong M bằng hàm băm H3: q=H3(m);

dùng hàm băm H1, H2 để tính T1=H2(H1(m).Kpriv.q), và T2=Kpriv.q; và

trả về trapdoor T=(T1, T2) với H1, H2, H3 là các hàm được định nghĩa cụ thể khi thực thi;

xây dựng hàm test(P,T) để nhận bản mã P và trapdoor T=(T1,T2) nếu H2(P.T2)=T1 thì trả về 1 (hay true), ngược lại trả về 0 (hay false).


(1) Sinh khóa công khai $K_{pub}$=f.h, khóa riêng $K_{priv}$=f
Alice
Bob
(3) T=Trapdoor(m,$K_{priv}$)=(T1,T2)
với T1=H2(H1(m).$K_{priv}$.H3(m)), T2=$K_{priv}$.H3(m)
(2) P=PEKS(m,$K_{pub}$)= H1(W).$K_{pub}$
Máy chủ
(4) Test(P,T) với T=(T1,T2)
Nếu Test(P,T) trả về true nếu H2(P.T2)=T1.
Ngược lại, trả về false.

Hình 1