XÁC THỰC CƠ SỞ DỮ LIỆU MÃ HOÁ THUÊ NGOÀI DỰA TRÊN XÁC THỰC LÔ

Hồ Kim Giàu[1*], Nguyễn Hiếu Minh[2]
[1]Học viện Kỹ thuật quân sự
[2]Học viện Kỹ thuật mật mã

TÓM TẮT
Khi thuê ngoài dữ liệu, Chủ sở hữu dữ liệu giao toàn quyền quản trị cơ sở dữ liệu của mình cho nhà cung cấp dịch vụ. Tuy nhiên, nhà cung cấp dịch vụ có thể không đảm bảo an toàn tuyệt đối cho dữ liệu của chủ sở hữu. Ngoài ra, các kẻ tấn công trên môi trường mạng có thể xâm nhập và thay đổi dữ liệu. Để bảo vệ cơ sở dữ liệu của mình, chủ sở hữu thường mã hoá dữ liệu trước khi lưu trữ lên máy chủ của nhà cung cấp dịch vụ. Từ đó, yêu cầu đảm bảo tính toàn vẹn và tính xác thực của dữ liệu khi truy vấn dữ liệu cần phải quan tâm giải quyết.

Trong bài báo này, chúng tôi trình bày các vấn đề đảm bảo tính toàn vẹn dữ liệu và đề xuất mô hình xác thực khi truy vấn cơ sở dữ liệu mã hoá thuê ngoài. Chúng tôi đề xuất lược đồ xác thực lô mới, áp dụng nó để xác thực dữ liệu và đánh giá độ phức tạp tính toán của thuật toán.
Từ khóa: *Thuê ngoài CSDL; CSDL mã hoá; xác thực; xác thực lô.*

Ngày nhận bài: 11/6/2019; Ngày hoàn thiện: 06/8/2019; Ngày đăng: 12/8/2019

AUTHENTICATION OF ENCRYPTED OUTSOURCED DATABASE BASED ON BATCH VERIFICATION

Ho Kim Giau[1*], Nguyen Hieu Minh[2]
[1]Military Technology Academy
[2]Academy of Cryptography Techniques

ABSTRACT
When outsourcing data, the Data owner gives full authority of his database to the service provider. However, the service provider may not guarantee the safety of the owner's data. In addition, attackers on the network environment can steal and change data. To protect own database, Data owner often encrypts data before storing it on the service provider's server. From there, it is required to ensure the integrity and authenticity of the data when querying the encrypted database.

In this paper, we present data integrity issues and propose authentication models when querying an outsourced encrypted database. We propose a new batch verification scheme, apply it to verify data and evaluate the computation complexity of the algorithm.
Keywords: *Outsourced database; encrypted database; authentication; batch verification.*

Received: 11/6/2019; Revised: 06/8/2019; Published: 12/8/2019

* Corresponding author: *Tel: 0919.447.897*; *Email: hkgiau@gmail.com*

1. Giới thiệu

Chủ sở hữu dữ liệu (Data Owner - DO) sử dụng hình thức thuê ngoài cơ sở dữ liệu (Outsourced Database - ODB) để giảm các chi phí và tăng khả năng xử lý. Trong mô hình ODB, DO cần một nhà cung cấp dịch vụ cơ sở dữ liệu (Database Service Provider – DSP). DO chia sẻ dữ liệu của mình cho người dùng thông qua máy chủ DSP. Tuy nhiên, máy chủ DSP có thể không đảm bảo sự an toàn cho dữ liệu của DO. Có nhiều nguyên nhân khiến DO không thể tin tưởng máy chủ DSP như:

1. Phần mềm máy chủ DSP bị lỗi.
2. Thao tác của nhân viên bị sai hoặc cố tình can thiệp bất hợp pháp vào dữ liệu.
3. Hệ thống của DSP bị tấn công.

Dữ liệu là tài sản rất quan trọng của DO. Mọi lộ lọt thông tin có thể gây thiệt hại lớn. Do đó, DO phải bảo vệ tính bí mật dữ liệu của mình thường bằng cách mã hoá dữ liệu trước khi lưu trữ lên DSP.

Bên cạnh bảo vệ tính bí mật, DO còn phải bảo vệ tính toàn vẹn của dữ liệu. Nghĩa là khi dữ liệu bị can thiệp bất hợp pháp (sửa, xoá...) thì DO phải phát hiện và xử lý để đảm bảo rằng khi DSP trả dữ liệu về cho người dùng thì DO phải xác thực dữ liệu đó là nguyên bản như ban đầu khi DO đưa dữ liệu lên DSP. Có nhiều nghiên cứu xác thực cơ sở dữ liệu thuê ngoài, trong đó có các phương pháp như:

• Sử dụng cấu trúc dữ liệu xác thực (Authenticated Data Structure – ADS) [1], [2], [3]. Tuy nhiên, ADS không hiệu quả đối với dữ liệu động do thời gian thay đổi cấu trúc của ADS rất lớn. Cụ thể, nếu một bản ghi được chèn hoặc xóa, tất cả các bản ghi kế tiếp bản ghi bị sửa đổi cũng thay đổi theo.

• Mykletun [4] giới thiệu chữ ký số gộp (Condensed-RSA) để giảm thời gian tính toán, tuy nhiên không đề xuất phương pháp xử lý truy vấn. Sau đó, Yuan [5] đã đề xuất một lược đồ truy vấn gộp có thể xác thực cho cơ sở dữ liệu thuê ngoài. Mỗi bản ghi được gán thẻ (tag) xác thực dùng để kiểm tra tính toàn vẹn của kết quả truy vấn gộp. Tuy nhiên, phương pháp này chỉ kiểm chứng trên dữ liệu rõ và phải kiểm tra thẻ của các bản ghi liên quan đến truy vấn. Nghĩa là hệ thống phải tính toán lại chữ ký của bản ghi mặc dù truy vấn chỉ lấy một vài thuộc tính trong bản ghi.

Các nghiên cứu xác thực CSDL thuê ngoài thường xây dựng các cấu trúc hỗ trợ để xác thực, do đó nếu CSDL động thì tốn nhiều chi phí để tính toán lại. Ngoài ra, một số nghiên cứu chỉ hỗ trợ vài dạng truy vấn đặc biệt mà chưa hỗ trợ truy vấn trên nhiều bảng. Trong bài báo này, chúng tôi tập trung vào việc xác thực dữ liệu trả về mà không đề xuất phương pháp truy vấn trên dữ liệu mã hoá. Khi truy vấn, DSP trả về dữ liệu mã. Phương pháp đề xuất sẽ kiểm tra kết quả trả về, nếu dữ liệu trả về là đúng thì tiến hành giải mã và trả kết quả rõ cho người dùng. Ngược lại, sẽ thông báo đến cho DO để có phương án xử lý. Phương pháp đề xuất hiệu quả khi CSDL động do không phải xây dựng lại các cấu trúc xác thực.

Trong bài báo này, chúng tôi quy ước một số ký hiệu như bảng 1.

Bảng 1: *Các ký hiệu và phép toán*

Ký hiệu	Ý nghĩa
$\|\|$	Phép toán nối chuỗi
$H()$	Hàm băm một chiều
$E_k(m)$	Hàm mã hoá dữ liệu m với khoá k ($E()$ là các thuật toán DES, AES...)
$D_k(c)$	Hàm giải mã dữ liệu c với khoá k (Hàm ngược của hàm E)
$S(m)$	Hàm tạo chữ ký cho thông điệp m.
$V(\sigma_i)$	Hàm xác thực lô cho nhiều chữ ký σ_i.

Phần còn lại của bài báo được tổ chức như sau: Trong phần 2, chúng tôi giới thiệu xác thực lô và cải tiến của nó để chống lại các tấn công. Phần 3 đề xuất mô hình xác thực và các thuật toán xử lý để thực hiện quá trình kiểm tra, giải mã dữ liệu trả về kết quả rõ cho người dùng; phần 4 phân tích, đánh giá về độ phức tạp thời gian tính toán của các thuật toán; phần 5 kết luận nêu kết quả đạt được và hướng nghiên cứu tiếp theo.

2. Xác thực lô (Batch verification)

2.1 *Chữ ký số*

Chữ ký số được dùng để xác thực về nguồn gốc và tính toàn vẹn của thông tin. Hiện nay, các thuật toán chữ ký số thường dựa trên hai hệ mật phổ biến là RSA và Elgamal. Hệ mật RSA dựa trên độ khó của bài toán phân tích thừa số nguyên tố của số nguyên lớn. Elgamal dựa trên độ khó của bài toán logarit rời rạc. Có một số chữ ký số dựa trên hệ mật Elgamal được xem là chuẩn như: DSA, GOST và KCDSA. Chữ ký số DSA (Digital Signature Algorithm) là dạng chữ ký trong chuẩn chữ ký số (Digital Signature Standard – DSS) của chính phủ Mỹ và được

đưa ra bởi FIPS-186 [6]. GOST là chuẩn chữ ký của Liên bang Nga [7] và KCDSA là chuẩn chữ ký số của Hàn Quốc [8].

Năm 1989, Schnorr đề xuất một lược đồ chữ ký số [9] và được ứng dụng trong nhiều lĩnh vực xác thực. Cho các tham số p, q, g trên bài toán logarit rời rạc (DLP), p là một số nguyên tố lớn có kích thước trong khoảng từ 512 đến 1024 bit, q là một ước số nguyên tố của $p-1$ có kích thước 160 bit, g là phần tử sinh cấp q. Thuật toán tạo khóa, ký và xác thực được định nghĩa:

- Tạo khoá: khoá bí mật x, khoá công khai y.
  Chọn số ngẫu nhiên $x \in 1, ..., q-1$ và tính $y \leftarrow g^x \bmod p$.
- Ký: tạo chữ ký σ cho thông điệp $m \in \{0,1\}^*$
  1. Chọn số ngẫu nhiên $t \in Z_q$ và cho $r \leftarrow g^t \bmod p$.
  2. Tính $h \leftarrow H(m||r)$.
  3. Tính $s \leftarrow t - hx \bmod q$.
  4. Xuất $\sigma \leftarrow (h, s)$.
- Xác thực: Kiểm tra chữ ký σ đúng/sai.
  1. Tính $r' \leftarrow g^s y^h (\bmod p)$.
  2. Tính $h' \leftarrow H(m||r')$.
  3. Nếu $h' = h$, trả về 1 (đúng), ngược lại trả về 0 (sai).

Tuy nhiên, Hiraku Morita và cộng sự chỉ ra sự không an toàn của thuật toán Schnorr [10], và đã đề xuất thuật toán Schnorr cải tiến bằng cách sửa đổi giá trị h trong thuật toán ký. Morita thêm một giá trị ψ vào hàm băm dựa trên khoá xác thực. h được tính như sau:

$$h \leftarrow H(m||r||\psi) \text{ với } \psi \leftarrow g^x$$

Khi xác thực chữ ký, ta tính $h' \leftarrow H(m, r', y)$ và so sánh với h. Nếu $h' = h$, trả về 1, ngược lại trả về 0.

Lưu ý ở thuật toán ký, Morita cho rằng biểu thức của $\psi \leftarrow g^x$ được tính lại mỗi lần thực hiện ký mà không nên sử dụng khóa xác thực y. Điều này chống lại việc giả mạo khoá ký.

Năm 1998, Harn [11] cải thiện thuật toán DSA bằng cách chọn số nguyên tố p có độ dài trong khoảng từ 512 đến 1024 bit, q là số nguyên tố 160 bit. Thuật toán tạo khóa, ký và xác thực được định nghĩa:

- Tạo khoá: tạo khoá bí mật x, khoá công khai y.
  Chọn $x \in Z_q$ và cho $y \leftarrow g^x \bmod p$.
- Ký: tạo chữ ký σ cho thông điệp $m \in \{0,1\}^*$ dựa trên khoá bí mật x
  1. Chọn số nguyên ngẫu nhiên $t \in Z_q$.
  2. Tính $h \leftarrow H(m)$.
  3. Tính $r \leftarrow (g^t \bmod p) \bmod q$
  4. Tính $s \leftarrow rt - hx \bmod q$.
  5. Xuất $\sigma \leftarrow (r, s)$
- Xác thực: Kiểm tra chữ ký σ đúng/sai
  1. Tính $h \leftarrow H(m)$.
  2. Tính $r' \leftarrow ((g^{sr^{-1}} y^{hr^{-1}}) \bmod p) \bmod q$
  3. Nếu $r' = r$, trả về 1 (đúng), ngược lại trả về 0 (sai).

2.2 *Xác thực lô*

Xác thực lô (batch verification) là hình thức xác thực nhiều chữ ký số cùng một lúc và hiệu quả hơn việc xác thực từng chữ ký riêng lẻ. Tuỳ thuộc vào các chữ ký số được ký bởi các lược đồ khác nhau sẽ có thuật toán xác thực lô khác nhau. Có hai loại xác thực lô: xác thực lô tương tác và xác thực lô xác suất. Xác thực lô tương tác là dạng mà người ký tạo ra chữ ký σ thông qua sự tương tác với người xác thực, và sau đó người xác thực xác nhận tất cả các chữ ký σ này cùng một lúc. Xác thực lô xác suất là đối với mỗi thông điệp m được ký, một số nguyên t ngẫu nhiên được chọn riêng và sau đó được tính lại giá trị r (ví dụ trong DSA, $r = (g^t \bmod p) \bmod q$). Thay vì ký thông điệp m trực tiếp, các lược đồ chữ ký số phải ký dựa trên kết quả băm một chiều của m.

Định nghĩa 1: Cho k thông điệp m_i tương ứng với k chữ ký số σ_i được ký bởi khoá bí mật của người ký, $i = 1, ...k$. Với y là khoá công khai của người ký, hàm V được gọi là hàm xác thực lô nếu $V_y(\sigma_1, \sigma_2, ..., \sigma_k) \in \{0, 1\}$
Nếu kết quả của hàm $V_y = 1$, thì tất cả các thông điệp m_i là đúng với thông điệp của người ký. Nếu một trong những thông điệp m_i bị thay đổi thì hàm $V_y = 0$.

Năm 1994, Naccache và cộng sự [12] đề xuất xác thực lô tương tác gồm hai phần: phần tạo chữ ký và phần xác thực. Phần tạo chữ ký được người ký và người xác thực có sự tương tác với nhau. Phần xác thực được người xác thực kiểm tra nhiều chữ ký cùng một lúc. Cho n thông điệp m_i, phần tạo chữ ký được thực hiện:

- Với $i = 1, ..., n$, người ký chọn ngẫu nhiên $k_i \in Z_q$, tính $\lambda_i = g^{k_i} \bmod p$ và gửi λ_i cho người xác thực.
- Người xác thực gửi lại thông điệp ngẫu nhiên b_i có chiều dài e bit.
- Người ký tính $s_i = k_i^{-1}(H(m_i||b_i) + \lambda_i x) \bmod q$ và gửi s_i cho người xác thực.

Phần xác thực: Người xác thực kiểm tra

$\prod_{i=1}^{n}\lambda_i = g^{\sum_{i=1}^{n} s_i^{-1}H(m_i||b_i)} y^{\sum_{i=1}^{n} s_i^{-1}\lambda_i} \bmod p$

và thay thế $\{\lambda_i, s_i, b_i, m_i\}_{i=1,...,n}$ bởi $\{r_i = \lambda_i \bmod q, s_i, m_i||b_i\}_{i=1,...,n}$. Tuy nhiên, Lim và Lee [13] chỉ ra lược đồ xác thực lô tương tác của Naccache có thể bị tấn công.

Năm 1995, Yen và Laih [14] đề xuất xác thực lô dựa trên lược đồ Schnorr hoặc Brickell-McCurley, tuy nhiên phương pháp này sau đó được Boyd và Pavlovski [15] đã chỉ ra là không an toàn.

Năm 1998, Bellare, Garay và Rabin [16] đưa ra phương pháp xác thực lô theo luỹ thừa modulo. Xác thực lô cho lũy thừa modulo được Bellare và cộng sự trình bày như sau:

Cho G là một nhóm có cấp là q, g là phần tử sinh của G. Hàm luỹ thừa modulo là $x \leftarrow g^x$, với $x \in Z_q$. Định nghĩa biểu thức $EXP_{G,g}(x, y) = 1$ nếu $g^x = y, x \in Z_q, y \in G$. Nếu muốn xác thực nhiều giá trị $(x_1, y_1), ..., (x_n, y_n)$ là kiểm tra biểu thức $EXP_{G,g}(x_i, y_i) = 1$ với $i = 1, ..., n$ thì cách ngây thơ nhất là tính g^{x_i} và so sánh với y_i. Tuy nhiên, cách này sẽ tốn n lần tính luỹ thừa. Bellare đề xuất ba phương pháp tính toán nhanh xác thực lô cho biểu thức $EXP_{G,g}$ là kiểm tra tập hợp con ngẫu nhiên (random subset test), kiểm tra số mũ nhỏ (small exponents test) và kiểm tra khối (bucket test).

Năm 1998, Harn đề xuất thuật toán xác thực lô dựa trên DSA [11] Cho k thông điệp m_i với k chữ ký DSA $\sigma_i(r_i, s_i), i = 1, 2, ..., k$, đầu tiên ta tính:

$$u = \sum_{i=1}^{k} s_i r_i^{-1};\ v = \sum_{i=1}^{k} H(m_i) r_i^{-1}$$

Sau đó, ta kiểm tra phương trình:

$$\prod_{i=1}^{k} r_i \bmod q = (g^u y^v \bmod p) \bmod q \quad (1)$$

Ta kết luận rằng $(\sigma_1, \sigma_2, ..., \sigma_k)$ là một lô k chữ ký hợp lệ nếu thoả mãn phương trình (1).

Tuy nhiên, Boyd và Pavlovski đã chỉ ra rằng thuật toán xác thực lô của Harn vẫn có thể bị tấn công [15]. Zhou và cộng sự [17] đã đề xuất sự cải tiến của xác thực lô của Harn bằng cách tính hàm băm $h = H(r||m)$ trong khi ký và xác thực thay vì $H(m)$ như của Harn. Zhou cũng chỉ ra rằng khi thêm giá trị r_i vào hàm băm thì kẻ tấn công sẽ không biết được giá trị băm h_i và không thể tính toán giá trị r_i. Do đó, không thể tấn công giả mạo chữ ký. Ngoài ra, Shao đưa ra ví dụ chứng minh rằng lược đồ xác thực lô của Harn có kết quả không chính xác khi thực hiện [18].

Năm 2001, Shao đề xuất thuật toán xác thực lô dựa trên lược đồ chữ ký số Schnorr [18]. Giả sử có k thông điệp được ký bởi thuật toán Schnorr, nghĩa là có k chữ ký $\sigma_i(r_i, s_i), i = 1, 2, ..., k,\ i = 1, 2. \ldots, k$ được ký cùng một người ký với khoá bí mật x. Để xác thực k chữ ký cùng lúc, Shao chọn ngẫu nhiên k số nguyên $u_i \in (1, 2^{32}), i = 1, 2, ..., k$ và kiểm tra nếu thoả mãn phương trình:

$$\prod_{i=1}^{k} r_i^{u_i} = (g^{\sum_{i=1}^{k} u_i s_i} y^{\sum_{i=1}^{k} u_i H(m_i||r_i)}) \bmod p$$

thì k chữ ký đó là hợp lệ.

2.3 Đề xuất lược đồ xác thực lô dựa trên hai bài toán khó

Trong phần này, chúng tôi đề xuất lược đồ xác thực chữ ký số lô dựa trên độ khó của hai bài toán là phân tích thừa số và logarit rời rạc. Muốn phá vỡ lược đồ chữ ký này yêu cầu giải quyết đồng thời hai bài toán khó là logarit rời rạc trên trường $GF(p)$ và phân tích số. Các giá trị được dùng trong lược đồ: sử dụng số nguyên tố p có dạng $p = 2n+1$, với $n = qq'$, q và q' là số nguyên tố có độ dài ít nhất 1024 bit, trong đó q, q' = 3 (mod 4). g là phần tử sinh của Z_p^* có cấp bằng n, nghĩa là $g^n = 1 \pmod p$. Các định nghĩa 2, bổ đề 1, 2 về lý thuyết số được tham khảo từ tài liệu [19].

Định nghĩa 2: Cho $a \in Z_n^*$. a được gọi là một thặng dư bậc hai modulo n, hay là số chính phương modulo n, nếu tồn tại $x \in Z_n^*$ sao cho $x^2 = a \pmod n$. Nếu không tồn tại x, thì a được gọi là số không chính phương modulo n. Tập các số chính phương modulo n được ký hiệu Q_n và tập các số không chính phương modulo n ký hiệu là $\bar{Q}_n$.

Bổ đề 1: Cho số nguyên tố p với $p = 3 \pmod 4$, và số chính phương $a \in Q_p$. a có hai giá trị căn bậc 2 modulo p là $\{r, -r\}$ với $r = a^{(p+1)/4} \bmod p$.

Bổ đề 2: Cho $n = pq$ là với p, q là hai số nguyên tố khác nhau, thì $|Q_n| = (p-1)(q-1)/4$.

Thuật toán tạo khóa, ký, kiểm tra chữ ký và xác thực lô được định nghĩa như sau:

- Tạo khoá:

Thuật toán 1 Tạo khoá công khai, khoá bí mật

1: Chọn ngẫu nhiên một số nguyên x : $1 < x < n$.
2: Tính $y = g^x \bmod p$.
3: Khoá công khai là (n, y). Khoá bí mật là (x, q, q')

- Tạo chữ ký $S(m)$:

Thuật toán 2 Tạo chữ ký σ cho thông điệp $m \in \{0,1\}^*$

Input: Thông điệp m.
Output: Chữ ký σ.
Method:
1: Chọn số t ngẫu nhiên, $1 < t \leq n-1$, tính $r = g^t \bmod p$.
2: Tính $s' = t - H(m||r)x \pmod n$, nếu s' không là số chính phương modulo n thì quay lại bước 1.
3: Tính s với $s^2 = s' \bmod n$.
4: Chữ ký $\sigma \leftarrow (r, s)$

- Kiểm tra chữ ký:

Thuật toán 3 Kiểm tra chữ ký $\sigma(r, s)$ của thông điệp m

Input: Thông điệp m, chữ ký $\sigma(r, s)$.
Output: 1 (Đúng) hoặc 0 (Sai).
Method:
1: Tính $s' = s^2 \bmod n$
2: Tính $r' = g^{s'} y^{H(m||r)} \bmod p$
3: Nếu $r' = r$ thì trả về 1 (đúng), ngược lại trả về 0 (sai).

- Xác thuật lô $V(\sigma_i)$:

Thuật toán 4 Xác thực k chữ ký $\sigma_i(r_i, s_i)$ cho k thông điệp $m_i, i = 1, 2, ..., k$, được ký bởi cùng một người ký.

Input: Thông điệp m_i, k chữ ký $\sigma_i(r_i, s_i)$, $1 \leq i \leq k$.
Output: 1 (Đúng) hoặc 0 (Sai).
Method:
1: Tính $s'_i = s_i^2 \bmod n$
2: Tính:

$$u = \sum_{i=1}^{k} s'_i \; ; \; v = \sum_{i=1}^{k} H(m_i||r_i)$$

3: Kiểm tra xác thực lô theo phương trình sau:

$$\prod_{i=1}^{k} r_i = g^u y^v \bmod p \quad (2)$$

4: Nếu phương trình (2) thoả mãn, ta kết luận $(\sigma_1, \sigma_2, ..., \sigma_k)$ là k chữ ký hợp lệ, trả về 1 (đúng), ngược lại trả về 0 (sai).

Lưu ý:
Trong giai đoạn tạo chữ ký $S(m)$, ta phải tính giá trị s sao cho $s^2 = s' \bmod n$ với $n = qq'$. Theo định lý số dư Trung Hoa (Chinese Remainder Theorem - CRT) ta có hệ phương trình:

$$s^2 = s' \bmod q$$
$$s^2 = s' \bmod q'$$

Mỗi phương trình có hai giá trị căn bậc hai tương ứng lần lượt là $\{a, -a\}$, $\{b, -b\}$ với $a = s'^{(q+1)/4}$, $b = s'^{(q'+1)/4}$. Do đó, ta có 4 nghiệm tương ứng cho 4 hệ phương trình đồng dư:

$$s_1 = a \bmod q, \qquad s_1 = b \bmod q' \quad (3)$$
$$s_2 = a \bmod q, \qquad s_2 = -b \bmod q' \quad (4)$$
$$s_3 = -a \bmod q, \qquad s_3 = b \bmod q' \quad (5)$$
$$s_4 = -a \bmod q, \qquad s_4 = -b \bmod q' \quad (6)$$

Giải một trong 4 hệ phương trình trên ta sẽ tìm được s. Giả sử giải hệ phương trình (3), theo định lý CRT, ta có:
$s = (aq'(q'^{-1} \bmod q) + bq(q^{-1} \bmod q')) \bmod n$.
Như vậy, trong giai đoạn tạo chữ ký, ta chỉ cần chọn ngẫu nhiên số t sao cho $s' \in Q_n$.

2.3.1 Tính chất của lược đồ chữ ký đề xuất: Dựa vào các thuật toán đã được đề xuất, chúng tôi đưa ra các tính chất sau:

- Tính đúng đắn:
Định lý 1: Chữ ký $\sigma(r, s)$ được ký bởi $S(m)$ là chữ ký số hợp lệ tương ứng với thông điệp m.

Chứng minh.
Tính: $s' = s^2 \bmod n$
Ta có:

$$\begin{aligned} r &= g^t \bmod p \\ &= g^{s'+H(m||r)x} \bmod p \\ &= g^{s'+H(m||r)x} \bmod p \\ &= g^{s'} g^{H(m||r)x} \bmod p \\ &= g^{s'} y^{H(m||r)} \bmod p \\ &= r' \end{aligned}$$

Do đó chữ ký $\sigma(r, s)$ là chữ ký hợp lệ.
Định lý 2: Các chữ ký $(\sigma_1, \sigma_2, ..., \sigma_k)$ được ký bởi $S(m)$ là k chữ ký số hợp lệ nếu phương trình (2) đúng.
Chứng minh. Giả sử có k chữ ký (r_i, s_i) cho k thông điệp $m_i, i = 1, 2, ..., k$, được ký bởi cùng một người ký sử dụng khoá bí mật (x, q, q'). Để xác định k chữ ký này là đúng thì từng chữ ký phải thoả mãn phương trình:

$$r = g^{s'} y^{H(m||r)} \bmod p$$

Với $s' = s^2 \bmod n$.
Như vậy:

$$\begin{aligned} r_1 &= g^{s'_1} y^{H(m_1||r_1)} \bmod p \\ r_2 &= g^{s'_2} y^{H(m_2||r_2)} \bmod p \\ &\vdots \\ r_k &= g^{s'_k} y^{H(m_k||r_k)} \bmod p \end{aligned}$$

Do đó: $\prod_{i=1}^{k} r_i =$
$g^{s'_1} y^{H(m_1||r_1)} \dots g^{s'_k} y^{H(m_k||r_k)} \bmod p$
$= g^{s'_1+\dots+s'_k} y^{H(m_1||r_1)+\dots+H(m_k||r_k)} \bmod p$
$= g^{\sum_{i=1}^{k} s'_i} y^{\sum_{i=1}^{k} H(m_i||r_i)} \bmod p$

- Tính ngẫu nhiên:
  Định lý 3: Chữ ký $\sigma(r, s)$ được tạo ngẫu nhiên bởi người ký.
  Chứng minh. Trong lược đồ chữ ký được đề xuất, người ký chọn giá trị t ngẫu nhiên: $1 < t \leq n - 1$ và tính các giá trị r, s dựa trên t. Kẻ tấn công không thể ký một chữ ký hợp lệ (r, s) thay cho người ký ban đầu. Bởi vì, để nhận giá trị ngẫu nhiên t từ r và s là tính toán không khả thi. Nếu muốn nhận t từ r, kẻ tấn công phải giải bài toán logarit rời rạc trên $GF(p)$. Nếu muốn nhận t từ giá trị s kẻ tấn công phải giải bài toán phân tích số n thành q, q'.

2.3.2 Đánh giá độ an toàn của lược đồ xác thực lô: Trong phần này, chúng tôi đưa ra một số trường hợp mà kẻ tấn công có thể gây mất an toàn cho lược đồ xác thực lô.

- Trường hợp 1: Các hình thức tấn công trên lược đồ Rabin, Schnorr đều không thể thực hiện trên lược đồ xác thực lô đề xuất, bởi vì muốn phá vỡ lược đồ, kẻ tấn công phải giải đồng thời hai bài toán khó.
- Trường hợp 2: Theo lược đồ xác thực lô của Shao đề xuất, kẻ tấn công giải bài toán logarit rời rạc có thể tìm được khóa bí mật x dựa vào khoá công khai y, và tìm được giá trị t dựa vào giá trị r từ đó tính được giá trị s. Trong lược đồ xác thực lô đề xuất, kẻ tấn công muốn tính được s phải cần giá trị q, q'. Điều này đòi hỏi giải quyết bài toán khó là phân tích thừa số n. Như vậy, để có tất cả các thông tin bí mật từ chữ ký thì kẻ tấn công phải giải cả hai bài toán logarit rời rạc và bài toán phân tích thừa số.
- Trường hợp 3: Giả sử kẻ tấn công có thể giải quyết được bài toán phân tích thừa số, nghĩa là kẻ tấn công phân tích n và tìm được số q, q'. Tuy nhiên, chúng không thể tính được s do không thể tìm được x. Muốn tìm được x, kẻ tấn công phải giải bài toán logarit rời rạc trên trường $GF(p)$.

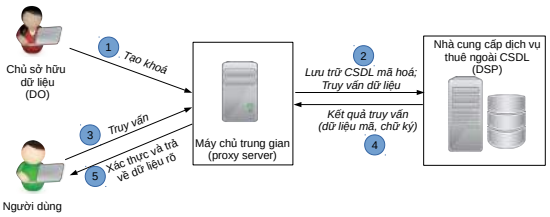
3. Xác thực dữ liệu mã hóa thuê ngoài

Trong phần này, chúng tôi đề xuất mô hình xác thực dữ liệu mã hoá trả về từ DSP là đúng đắn. Khi trả về dữ liệu truy vấn, DSP kèm theo các chữ ký của DO tạo ra trước khi lưu trữ, máy chủ DO sẽ tiến hành xác thực chữ ký của các dữ liệu tương ứng, nếu dữ liệu không bị thay đổi, DO giải mã và trả kết quả rõ cho người dùng.

3.1 *Mô hình xác thực*

Để xác thực dữ liệu trả về khi truy vấn dữ liệu thuê ngoài, chúng tôi đề xuất mô hình gồm bốn thành phần: chủ sở hữu dữ liệu (DO), nhà cung cấp dịch vụ (DSP), người dùng (Querier) và máy chủ trung gian (proxy server). Mô hình hoạt động xác thực được mô tả như hình 1

- Chủ sở hữu dữ liệu: là cá nhân, tổ chức thuê dịch vụ cơ sở dữ liệu từ DSP. DO là người tạo ra CSDL và ký trên dữ liệu bằng khoá bí mật của mình. DO chia sẻ dữ liệu cho người dùng.
- Nhà cung cấp dịch vụ: DSP cung cấp cho DO khả năng tạo, lưu trữ, cập nhật và truy vấn cơ sở dữ liệu trên máy chủ của họ.
- Người dùng: là cá nhân, tổ chức muốn truy cập CSDL của DO. Người dùng gửi câu truy vấn tới máy chủ DSP và nhận dữ liệu



Hình 1: *Mô hình xác thực cơ sở dữ liệu mã hoá thuê ngoài*

theo yêu cầu. Dữ liệu trả về đến người dùng là dữ liệu rõ.

- Máy chủ trung gian: là máy chủ thuộc quyền quản lý của DO, nằm ngoài phạm vi của DSP. Giả định máy chủ trung gian là tin cậy. Máy chủ trung gian không lưu trữ CSDL mà chỉ thực hiện nhiệm vụ xử lý xác thực, giải mã kết quả trả về cho người dùng.

Mô hình xác thực hoạt động theo 5 bước:

- Bước 1: DO tạo các khoá: khoá mã để mã hoá dữ liệu, khoá bí mật để tạo chữ ký và khoá công khai để xác thực chữ ký.
- Bước 2: DO mã hoá dữ liệu và tạo chữ ký trên dữ liệu, sau đó lưu trữ dữ liệu đã mã hoá và chữ ký lên DSP.
- Bước 3: Người dùng gửi câu truy vấn đến máy chủ DSP thông qua máy chủ trung gian để lấy dữ liệu theo nhu cầu. Máy chủ trung gian xử lý, chuyển đổi câu truy vấn người dùng rồi gửi câu truy vấn đến máy chủ DSP.
- Bước 4: Máy chủ DSP trả dữ liệu thoả điều kiện truy vấn kèm theo các chữ ký của dữ liệu về cho máy chủ trung gian.
- Bước 5: Máy chủ trung gian tiến hành xác thực dữ liệu, nếu dữ liệu đó hợp lệ sẽ tiến hành giải mã và trả về dữ liệu rõ cho người dùng.

3.2 *Quá trình hoạt động*

Quá trình xác thực CSDL mã hoá thuê ngoài thực hiện qua ba giai đoạn: giai đoạn tạo khoá (bước 1): Chủ sở hữu dữ liệu thực hiện chọn khoá để sử dụng trong toàn bộ quá trình tạo và xác thực dữ liệu. Việc tạo khoá được xử lý ngoài phạm vi của DSP và do DO quản lý; Giai đoạn lưu trữ dữ liệu thuê ngoài (bước 2) và giai đoạn xác thực dữ liệu truy vấn (bước 3, 4, 5).

Giả sử với một cơ sở dữ liệu D chứa nhiều bảng. Bảng T có m_T cột chứa n_T bản ghi $r = (r_{i1}, r_{i2}..., r_{im_T})$, với r_{ij} là dữ liệu tại dòng thứ i và cột thứ $j(1 \leq i \leq n_T, 1 \leq j \leq m_T)$. Chọn hai số nguyên tố lớn q, q' có độ dài 1024 bit. Tính $n = qq'$. Cho p là số nguyên tố có dạng $p = 2n + 1$. g là phần tử sinh thuộc Z_p^* có cấp là n sao cho $g^n = 1 \pmod p$. Các giai đoạn hoạt động xác thực CSDL thuê ngoài được thực hiện như sau:

- Giai đoạn tạo khoá:

Thuật toán 5 Tạo khoá công khai, khoá bí mật

1: Chọn ngẫu nhiên k có độ dài 128 bit làm khoá riêng dùng để mã hoá/giải mã dữ liệu.
2: Chọn ngẫu nhiên x với $x \in Z_n^*$
3: Tính $y = g^x \bmod p$.
4: Khoá công khai là (n, y). Khoá bí mật là (k, x, q, q')

- Giai đoạn lưu trữ dữ liệu: Với mỗi bảng $T \in D$:

Thuật toán 6 Lưu CSDL mã hoá

Input: Bảng T trong CSDL, khoá bí mật.
Output: Lưu bảng dữ liệu mã hoá kèm chữ ký lên máy chủ DSP.
Method:

$n = qq'$
for $i = 1$ **to** n_T **do**
  for $j = 1$ **to** m_T **do**
    $\mu_{ij} \leftarrow E_k(r_{ij})$
    while True **do**
      Chọn ngẫu nhiên $t \in Z_n^*$
      $r \leftarrow g^t \bmod p$
      $h \leftarrow H(\mu_{ij}||r)$.
      $s' \leftarrow t - hx \pmod n$.
      $a = s'^{(q+1)/4} \bmod q$
      $b = s'^{(q'+1)/4} \bmod q'$
      $s = (aq'(q'^{-1} \bmod q) + bq(q^{-1} \bmod q')) \bmod n$
      if $(s^2 \bmod n == s')$ **then**
        break
      end if
    end while
    $\sigma_{ij} \leftarrow (r, s)$.
    $d_i \leftarrow \{\mu_{ij}, \sigma_{ij}\}$
  end for
  $T_i' \leftarrow \{d_i\}$
end for
Lưu trữ bảng T' lên máy chủ DSP

• Giai đoạn xác thực dữ liệu truy vấn:

Khi người dùng gửi câu truy vấn đến DSP thông qua máy chủ proxy thì máy chủ DSP trả dữ liệu $T_r = \{\mu_{ij}, \sigma_{ij} | i = 1, 2, ...h_T; j = 1, 2, \ldots, k_T\}$ về máy chủ trung gian.

Thuật toán 7 Xác thực chữ ký và giải mã dữ liệu

Input: Bảng T_r, khoá k, khóa công khai.
Output: Bảng dữ liệu rõ cho người dùng.
Method:

for $i = 1$ **to** h_T **do**
  $u = 0, v = 0, r = 1$
  for $j = 1$ **to** k_T **do**
    $s' = s_{ij}^2 \bmod n$
    $u+ = s'$
    $v+ = H(\mu_{ij}||r_{ij})$
    $r* = r_{ij}$
    $a_{ij} = D_k(\mu_{ij})$
    $d_i \leftarrow \{a_{ij}\}$
  end for
  if $(r \bmod p == g^u y^v \bmod p)$ **then**
    $T_i \leftarrow \{d_i\}$
  end if
end for
Trả dữ liệu T về cho người dùng

3.3 *Một số trường hợp truy vấn CSDL*

3.3.1 Cập nhập dữ liệu: Các bản ghi của bảng T chứa dữ liệu mã và chữ ký. Những bản ghi này không phụ thuộc lẫn nhau nên khi thực hiện thao tác thêm, sửa, xoá thì dữ liệu được xử lý độc lập và cập nhập vào CSDL mà không cần phải tính toán, xây dựng lại bản ghi khác như các cấu trúc xác thực ADS.

Khi lưu trữ lên máy chủ DSP, bảng T có dạng $T = \{\mu_{ij}, \sigma_{ij} | i = 1...n_T, j = 1...m_T\}$. Giả sử muốn thêm bản ghi $r' = (r'_1, r'_2..., r'_{m_T})$ vào bảng T đầu tiên ta tính $\mu'_j = \{E_k(r'_j) | j = 1...m_T\}, \sigma'_j = \{S_x(\mu'_j) | j = 1...m_T\}$ sau đó thực hiện câu lệnh truy vấn "INSERT INTO T VALUES(μ'_1, $\sigma'_1, \mu'_2, \sigma'_2, ..., \mu'_{m_T}, \sigma'_{m_T}$);".

Khi thực hiện xóa dữ liệu thì người dùng gửi câu truy vấn "DELETE * FROM T WHERE <điều kiện>". Máy chủ DSP sẽ xoá bản ghi thoả mãn <điều kiện> mà không thực hiện tính toán lại các cấu trúc liên quan đến bản ghi bị xoá như khi sử dụng ADS. Như vậy, việc thao tác cập nhập dữ liệu thực hiện dễ dàng và không ảnh hưởng đến các bản ghi trong bảng. Điều này thích hợp với CSDL động mà mô hình ADS rất khó giải quyết.

3.3.2 Truy vấn dữ liệu từ nhiều bảng: Giả sử CSDL có hai bảng $T_1 = \{id1, \sigma_{id1}, \mu 1_{ij}, \sigma 1_{ij} | i = 1...n_T, j = 1...m_T\}, T_2 = \{id2, \sigma_{id2}, \mu 2_{ij}, \sigma 2_{ij} | i = 1...h_T, j = 1...k_T\}$. Người dùng gửi câu truy vấn "SELECT * FROM T_1 INNER JOIN T_2 ON $T_1.id1 = T_2.id2$;" đến DSP. Máy chủ DSP xử lý và trả kết quả $T_r = \{id1, \sigma_{id1}, \mu 1_{ij}, \sigma 1_{ij}, id2, \sigma_{id2}, \mu 2_{ij}, \sigma 2_{ij} | i = 1, 2, ...l_T; j = 1, 2, ..., p_T\}$ về máy chủ trung gian. Do $\sigma_{id1}, \sigma_{id2}, \sigma 1_{ij}, \sigma 2_{ij}$ được được tạo ra trên cùng khoá bí mật của DO nên ta có thể xác thực bằng thuật toán 7 mà không cần các thông tin phụ trợ kèm theo như các cấu trúc ADS.

4. Phân tích, đánh giá hiệu năng của phương pháp đề xuất

Trong phần này, chúng tôi tiến hành thử nghiệm thời gian thực hiện xác thực chữ ký số của lược đồ đề xuất và phân tích, đánh giá độ phức tạp thời gian tính toán của ba giai đoạn hoạt động xác thực CSDL thuê ngoài. Trong quá trình thử nghiệm, các thuật toán được cài đặt bằng ngôn ngữ Python và được thực hiện trên máy tính Core™ i3-2375M CPU@1.50GHz x 4, Ram 8GB, hệ điều hành Ubuntu 18.04.

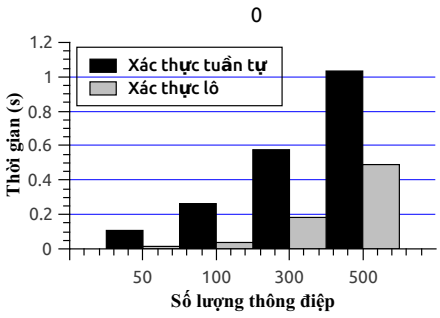
Thời gian thực hiện tạo chữ ký được mô tả trong bảng 2.

Bảng 2: *Thời gian thực hiện tạo chữ ký*

Số lượng thông điệp	Thời gian ký (s)
5	0,190114
100	1,22749
300	3,158304
500	5,548609

Giai đoạn tạo chữ ký và lưu trữ dữ liệu chỉ thực hiện một lần duy nhất khi DO lưu trữ dữ liệu lên DSP nên không ảnh hưởng đến hiệu năng của hệ thống khi truy vấn dữ liệu.

Để thử nghiệm xác thực chữ ký số, chúng tôi tiến hành trên hai phương pháp: xác thực lần lượt k chữ ký theo thuật toán kiểm tra chữ ký σ và phương pháp xác thực lô. Số lượng các thông điệp tương ứng với các trường dữ liệu cần xác thực. Theo hình 2, khi số lượng thông điệp là 100, thì thời gian xác thực tuần tự từng chữ ký là 0.263187 gấp 7 lần thời gian xác thực lô



Hình 2: *Thời gian thực hiện xác thực chữ ký*

là 0.037306. Do đó, khi bảng dữ liệu có nhiều bản ghi, thời gian xác thực tuần tự sẽ tăng đáng kể so với xác thực lô.

Thời gian thực hiện của các giai đoạn xác thực CSDL thuê ngoài phụ thuộc vào số lượng các phép tính toán. Chúng tôi quy ước độ phức tạp thời gian tính toán của các phép toán như trong bảng 3. Thời gian tính toán cộng, trừ, nhân toán học có thể bỏ qua vì nó nhỏ hơn nhiều so với thời gian tính luỹ thừa modulo, nhân modulo.

Bảng 3: *Ký hiệu thời gian thực hiện phép toán*

Ký hiệu	Ý nghĩa
T_{mul}	Thời gian tính phép nhân modulo
T_{exp}	Thời gian tính phép luỹ thừa modulo
T_S	Thời gian tạo chữ ký
T_E	Thời gian mã hoá
T_D	Thời gian giải mã
T_H	Thời gian thực hiện phép băm

Giả sử CSDL có t_1 bảng, mỗi bảng có tối đa n_T dòng và m_T cột. Khi truy vấn dữ liệu, máy chủ DSP trả về t_2 bảng, mỗi bảng có tối đa h_T dòng, k_T cột. Thời gian tính toán tối đa của các giai đoạn được mô tả như bảng 4.

Trong đó, thời gian tạo chữ ký T_S được thực hiện ngẫu nhiên chọn số t sao cho $s' \in Q_n$. Do $n = qq'$, nên $|Q_n| = (q-1)(q'-1)/4$. Như vậy, khả năng s' là số chính phương modulo n là:

$P(s' \in Q_n) = \frac{(q-1)(q'-1)}{4n}$

$\Rightarrow T_S = T_H + T_{P(s' \in Q_n)}(6T_{mul} + 3T_{exp} + 2T_{inv})$ với $T_{P(s' \in Q_n)}$ là thời gian chọn $s' \in Q_n$.

Trong thực tế, số lượng bảng trong CSDL không nhiều (thường không quá 100 bảng), và số cột không lớn (số cột biểu thị cho thuộc tính của đối tượng nên thường không quá 100 thuộc tính). Do đó, các giai đoạn xử lý phụ thuộc chủ yếu vào số dòng dữ liệu của từng bảng. Mặc khác, các giai đoạn này thực hiện tạo chữ ký và xác thực đồng thời với thao tác mã hoá/giải mã dữ liệu nên việc tạo chữ ký/xác thực xem như không đáng kể so với không tạo chữ ký. Ngoài ra, mô hình đề xuất có thể kết hợp với xử lý song song sẽ giảm đáng kể thời gian truy vấn CSDL.

Bảng 4: *Thời gian thực hiện các giai đoạn xác thực CSDL thuê ngoài*

Giai đoạn	Thời gian thực hiện
Tạo khoá	T_{exp}
Lưu trữ dữ liệu	$t_1 n_T m_T (T_E + T_S)$
Xác thực dữ liệu	$t_2 h_T (k_T (T_{mul} + T_D + T_H) + 2T_{exp} + T_{mul})$

5. Kết luận

Trong bài báo này, chúng tôi đã giới thiệu về vấn đề xác thực CSDL thuê ngoài, các lược đồ chữ ký số, đưa ra định nghĩa về xác thực lô. Đóng góp của bài báo là đề xuất thuật toán xác thực lô dựa trên hai bài toán khó là logarit rời rạc và phân tích số. Đồng thời đề xuất mô hình kiểm tra tính đúng đắn của CSDL được mã hoá. Mô hình này có thể kiểm tra tính đúng đắn của cơ sở dữ liệu mã hoá thuê ngoài, trả về kết quả rõ cho người dùng. Điều này giúp cho việc khi bản ghi bị lỗi hoặc chỉnh sửa, thì DO biết được dữ liệu không toàn vẹn từ đó có chính sách xử lý phù hợp. Bên cạnh đó, mô hình của chúng tôi hỗ trợ các loại truy vấn từ nhiều bảng và phù hợp với CSDL động. Hơn nữa, khi kiểm tra kết hợp với giải mã dữ liệu nên tốc độ xử lý hầu như không khác biệt so với không kiểm tra chữ ký (do vẫn phải giải mã dữ liệu). Hướng nghiên cứu tiếp theo là đề xuất mô hình kiểm tra tính đầy đủ và tính mới của dữ liệu trả về.

TÀI LIỆU THAM KHẢO

[1] Y. Zhang, J. Katz, and C. Papamanthou, "Integridb: Verifiable SQL for outsourced databases," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1480–1491.

[2] M. S. Niaz and G. Saake, "Merkle Hash Tree based Techniques for Data Integrity of Outsourced Data." in *GvD*, 2015, pp. 66–71.

[3] R. Jain and S. Prabhakar, "Trustworthy data from untrusted databases," in *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*. IEEE, 2013, pp. 529–540.

[4] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," *ACM Transactions on Storage (TOS)*, vol. 2, no. 2, pp. 107–138, 2006.

[5] J. Yuan and S. Yu, "Flexible and publicly verifiable aggregation query for outsourced databases in cloud," in *Communications and network security (cns), 2013 ieee conference on*. IEEE, 2013, pp. 520–524.

[6] P. FIPS, "186," *Digital Signature Standard*, vol. 1, 1994.

[7] R. GOST, "R 34.10-94. Russian Federation Standard," *Information Technology. Cryptographic data Security. Produce and check procedures of Electronic Digital Signature based on Asymmetric Cryptographic Algorithm. Government Committee of the Russia for Standards*, 1994.

[8] C. H. Lim and P. J. Lee, "A study on the proposed Korean digital signature algorithm," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 1998, pp. 175–186.

[9] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Conference on the Theory and Application of Cryptology*. Springer, 1989, pp. 239–252.

[10] H. Morita, J. C. Schuldt, T. Matsuda, G. Hanaoka, and T. Iwata, "On the security of the schnorr signature scheme and DSA against related-key attacks," in *International Conference on Information Security and Cryptology*. Springer, 2015, pp. 20–35.

[11] L. Harn, "Batch verifying multiple DSA-type digital signatures," *Electronics Letters*, vol. 34, no. 9, pp. 870–871, 1998.

[12] D. Naccache, D. M'RaÏhi, S. Vaudenay, and D. Raphaeli, "Can DSA be improved?—Complexity trade-offs with the digital signature standard—," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1994, pp. 77–85.

[13] C. H. Lim and P. J. Lee, "Security of interactive DSA batch verification," *Electronics letters*, vol. 30, no. 19, pp. 1592–1593, 1994.

[14] S.-M. Yen and C.-S. Laih, "Improved digital signature suitable for batch verification," *IEEE Transactions on Computers*, vol. 44, no. 7, pp. 957–959, 1995.

[15] C. Boyd and C. Pavlovski, "Attacking and repairing batch verification schemes," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2000, pp. 58–71.

[16] M. Bellare, J. A. Garay, and T. Rabin, "Fast batch verification for modular exponentiation and digital signatures," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1998, pp. 236–250.

[17] Y. Zhou, X. Zhu, and Y. Fang, "MABS: Multicast authentication based on batch signature," *IEEE transactions on Mobile Computing*, vol. 9, no. 7, pp. 982–993, 2010.

[18] Z. Shao, "Batch verifying multiple DSA-type digital signatures," *Computer Networks*, vol. 37, no. 3-4, pp. 383–389, 2001.

[19] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.