ĐẢM BẢO TIN CẬY CHO KẾT QUẢ TRUY VẤN TRÊN CƠ SỞ DỮ LIỆU LƯU TẠI NHÀ CUNG CẤP DỊCH VỤ: HIỆN TRẠNG NGHIÊN CỨU

Phạm Thị Bạch Huệ

Đặng Trần Khánh

TÓM TẮT

Sự phát triển nhanh chóng về kỹ thuật mạng và nhu cầu quản lý dữ liệu làm cho việc gửi cơ sở dữ liệu (CSDL) đến nhà cung cấp dịch vụ CSDL hiện nay trở thành một xu hướng mới. Vì người dùng có thể không tin vào máy chủ của nhà cung cấp dịch vụ nên có nhiều vấn đề về bảo mật được đặt ra. Có thể kể đến những vấn đề về tính bí mật của dữ liệu, tính riêng tư người dùng, tính riêng tư dữ liệu và đảm bảo tin cậy cho kết quả truy vấn. Trong đó, vấn đề đảm bảo tin cậy cho kết quả truy vấn có vai trò quan trọng.

Bài viết trình bày các giải pháp đã được đề nghị cho vấn đề đảm bảo tin cậy cho kết quả truy vấn trên CSDL, gồm CSDL quan hệ và CSDL XML. Ngoài việc trình bày các giải pháp hiện có, bài viết còn đánh giá từng giải pháp và đề ra hướng nghiên cứu trong tương lai.

Từ khoá: *Đảm bảo tin cậy cho kết quả truy vấn, CSDL XML, nhà cung cấp dịch vụ CSDL, máy chủ không tin cậy.*

ABSTRACT

With rapid advances in networking technologies and big demand for data management, outsourcing database services has recently been a new trend. In this outsourcing model, a service provider is typically not fully trusted, and thus they raise numerous problems related to security issues. These problems are referred to as data confidentiality, user privacy, data privacy, and query assurance. Among them, query assurance takes an important role to the success of the database outsourcing model.

In this paper, we present existing solutions to query assurance for outsourced databases, both traditional databases and XML databases. Besides, we discuss and evaluate all of them and propose future research directions related.

Keywords: *Query assurance, outsourced XML databases, untrusted server, database service provider.*

I. GIỚI THIỆU

Thông thường, CSDL được quản lý bởi chính tổ chức sở hữu chúng (in-house database). Việc này gây tốn nhiều chi phí khi hiện thực và duy trì hệ thống: chi phí phần cứng (máy móc, thiết bị mạng), chi phí phần mềm (bản quyền hệ quản trị CSDL, các công cụ hỗ trợ quản lý dữ liệu), chi phí thuê nhân viên quản lý CSDL và quản lý hệ thống mạng, …

Sự phát triển nhanh chóng trong công nghệ mạng và nhu cầu quản lý CSDL làm hình thành một dịch vụ mới, *database-as-a-service, gọi là dịch vụ gửi CSDL ở nhà cung cấp dịch vụ*. Với dịch vụ này, nhà cung cấp dịch vụ (Service Provider - SP) cung cấp nơi lưu trữ, dịch vụ quản lý dữ liệu do người sở hữu dữ liệu (Data Owner - DO) gửi đến, và cung cấp cho máy khách (Client) cơ chế thao tác trên dữ liệu. Người sử dụng dịch vụ này sẽ chỉ tốn một khoản chi phí nhỏ hơn nhiều lần so với việc tự xây dựng và quản lý hệ thống CSDL, trong khi vẫn được sử dụng một hệ thống được bảo trì và nâng cấp một cách chuyên nghiệp, lại không cần bận tâm đến việc quản lý dữ liệu và có thời gian tập trung vào các hoạt động chính yếu hơn

Người quản trị CSDL (database administrator) là người của SP, chịu trách nhiệm quản lý dữ liệu. Phía SP có đội ngũ nhân viên kỹ thuật chịu trách nhiệm cập nhật phần cứng, phần mềm, điều chỉnh hiệu năng làm việc của máy chủ, quản lý bảo mật và phân quyền. DO tạo, xóa, hiệu chỉnh nội dung CSDL, tạo chỉ mục,… và đăng tải chúng lên máy chủ của SP. Nói chung, DO có thể cập nhật CSDL thường xuyên. Client gửi nhu cầu truy vấn thông tin đến máy chủ của SP (truy vấn về dữ liệu của DO) và nhận về kết quả truy vấn. Về khía cạnh bảo mật cũng như về nội dung kết quả truy vấn được trả về, người truy cập có thể không tin vào phía máy chủ. Hệ thống cung cấp dịch vụ đăng tải CSDL (của SP) cùng sự hỗ trợ của DO cần đảm bảo dịch vụ đang cung cấp là bảo mật và kết quả trả về cho máy khách là đáng tin cậy.

Nói đến dịch vụ này, nếu CSDL gửi đến SP là CSDL truyền thống, ta gọi tắt là mô hình ODBS (Outsourced Database Service), nếu là CSDL XML ta gọi mô hình này là OXMLDBS (Outsourced XML Database Service). Có bốn mô hình ODBS/ OXMLDBS [9]:

- *SS model* (**S**ingle user-**S**ervice provider): DO, cũng chính là máy khách duy nhất, gọi chung là người dùng (user), gửi CSDL đến máy chủ và truy cập CSDL. Đây là mô hình khá phổ biến. Một tổ chức thuê SP lưu trữ dữ liệu nội bộ của mình và truy cập trên CSDL này.
- *MS model* (**M**ultiple data owner-**S**ervice provider): Giống với mô hình SS, chỉ khác là mô hình này có nhiều người DO cùng sở hữu một CSDL gửi tại máy chủ của SP và những người sở hữu dữ liệu này cũng là client duy nhất. Một ví dụ cho mô hình này là CSDL bảo hiểm, 1 nhân viên bảo hiểm là 1 DO, tạo và bảo trì một/ một số dòng dữ liệu trên CSDL lưu tại SP. Mỗi DO sở hữu các dòng dữ liệu do chính họ tạo ra (và chịu trách nhiệm quản lý các khách hàng liên quan).
- *SMS model* (**S**ingle data owner-**M**ultiple clients-**S**ervice provider): Đây là mô hình đặc trưng nhất trong các loại mô hình gửi CSDL đến SP và có các vấn đề về bảo mật phức tạp nhất. DO thuê nhà cung cấp dịch vụ lưu trữ CSDL của mình và bán thông tin cho các khách hàng có nhu cầu. Theo mô hình này, chỉ có duy nhất một DO gửi dữ liệu đến SP. Trừ DO ra, có nhiều client được truy cập đến CSDL theo giao ước.
- *MMS model* (**M**ultiple data owner-**M**ultiple clients-**S**ervice provider): Giống như mô hình SMS, chỉ khác là có n người cùng sở hữu một CSDL đang được đăng tải tại nhà cung cấp, $n \geq 2$.

Trong mô hình ODBS, việc quản lý dữ liệu và xử lý câu truy vấn thuộc về trách nhiệm của SP. Nếu không có giải pháp, cả dữ liệu và các câu truy vấn do người dùng gửi đến đều có thể bị phơi bày cho máy chủ hoặc kẻ tấn công hoặc một số người dùng có mục đích không tốt. Cho nên, bên cạnh việc bảo mật mạng liên lạc và các thủ tục phía client, giải pháp hiệu quả cho vấn đề bảo mật phía máy chủ là vô cùng quan trọng. Các yêu cầu quan trọng về bảo mật trong hệ thống ODBS được liệt kê như sau:

- *Tính bí mật của dữ liệu (Data Confidentiality)*: DO không muốn người ngoài hệ thống (outsider) hay ngay cả người quản lý máy chủ (server administrator) nhìn thấy nội dung của dữ liệu của mình, kể cả khi câu truy vấn được thực thi trên máy chủ.
- *Tính riêng tư (Privacy):*
  - *Tính riêng tư người dùng (User privacy):* Máy khách không muốn máy chủ và người quản trị CSDL biết họ đã truy vấn điều gì và kết quả trả về là gì. Định danh của người dùng đôi khi cần phải không cho máy chủ biết.
  - *Tính riêng tư của dữ liệu (Data privacy):* Máy khách không thể có nhiều thông tin hơn những thông tin mà họ được phép truy cập từ máy chủ.
- *Kết quả truy vấn là đáng tin cậy (Query Assurance - QA):* Máy khách phải được đảm bảo kết quả truy vấn được là đúng (correctness), là đủ (completeness) và mới nhất (freshness).

- *Tính đúng*: Kết quả trả về là đúng khi nó có nguồn gốc từ dữ liệu nguyên thủy ban đầu và không bị chỉnh sửa.
- *Tính đủ*: Kết quả trả về là đủ khi đó là tất cả các dòng dữ liệu lẽ ra phải được trả về (các dòng dữ liệu thỏa mãn điều kiện truy vấn), không bỏ bớt dòng nào mà cũng không trả ra nhiều hơn lượng dữ liệu cần được trả về.
- *Tính mới*: Kết quả trả về phải dựa trên tình trạng mới nhất của CSDL trên đó người dùng đã thực hiện truy vấn.

Bài viết quan tâm đến việc đảm bảo tin cậy cho kết quả truy vấn và trình bày các giải pháp lien quan đến vấn đề này.

II. ĐẢM BẢO TIN CẬY CHO KẾT QUẢ TRUY VẤN TRÊN MÔ HÌNH ODBS

1. Hướng tiếp cận dùng chữ ký điện tử

Tính đúng

Mykletun, Narasimha và G.Tsudik [5] đưa ra một hướng tiếp cận để đảm bảo tính đúng cho kết quả truy vấn dựa trên mô hình chữ ký điện tử. DO trước khi gửi dữ liệu đến SP sẽ thực hiện ký trên từng dòng dữ liệu. Client (cũng chính là DO theo mô hình SS) khi truy vấn và nhận kết quả trả về từ máy chủ sẽ thực hiện chứng thực để kiểm tra dữ liệu trả về có phải do chính DO đã tạo ra hay không. Mô hình chữ ký điện tử được chọn phổ biến hiện nay là RSA với chiều dài của chữ ký là 1024 bit.

Tuy nhiên, đánh giá cách tiếp cận này, tác giả cho rằng nếu kết quả trả về có nhiều dòng dữ liệu thì chi phí đường truyền client-server và chi phí cho việc chứng thực tại client là cao.

Trong trường hợp số lượng dòng dữ liệu trả về lớn thì dẫn đến việc lãng phí về mặt bandwidth cũng như thời gian tính toán tại client để chứng thực dữ liệu. Mekletun et al [5] đề nghị mô hình Condensed-RSA, gọi là mô hình chữ ký điện tử kết hợp. Giả sử có tập t message $\{m_1,\ldots,m_t\}$ với tập chữ ký tương ứng $\{ \sigma_1,\ldots, \sigma_t)$, được ký bởi cùng 1 người ký, chữ ký Condensed-RSA được tính bởi:

$\sigma_{1,t} = \prod_i \sigma_i \pmod n$, $i = 1..t$

Khi đó việc kiểm chứng chữ ký $\sigma_{1,t}$ tương đương với việc kiểm chứng t chữ ký σ_i riêng lẻ. Một lợi điểm khác là kích thước của Condensed-RSA bằng với kích thước của một RSA chuẩn. Như vậy, thay vì trả về toàn bộ các chữ ký của từng dòng riêng lẻ, server chỉ cần tính toán chữ ký Condensed-RSA và trả về cho client để có thể thực hiện việc chứng thực dữ liệu.

Trong phạm vi bài báo này, tác giả chưa cung cấp giải pháp cho việc đảm bảo tính đủ. Sau đó, tác giả đã phát triển giải pháp này nhằm đảm bảo thêm tính đủ bằng cách thay đổi cách ký trên từng record, trong khi tính đúng vẫn được đảm bảo [1].

Tính đủ

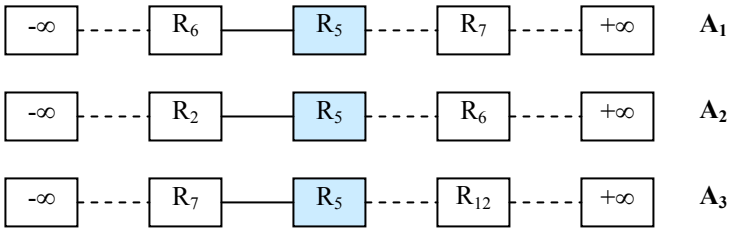
Narasimha và Tsudik [1] đề nghị hướng tiếp cận gọi là Digital Signature Aggregation and Chaining (DSAC) để đảm bảo tính đúng và tính đủ cho kết quả truy vấn. [1] là một mở rộng [5] nhằm bổ sung giải pháp cho tính đầy đủ. (Tính đúng tác giả vẫn dùng mô hình chữ ký điện tử kết hợp như đã trình bày). Do đó, phần tiếp theo của tài liệu chỉ trình bày biện pháp để đạt được tính đầy đủ.

Tính đầy đủ đạt được bằng cách xây dựng một mối liên kết bảo mật giữa các chữ ký của từng record, gọi là signature-chain. Chuỗi liên kết này đạt được bằng cách thay đổi cách tính chữ ký của từng record như sau:

$Sign(r) = h(h(r)||h(IPR_1(r))|| \ldots h(IPR_l(r)))SK$

Trong đó, h() là hàm băm mã hóa (như SHA), || là phép nối, IPR_i là record liền kề trước đó chuỗi record sắp xếp theo chiều (thuộc tính) i, l là số chiều (thuộc tính) có thể thực hiện truy vấn, SK là khóa riêng của DO.

Các record liền kề trước của mỗi record được xác định bằng cách sắp xếp quan hệ R theo các chiều có thể truy vấn, như hình sau:



Hình 1. *Sắp xếp quan hệ R theo các chiều truy vấn phục vụ cho DSAC*

Các record liền kề trước của R_5 lần lược là R_6, R_2, R_7. Khi đó, chữ ký của R_5 được tính như sau: $Sign(R_5) = h(h(R_5)||h(R_6)||h(R_2)||h(R_7))SK$.

Phương pháp chứng minh tính đủ

Hầu hết các nghiên cứu đều tập trung vào câu truy vấn điểm (point query - có kết quả trả về là các dòng dữ liệu có giá trị tại thuộc tính tương ứng bằng với giá trị điều kiện) và câu truy vấn miền (range query – có kết quả trả về là các dòng dữ liệu sao cho giá trị tại các thuộc tính tương ứng nằm trong cận trên và cận dưới của điều kiện miền).

Giả sử rằng câu truy vấn miền Q trên quan hệ $\Re$ trả về tập các dòng dữ liệu S, ta có:

$S = \{r| r \in, r.x \geq LB, r.x \leq UB\}$

LB, UB là hai giá trị biên và r là một quan hệ của quan hệ $\Re$.

Để chứng minh tính đủ, máy chủ trả về thêm 2 dòng dữ liệu nữa:

$S_b = \{r_L| r_i \in \Re, r_L.x = \max(r_i.x), r_i.x < LB, \forall i\} \cup \{r_U| r_j \in \Re, r_U.x = \min(r_j.x), r_j.x > UB, \forall j\}$ (*)

Nếu quan hệ $\Re$ được sắp xếp theo thuộc tính x và r_L, r_U thỏa biểu thức (*), không có dòng dữ liệu nào rơi vào r_L và min(S), r_U và max(S) thì kết quả có được có tính đủ.

Với chuỗi chữ ký như trên, server trả lời cho truy vấn miền (range query) bằng cách:

- Trả về tất cả các dòng thoả điều kiện.
- Hai dòng ở 2 biên của kết quả trả về (không thuộc tập kết quả).
- Chữ ký kết hợp của tập kết quả trả về.

Chữ ký kết hợp để chứng minh tính đúng. Hai dòng biên để lấy ra kết quả lẽ ra phải được trả về. Tính chữ ký kết hợp của tập lẽ ra phải được trả về, sau đó, so sánh với chữ ký kết hợp mà server trả về để kiểm tra xem kết quả trả về có đủ hay không. Trong cả [5] và [1], tác giả không đề cập đến tính mới.

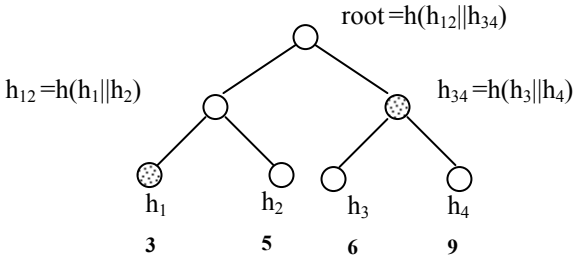
Tính mới

Trong các nghiên cứu liên quan chứng minh 3 tính chất của kết quả truy vấn, chỉ có [2], [8] có quan tâm đến tính mới, nhưng [2], [8] làm việc trên dữ liệu dạng cây, nên sẽ được trình bày ở III.

2. Hướng tiếp cận dùng cấu trúc dữ liệu xác thực

Cách tiếp cận dùng cấu trúc dữ liệu xác thực (AuthDS – Authenticated Data Structures) mở rộng ý tưởng của MHT [1] để đảm bảo tính đúng và đầy đủ cho kết quả truy vấn. Theo cách này, dữ

liệu được lưu tại các node lá và đã được sắp xếp. Các node lá này cũng chứa giá trị băm của dữ liệu, node trung gian chứa giá trị băm của kết hợp các giá trị băm ở các node con. Cách tính này đảm bảo thứ tự của dữ liệu trong 1 danh sách có sắp xếp. Node gốc được ký dùng mô hình chữ ký điện tử khóa công khai, thường là RSA.



Hình 2. *Cây MHT*

Để đảm bảo tính đúng, máy chủ chỉ cần trả về giá trị cần tìm và VO (Verification Object, là tập hợp các node trung gian cho phép tính lại các giá trị băm và xác thực node gốc) để xác thực giá trị node đó là thực sự hiện diện trên cây MHT. Vì dữ liệu trên cây là có thứ tự ở các node lá nên ta hoàn toàn có thể áp dụng phương pháp chung để đảm bảo tính đủ. Ngoài ra, khi chứng minh tính đủ, máy chủ phải trả ra thêm hai node biên nữa.

Pang et al [11] ứng dụng kết hợp ý tưởng cây MHT với cấu trúc dữ liệu B-tree để đảm bảo tính đúng cho kết quả truy vấn. Tuy nhiên, thay vì chỉ ký ở node gốc của cây, tác giả đề nghị ký trên tất cả các node với mục đích làm giảm chi phí trong quá trình xác nhận. Tuy nhiên, theo [1], tác giả cho rằng phương pháp này là không hiệu quả, không bảo mật và không thể mở rộng để đảm bảo tính đúng cho kết quả trả về. Hạn chế này không phải do việc vận dụng cây MHT mà là do giải pháp mà tác giả đề nghị.

Ngoài ra, có nhiều cách kết hợp phương pháp dùng AuthDS này với các cấu trúc dữ liệu khác nhau nhằm làm tăng hiệu quả trong quá trình đảm bảo tin cậy cho kết quả trả về: ASB (Aggregated Signatures with B^+-tree), MB (Merkle B-tree), EMB (Embedded Merkle B-tree) [3].

3. Hướng tiếp cận Challenge – Response

Radu Sion đề nghị cơ chế đảm bảo tính đúng cho câu truy vấn bất kỳ, dựa trên nghi thức Challenge – Response [10]. DO chia dữ liệu thành k đoạn nhỏ, tính giá trị băm cho từng đoạn trước khi lưu chúng ở máy chủ của nhà cung cấp dịch vụ. Ngoài ra, cách tiếp cận này còn dùng một số câu truy vấn đã biết kết quả cho từng đoạn nhằm phục vụ cho việc xác nhận tính đủ của kết quả trả về sau này.

Giả sử S là dữ liệu cần gửi đến SP. S được phân thành nhiều đoạn Si, mỗi Si sẽ được xác định bởi một hàm băm dùng để đảm bảo dữ liệu là chính xác, không bị thay đổi. Giá trị này gọi là "identity-hash", được sử dụng để chứng thực các câu truy vấn "identity query", là những câu truy vấn trả về toàn bộ dữ liệu trong Si.

Quá trình thực thi các câu truy vấn như sau:

- Trong tập các query Q {Q1, Q2, Q3, .. Qa} cần thực thi, querier sẽ chèn vào câu query Qx tại một vị trí bất kỳ, querier đã biết trước kết quả trả về của Qx. Đồng thời, querier tính toán một challenge token bằng {H(ε||ρ(Qx)), ε}. Trong đó, H() là hàm mã hóa một chiều bất khả đảo (non-invertible one-way hashing function); ε là một giá trị duy nhất theo thời gian (timestamp) để đảm bảo challenge token là duy nhất; ρ(Qx) : là kết quả trả về đã được biết trước bởi querier.

- Nhiệm vụ của server là thực thi các câu query và xác định được giá trị x bằng cách áp dụng hàm H() cho các kết quả. Và gởi kèm x về cùng với kết quả của các truy vấn.

Tuy nhiên, phương pháp trên chỉ tập trung cho giải quyết các câu truy vấn đọc dữ liệu (select query) chứ chưa giải quyết vấn đề cho các câu truy vấn cập nhật/thêm mới dữ liệu (update/insert query). Việc cập nhật dữ liệu thực hiện bằng cách đọc toàn bộ đoạn dữ liệu có chứa dòng cần update (hay sẽ chứa hàng insert). Sau đó thực hiện cập nhật dữ liệu, tính toán lại "identity hash" rồi cập nhật trở lại server. Tuy nhiên, việc xử lý tình huống cập nhật dữ liệu vẫn chỉ là bước khởi đầu.

4. Nhận xét

- Các cách tiếp cận dựa trên chữ ký điện tử và cấu trúc dữ liệu xác thực hiện tại vẫn chỉ giải quyết cho các câu truy vấn chỉ đọc đơn giản không có các hàm tính gộp (SUM, AVG, …), còn phụ thuộc vào dạng thức của câu truy vấn. Cần phải phân tích câu truy vấn thành từng phần riêng lẻ mới có thể có những tác vụ thích hợp.
- Một giới hạn lớn của AuthDS là đòi hỏi phải bảo trì một cấu trúc dữ liệu phức tạp bên cạnh dữ liệu thực sự. Cấu trúc này cần phải được tính toán đầy đủ trước khi đưa lên server. Mỗi thay đổi cập nhật dữ liệu đòi hỏi phải tốn chi phí không nhỏ để cập nhật lại các số liệu trong cấu trúc. Bên cạnh đó, để có thể đảm bảo tính đúng cũng như đầy đủ của cây truy vấn theo khoảng (range query) đòi hỏi phải xây dựng một cấu trúc cho từng thuộc tính, theo trật tự sắp xếp. Kết quả là chi phí tính toán tại DO tăng cao.
- Li [3] nhận xét rằng việc đánh giá chi phí cho cách tiếp cận dùng chữ ký điện tử thường bỏ qua các phép tính băm, ký, xác nhận và nhân modulo. Các phép toán này có chi phí rất lớn. Chi phí cho phép băm ít hơn phép ký. Tác giả đề nghị giải pháp dùng càng ít phép ký càng tốt.
- Theo Mycletun et al [5], mặc dù chi phí tính toán tại client trong mô hình AuthDS có thể thấp hơn so với mô hình chữ ký điện tử, nhưng chi phí đường truyền client-server cao hơn nhiều so với mô hình chữ ký kết hợp.
- Đối với cách tiếp cận AuthDS, khi kết quả trả về là nhiều mẫu tin thì chi phí đường truyền sẽ giảm bớt so với kết quả trả về chỉ có 1 mẫu tin, vì khi đó có nhiều node cha chung.
- Hướng tiếp cận của Radu [10] có thể áp dụng cho tất cả các loại truy vấn, kể cả việc sử dụng các hàm gộp mà [1], [5], [3] chưa giải quyết được. Ưu điểm chính của phương pháp này là không cần phân tích cú pháp của các câu truy vấn nên có thể triển khai dễ dàng hơn. Tuy nhiên, hướng tiếp cận này vẫn còn một số điều cần xem xét như sau:
  - Chỉ áp dụng cho tập các câu truy vấn, chưa giải quyết cho trường hợp thực thi từng câu truy vấn riêng lẻ, vốn được sử dụng khá nhiều trong thực tế.
  - Để giải quyết vấn đề này có thể sử dụng các hướng như sau: (1) sử dụng các fakequery kèm theo để biến câu truy vấn đơn thành tập các câu truy vấn. Tuy nhiên, cách này có thể làm quá tải server, giảm hiệu năng của toàn hệ thống do phải thực hiện các fake-query quá nhiều so với các truy vấn thực sự. (2) các câu query riêng lẻ được tập trung lại tại một trust-server và gửi đến server dưới dạng tập các câu truy vấn theo đúng tinh thần của giải pháp. Phương thức này hầu như không khả thi do thời gian trễ của câu query trong thời gian chờ đợi là không thể chấp nhận được. (3) là kết hợp của (1) và (2).
  - Chưa chứng minh triệt để kết quả trả về là đầy đủ. Xác suất để server không thực thi hoặc thực thi không hoàn chỉnh đối với câu query cuối cùng là 33%. Đây là một xác suất khá cao. Điều này phần nào làm giảm bớt tính tin cậy của giải pháp.

Các hướng tiếp cận trên đều được thực hiện cho các dữ liệu dạng quan hệ (relational database). Do đó, để có thể áp dụng được trong CSDL XML cần phải có một số thay đổi nhất định.

III. ĐẢM BẢO TIN CẬY CHO KẾT QUẢ TRUY VẤN TRÊN DỮ LIỆU DẠNG CÂY

1. Trường hợp dữ liệu cây chỉ mục

- **Kiểm tra tính đúng**: Trước khi gửi dữ liệu đến nhà cung cấp dịch vụ, người sở hữu dữ liệu tính giá trị băm h(m) cho từng node, sau đó thực hiện ký và lưu chữ ký cùng với bảng EncryptedTable tại máy chủ. Bảng EncryptedTable có 3 cột NID, EncryptedNode, Signature. Bằng cách này, người dùng có thể xác nhận tính đúng cho từng node trả về dùng mô hình chữ ký điện tử khóa công khai. Để giảm chi phí liên lạc và chi phí tính toán, tác giả dùng mô hình chữ ký kết hợp cho các node trong tập truy cập thừa [6].
- **Kiểm tra tính đủ:** Khi người dùng yêu cầu máy chủ trả về tập hợp thừa A gồm t node A $=\{m_1, ..., m_t\}$, máy chủ sẽ trả về tập hợp R gồm t node R = $\{n_1, ..., n_t\}$ [6]. Người dùng phải có thể chứng minh được mỗi phần tử trong tập R đều tìm thấy trong tập A. Cụ thể hơn, người dùng yêu cầu các node (ở dụng mã hóa) thông qua NID của chúng. Người dùng phải có thể chứng minh được rằng tìm thấy giá trị NID tương ứng trong tập R đối với từng giá trị NID của tập A. Giải pháp cho vấn đề này là như sau:
  - Trường EncryptedNode thay vì chỉ chứa nội dung node, giờ bao gồm thêm NID của chính node đó, được mã hóa và lưu tại máy chủ.
  - Sau đó, DO ký trên giá trị mã hóa đó dùng mô hình chữ ký điện tử RSA, và lưu lại chữ ký.
  - Chữ ký dùng để kiểm tra tính đúng. Tính đủ được kiểm tra bằng cách trích giá trị NID trong nội dung của từng node (dạng mã hóa), kiểm tra xem từng NID của tập A có tìm thấy trong tập R và ngược lại hay không.
- **Kiểm tra tính mới:** Để kiểm tra tính mới, người dùng phải có thể xác minh được rằng máy chủ đã trả về kết quả dựa trên tình trạng mới nhất của CSDL. Tác giả đề ra giải pháp dùng nhãn thời gian. Nhãn thời gian của từng node con được lưu ở node cha, nghĩa là 1 node sẽ lưu nhãn thời gian của những node con. Nhãn thời gian của một node chỉ thay đổi khi DO cập nhật nội dung của node đó. SNODE lưu nhãn thời gian của root (cùng với những thông tin khác như metada, địa chỉ của node gốc,…). Nhãn thời gian của node root chỉ thay đổi khi DO thay đổi nội dung của root.

Field EncryptedNode, ngoài giá trị thật sự của field và NID, DO bổ sung thêm thông tin nhãn thời gian của node, sau đó mới thực hiện mã hóa, ký và lưu vào bảng B+EncryptedTable.

Khi nhận được NID do máy chủ trả về, người dùng kiểm tra tính mới bằng cách:

1. Truy cập node root dùng SNODE, và lần theo cây cho đến khi đến được node cha của node cần kiểm tra, người dùng biết nhãn thời gian thực sự của node này (là nhãn thời gian phản ánh tình trạng mới nhất của dữ liệu).
2. Kiểm tra nhãn thời gian của node này bằng cách giải mã field EncryptedNode có bằng giá trị nhãn thời gian thực sự hay không.

Những người dùng hợp lệ luôn biết thong tin để truy cập SNODE. Ngoài ra, DO thông báo về nhãn thời gian của SNODE khi vừa thực hiện cập nhật dữ liệu.

NID	EncryptedNode	Signature
0	D0a1n2g3Kh75nhs&#2.l$	S0
1	T9&8ra§ÖÄajh³q91c%.h3	S1
2	H&$uye”µnÜis57ß@j9.M	s2
3	L?{inh*ß²³&§gnaD1x<-≠	S3
4	Wh09a/[%?Ö*#Aj2k;}o≤	S4
5	j8Hß}[aHo$§angµGl0:”Ω	S5
6	#Xyi29?ß~R@€>Kh{}-©	S6
7	√B³!jKDÖbd0K3}%§5,¥	S7
8	T-§µran&gU19=75mz*£	S8

Chữ ký điện tử dùng để đảm bảo tính đúng.

Có chứa giá trị NID để đảm bảo tính đủ và giá trị nhãn thời gian (của node con) để đảm bảo tính mới.

Hình 3. Các *thông tin bổ sung để đảm bảo tin cậy cho kết quả truy vấn*

2. Trường hợp dữ liệu XML

XML là một dạng dữ liệu bán cấu trúc (semistructured data), dạng cây (treestructured). Tương tự như RDB truyền thống, mỗi tài liệu XML đều được đặc trưng bởi một lược đồ (schema) định nghĩa mối quan hệ cha con giữa các node, số lượng thuộc tính của của mỗi node. Lược đồ có 2 loại node là element node (gọi là t-node) và attribute node (gọi là a-node).

Sau đây là phần trình bày hai phương pháp thường dùng để lưu trữ tài liệu XML: dạng bảng (table-based) và dạng node (node-based).

Dạng bảng (table-based)

Từ cây cấu trúc của tài liệu, ta chuyển sang dạng lược đồ quan hệ theo các bước sau.

- Gán nhãn (labeling) các t-node cấu trúc sao cho mỗi node có một giá trị nhãn duy nhất.
- Mỗi t-node cấu trúc được chuyển thành một bảng tương ứng có tên là tên của t-node kết hợp với giá trị nhãn. Các a-node con của t-node này được chuyển thành các cột của bảng. Mỗi bảng bổ sung thêm cột nodeid là định danh của node trong bảng dữ liệu. Nếu t-node có cha, thì bổ sung thêm cột pnodeid tham chiếu đến bảng phát sinh từ t-node cha.

Dạng node: (node-based)

Các này lưu từng t-node và a-node của cây dữ liệu.

Tương tự như phương pháp trên, đầu tiên, các node cấu trúc (bao gồm cả t-node và a-node) đều phải được gán nhãn. Phương pháp gán nhãn tương tự như trên (chỉ khác là việc gán nhãn bao gồm cả a-node). Khi đó, việc lưu xuống CSDL quan hệ sẽ tồn tại 1 bảng dữ liệu để lưu t-node và a-node, nhưng cấu trúc của chúng là như sau:

t-node(nodeid, xtype, datatype, nameid, pnodeid, lmaid, value)

a-node(nodeid, xtype, datatype, nameid, pnodeid, sibid, value)

Trong đó, nodeid là định danh của node, xtype dùng để phân biệt các loại đối tượng, datatype dùng để xác loại dữ liệu, nameid là định danh của tên của node (t-node và a-node), pnodeid là định danh của t-node cha của t-node hiện tại, đối với a-node, pnodeid là định danh của t-node cha của t-node cha của a-node hiện tại, lmaid là định danh của a-node trái nhất, sibid là định danh của a-node anh em bên phải.

Phương pháp đảm bảo tin cậy cho kết quả truy vấn

Với dữ liệu XML lưu dạng bảng: Dữ liệu XML sau khi được chuyển sang dạng bảng, ta có thể áp dụng các phương pháp đảm bảo truy vấn trên mô hình ODBS như: DSAC [1] hay EMB Tree [3].

Tuy nhiên, lược đồ của CSDL XML có thể thay đổi bất cứ lúc nào, dẫn đến việc thay đổi cấu trúc bảng tương ứng, làm thay đổi nhiều đến dữ liệu đã được lưu trữ. Dù dùng phương pháp nào trong các phương pháp đảm bảo tin cậy cho kết quả truy vấn kể trên, việc thêm cột vào một bảng có thể dẫn đến việc tính toán lại toàn bộ các chữ ký điện tử, mã hóa lại toàn bộ dữ liệu, …. Điều này hoàn toàn không có lợi, nhất là khi dữ liệu đã gửi đến SP. Về phía người dùng, cần chuyển đổi ngôn ngữ truy vấn trên liệu XML (XPath, XQuery,…) sang ngôn ngữ SQL.

Dù vậy, trường hợp CSDL XML không thay đổi về lược đồ thì vẫn có thể áp dụng phương pháp lưu trữ dữ liệu XML theo kiểu này này để có thể tận dụng được các kết quả đã được nghiên cứu tốt trên CSDL quan hệ.

Với dữ liệu XML lưu dạng node: Cách lưu trữ này thể hiện đúng bản chất dạng cây của tài liệu XML, nên khắc phục được khuyết điểm của phương pháp table-based. Việc thay đổi cấu trúc của tài liệu XML không ảnh hưởng nhiều đến nội dung lưu trữ hiện tại mà chỉ ảnh hưởng đến node cần cập nhật. Với dạng thức lưu trữ như vậy, việc thay đổi lược đồ (bổ sung/bỏ bớt một thuộc tính) chỉ đơn thuần như một tác vụ insert/delete đơn giản, và chỉ ảnh hưởng đến node hiện tại.

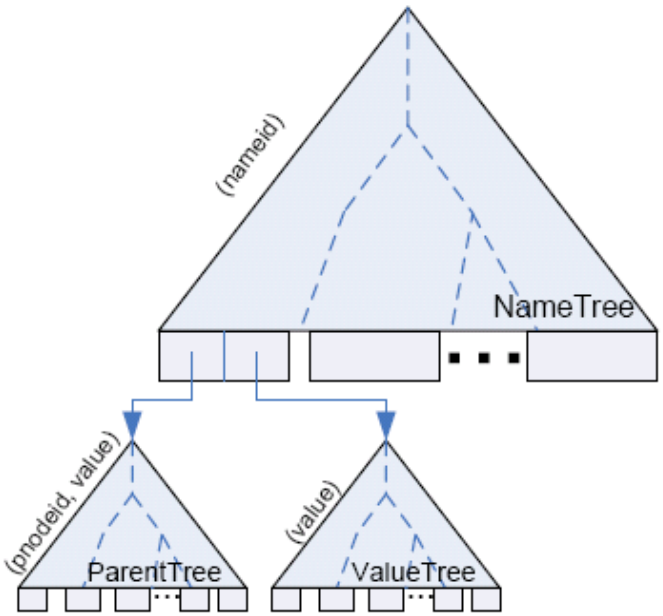
Theo [8], một yếu tố quan trọng của một giải pháp đảm bảo độ tin cậy khi truy vấn trên OXMLDB là cấu trúc chỉ mục. Chỉ mục giữ vai trò quản lý lưu trữ và tìm kiếm trên dữ liệu. Bằng cách nhúng thêm thông tin vào cấu trúc chỉ mục, có thể đảm bảo tin cậy cho kết quả truy vấn như sau.

Xét một tài liệu XML, các *node* được định vị bởi *path,* tức đường đi từ *node* gốc đến *node* hiện tại. Các truy vấn trên XML thông thường được xác định *path*. Như vậy, chỉ mục XML, ngoài giá trị của *node*, cần phải chứa thêm thông tin về *path* của node.

Quay lại phương pháp lưu trữ dữ liệu XML dạng node, giá trị *nameid* là duy nhất đối với mỗi node cấu trúc, do đó có thể được sử dụng tương đương với *path*. Như vậy, mỗi *node* cần được chỉ mục trên bộ hai thuộc tính (*nameid, value*). Ngoài việc truy vấn theo giá trị, với bản chất cha/con của dữ liệu dạng cây XML thì yêu cầu truy vấn các *node* con khi đã biết được *node* cha là thường xuyên. Để chứng minh tính đủ cho các truy vấn này, cần bổ sung thêm chỉ mục của bộ ba giá trị (*nameid, pnodeid, value*) để các *record* được sắp xếp theo *node* cha.

Nested B^+-Tree (NBT)

Để đảm bảo được yêu cầu trên, có thể sử dụng kết hợp các cấu trúc cây như sau. Xây dựng một cây B+-Tree với khóa so sánh là *nameid*, gọi là *NameTree*. Tại node lá của *NameTree* chứa gốc của hai cây B+Tree theo khóa lần lượt là (*pnodeid, value*) và (*value*). Hai cây này có tên là: *ParentTree* và *ValueTree*. Tập hợp ba loại cây này tạo thành một cấu trúc dữ liệu, gọi là *Nested B+Tree*, cho phép lập chỉ mục cho tài liệu XML trên hai bộ giá trị (*nameid, pnodeid, value*) và (*nameid, value*).



Hình 4. *Nested B+-Tree*

Nested Merkle B$^+$-Tree

Dựa trên ý tưởng của MHT, cộng thêm một số thông tin vào NBT 3 tính chất cho kết quả truy vấn sẽ được chứng minh.

Từng node của NBT là giá trị băm của các node con. Cách tính như sau:

a-node: $H_{a\text{-}node} = h(nodeid||xtype|| \dots || value)$

t-node: $H_{t\text{-}node} = h(h(nodeid|| \dots||value) \cup_i H_{attr})$

Node lá của ValueTree, ParentTree:

$$H_L = h(\cup_i H_{data\text{-}record})$$

Node trung gian:

$$H_i = h(\cup_i H_{child\text{-}node})$$

Lá của NameTree:

$$H_{L\text{-}N} = h(H_{vtree} || H_{ptree})$$

Node gốc của NameTree:

$$H_R = h(\varepsilon || \cup_i H_{child\text{-}node})$$

H_{attr}: là giá trị băm của một a-node của một t-node cho trước.

$H_{data\text{-}record}$ là $H_{a\text{-}node}$ hoặc $H_{t\text{-}node}$ có kết nối.

$H_{child\text{-}node}$ là H_L, H_I, hoặc $H_{L\text{-}N}$.

H_{ptree} và H_{vtree} là giá trị băm của node gốc tương ứng của ParentTree hoặc ValueTree.

ε là giá trị nhãn thời gian và h() là hàm băm một chiều (SHA1, MD5, …). Ngoài ra, ta ký trên gốc của NameTree dùng khóa bí mật của mô hình chữ ký điện tử (RSA, DSA). Khóa công khai tương ứng và nhãn thời gian ε được công bố rộng rãi cho các máy khách.

Cách đảm bảo QA cho kết quả truy vấn

Tính đúng và tính đủ: Giả sử rằng kết quả bao gồm các mục ở mức lá của ValueTree chứa các con trỏ trỏ đến các mẫu tin nằm trong khoảng cho trước (truy vấn đoạn). Như đã đề cập, sẽ có thêm 2 mẫu tin nữa đi kèm theo kết quả trả về. Kết quả là $\{L_i, L_{i+1}, \ldots, L_j\}$, ngoài ra máy chủ trả về thêm các mẫu tin và giá trị băm của những mục không nằm trong kết quả để máy khách có thể tính lại giá trị băm của những node lá giữa L_i và L_j, gọi là co-path. Kết quả thực tế và co-path được đóng gói trongmột cấu trúc gọi là VO và gửi đến máy khách. Bằng cách tính toán đệ quy, máy chủ trả về node gốc, chữ ký của node gốc và nhãn thời gian. Máy khách tính toán lại giá trị băm của node gốc và xác nhận chúng dùng chữ ký, qua đó kiểm tra được kết quả có đúng và đủ hay không.

Tính mới: Cùng với kết quả trả về, máy chủ trả về giá trị nhãn thời gian của node gốc. Sau khi xác nhận chữ ký của node gốc, máy khách so sánh nhãn thời gian này với nhãn thời gian được người sở hữu dữ liệu công bố rộng rãi trước kia. Nếu bằng thì máy khách hoàn toàn có thể tin cậy về tính mới của kết quả.

Ngoài ra, cũng có một số kết quả khác quan tâm đến vấn đề bảo đảm tính đúng và đủ cho tài liệu XML. Tuy nhiên, các kết quả sắp được trình bày liên quan đến việc phát hành tài liệu XML.

IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Việc đảm bảo tin cậy cho kết quả truy vấn trên dữ liệu XML chỉ mới khởi đầu, chủ yếu chỉ cho mô hình ODBS đơn giản nhất là SS. Đa số các cách tiếp cận cho vấn đề này trên mô hình ODBS và OXMLDBS chỉ cho truy vấn chỉ đọc, loại truy vấn điểm và truy vấn miền. Các loại truy vấn dạng cập nhật và tính gộp (aggregation) chưa được giải quyết. Đảm bảo tính đúng cho kết quả trả về không phải là một vấn đề khó. Nhưng đối với việc đảm bảo tính đủ và tính mới, các giải pháp hiện tại gặp phải hạn chế là yêu cầu phía client tính toán quá nhiều. Điều này làm giảm giá trị mà mô hình ODBS/OXMLDBS mang lại.

Từ những nhận xét trên, việc nghiên cứu nhằm đề ra giải pháp cho việc đảm bảo tin cậy cho các loại câu truy vấn (gồm cả thao tác cập nhật dữ liệu và các câu truy vấn phức tạp hơn) là cần thiết. Giải pháp này cần giảm thiểu chi phí tính toán phía client, và có thể áp dụng cho mô hình phức tạp nhất trong các mô hình ODBS.

TÀI LIỆU THAM KHẢO

[1] Narasimha M., & Tsudik G. *Authentication of outsourced databases using signature aggregation and chaining*. Proceedings of the 11th International Conference on Database Systems for Advanced Applications (pp. 420-436), Singapore, 2006.

[2] Tran Khanh Dang. *Ensuring correctness, completeness, and freshness for Outsourced Tree-Indexed Data*. Information Resource Management Journal, 2008.

[3] Feifei Li, Marios Hadjieleftheriou, George Kollios, Leonid Reyzin. *Dynamic Authenticated Index Structures for Outsourced Databases* Sigmod 2006, Chicago, Illinois, USA.

[4] E. Bertino, B. Carminati, E. Ferrari, B. Thuraisingham, A. Gupta. *Selective and Authentic Third-Party Distribution of XML Documents*, IEEE Transactions on Knowledge and Data Engineering, 16(10), 1263-1278 2-2002.

[5] Einar Mykletun, Maithili Narasimha, Gene Tsudik. *Authentication and Integrity in Outsourced Databases*. In ISOC Symposium on Network and Distributed System Security NDSS, 2004.

[6] Lin, P., & Candan, K.S. (2004). *Hiding traversal of tree structured data from untrusted data stores*. Proceedings of the 2nd International Workshop on Security in Information Systems (pp. 314-323), Porto, Portugal.

[7] Tran Khanh Dang. *Security Issues in Outsourced XML Databases*, A book chapter in the book titled "Open and Novel Issues in XML Database Applications: Future Directions and Advanced Technologies", IGI Global, April 2008.

[8] Viet Hung Nguyen, Tran Khanh Dang. *A novel solution to query assurance verification for dynamic outsourced XML databases*, Journal of Software, Vol 3, No. 4, April 2008.

[9] Tran Khanh DANG. *Security Protocols for Outsourcing Database Services.* Information and Security: An International Journal, ProCon Ltd.,Sofia, Bulgaria, ISSN 1311-1493, Vol. 18, 2006, p.85-108.

[10] Radu Sion. *Query Execution Assurance for Outsourced Databases*. Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005.

[11] Pang, H.H., & Tan, K-L. (2004). *Authenticating query results in edge computing*. Proceedings of the 20th International Conference on Data Engineering (pp. 560-571), Boston, USA.

[12] Viet Hung Nguyen. *Security Issues in Querying Dynamic Outsourced XML databases*, Master thesis, 2007.