Số nguyên tố an toàn trong các giao thức DH-KE

Nguyễn Thanh Sơn

***Tóm tắt*—Việc sinh các số nguyên tố "an toàn" p, mà ở đó tất cả các ước nguyên tố khác 2 của $p-1$ đều là ước nguyên tố lớn, là hết sức cần thiết để tránh các tấn công nhóm con nhỏ được chỉ ra bởi hai tác giả Chao Hoom Lim và Pil Joong Lee. Một thuật toán hiện có để sinh các số nguyên tố như vậy cũng đã được trình bày bởi hai tác giả này. Tuy nhiên, hạn chế của phương pháp đó là thuật toán không phải khi nào cũng trả về được một số nguyên tố an toàn. Một phần lý do cho vấn đề này là vì thuật toán không (và khó có thể) được phân tích và đánh giá kỹ lưỡng về mặt toán học. Do đó, mục đích chính của bài báo là đề xuất một thuật toán mới để sinh các số nguyên tố an toàn và kèm theo các đánh giá chi tiết về mặt toán học.**

***Abstract*—The generate of "safe" primes p, where all prime divisors of $p-1$ are large prime divisors, is essential to avoid small subgroup attacks which are point out by two authors Chao Hoom Lim and Pil Joong Lee. An existing algorithm for generating such primes has also been presented by these two authors. However, the drawback of that method is that the algorithm does not always return safe prime numbers. Part of the reason for this is that the algorithm is not (and hardly) be thoroughly analyzed and evaluated mathematically. Therefore, the main purpose of this paper is to propose a new algorithm for generating safe prime numbers, including detailed mathematical evaluations.**

Từ khóa—Thuật toán sinh số nguyên tố an toàn; giao thức DH-KE.

Keywords—Safe prime generation algorithm; DH-KE protocol.

I. ĐẶT VẤN ĐỀ

Đối với các ứng dụng mật mã như hệ mật khóa công khai và lược đồ chữ ký số có độ an toàn dựa trên độ khó của bài toán logarit trên $GF(p)$ thì bộ tham số (p, q, g) (với p là số nguyên tố, q là ước nguyên tố của $p-1$ và $ord(g) = q$) được lựa chọn để sử dụng chỉ cần chống được các tấn công theo các phương pháp giải bài toán logarit như Pollard Lambda, Pollard Rho [3], Pollig Hellman [1],… Cụ thể, các tham số p, q được khuyến cáo để sử dụng trong các chuẩn mật mã đều phải là các số nguyên tố lớn. Chẳng hạn, trong chuẩn FIPS PUB 186-3 [4] quy định p có kích thước tối thiểu là 1024-bit và độ dài bit của q tương ứng là 160-bit.

Khi ứng dụng trong thực tế các lược đồ chữ ký số trong các giao thức thỏa thuận khóa kiểu Diffie-Hellman (DH-KE) đã nảy sinh một kiểu tấn công của chính người tham gia hệ thống nhằm tìm khóa mật của người còn lại. Loại tấn công này được Chao Hoom Lim và Pil Joong Lee (Lim-Lee) công bố năm 1997 [2] và được các tác giả của [11] nghiên cứu, xem xét để đề xuất tiêu chuẩn cho tham số modulo p. Cũng để chống lại tấn công trên, năm 2006, Tổ chức Tiêu chuẩn hóa quốc tế (ISO) đã ban hành chuẩn ISO/IEC 11770-4:2006 (xem ISO/IEC 11770-4:2006, Clause 5) được phát biểu như sau:

Tiêu chuẩn. (Tiêu chuẩn về sự phân rã $p-1$)

Số nguyên tố p dùng trong các giao thức thỏa thuận khóa DH-KE với phần tử sinh $g \in GF(p)$ có cấp bằng q, phải thỏa mãn các điều kiện sau:

a) $p = 2qq_1q_2 \ldots q_k + 1$ *với* q_i *là các số nguyên tố (không nhất thiết khác nhau).*

b) $q_i > q$ $(1 \le i \le k)$.

(1)

Trong Phần II, tác giả chỉ ra rằng: Để chống được tấn công của Lim-Lee thì điều kiện (b) chỉ cần là:

$$q_i \ge q \ (1 \le i \le k). \qquad (2)$$

Trên cơ sở đó đưa ra định nghĩa về bộ tham số (p, q, g) an toàn cho DH-KE như sau.

Định nghĩa 1. *Bộ tham số (p, q, g) ngoài việc thỏa mãn việc giải bài toán logarit rời rạc theo cơ số g là khó còn thỏa mãn thêm các điều kiện*

Bài báo được nhận ngày 12/7/2020. Bài báo được nhận xét bởi phản biện thứ nhất ngày 27/7/2020 và được chấp nhận đăng ngày 27/7/2020. Bài báo được nhận xét bởi phản biện thứ hai ngày 03/8/2020 và được chấp nhận đăng ngày 03/8/2020.

(1) và (2) được gọi là bộ tham số an toàn cho giao thức DH-KE trên $GF(p)$.

Cũng trong công trình của mình [2], Lim-Lee đã đưa ra một thuật toán mang tính thực nghiệm để sinh các số nguyên tố an toàn như vậy. Thuật toán này sau đó cũng đã được sử dụng để sinh các số nguyên tố an toàn trong các thư viện mật mã như Miracl [9], PGP [6], GNU PG [8] và Gutmann's cryptlib [7]. Tuy nhiên như đã đề cập, thuật toán được đề xuất bởi Lim-Lee mới chỉ mang ý nghĩa về thực nghiệm. Điều này là do thuật toán của hai tác giả trên không phải khi nào cũng trả về kết quả và cũng không được phân tích chi tiết về mặt toán học. Do vậy, mục tiêu của bài báo này là đề xuất một thuật toán sinh số nguyên tố "an toàn" mới và kèm theo đảm bảo toán học cho nó. Cụ thể, thuật toán được đề xuất sẽ được trình bày trong Phần III và các phân tích về tính đúng đắn, độ phức tạp của thuật toán này sẽ được đánh giá trong Phần IV.

II. SỐ NGUYÊN TỐ AN TOÀN TRONG CÁC GIAO THỨC DH-KE

A. Độ phức tạp của tấn công tìm khóa mật trong giao thức DH-KE của Lim-Lee

Để hiểu rõ hơn về việc cần sử dụng bộ tham số an toàn trong các giao thức DH-KE, trong mục này sẽ trình bày lại tấn công của hai tác giả trên, đưa ra những phân tích để lý giải cho chuẩn này.

Trong bài viết của mình, Lim-Lee đã thực hiện việc tấn công[1] lên họ giao thức trao đổi khóa MTI [5] với kết quả thu được là:

Bổ đề 1. (Độ phức tạp của tấn công tìm $x_B \bmod u$)

Với u là ước bất kỳ của p – 1 thì chỉ sau $x_B \bmod u$ (hoặc $k_B \bmod u$) bước tính toán thì A sẽ tìm được $x_B \bmod u$ (hoặc $k_B \bmod u$) với x_B là khóa ký của B (hoặc k_B là khóa ngắn hạn theo phiên do B thực hiện trong giao thức).

Nói một cách khác, độ phức tạp của tấn công Lim-Lee bằng O(u).

[1] Hơn thế nữa, tấn công của Lim-Lee có thể áp dụng lên giao thức trao đổi khóa HMQV, giao thức trao đổi khóa KEA+ trong trường hợp số nguyên tố *p* không phải là số nguyên tố an toàn.

Như vậy việc tìm x_B (hoặc k_B), chỉ cần tìm nốt $x_B \; div \; u$ (hoặc $k_B \; div \; u$). Việc này chính là thực hiện giải bài toán sau.

Bài toán 1.

Cho $g \in GF(p)$ với $ord(g) = q$ và $u > 1$ là một số nguyên bất kỳ sao cho $gcd(u, q) = 1$. Xét phương trình: $b = g^x \bmod p$. (3)

Hãy tìm x khi đã biết $x_0 = x \bmod u$.

Cách giải bài toán 1

Ký hiệu $B = b.g^{-x_0} \bmod p$, $G = g^u \bmod p$ và $x_1 = x \; div \; u$ thì (3) trở thành

$$B = G^{x_1} \bmod p.$$

Do từ $x < q$ nên $x_1 < q/u$ nên nếu lấy $m = \lceil \sqrt{q/u} \rceil$ thì $x_1 = x'_1 + x"_1.m$ với $0 \le x'_1$, $x"_1 < m$. Với việc sử dụng phương pháp của Daniel Shank, sẽ tìm được x_1 với không quá m phép nhân trong $GF(p)$ và lưu trữ m phần tử của $GF(p)$.

Cùng với Bổ đề 1, ta thu được kết quả sau.

Kết quả 2. (Độ phức tạp của tấn công tìm x_B)

Độ phức tạp của tấn công tìm x_B là $max\{u, \lceil \sqrt{q/u} \rceil\}$. (4)

Chú ý 1. *Trong các giao thức DH-KE có xác thực thì khi biết khóa ngắn hạn theo phiên k_B, từ lược đồ chữ ký được sử dụng trong giao thức nên luôn tính được khóa bí mật x_B.*

B. Vai trò của bộ tham số an toàn trong giao thức DH-KE

Trong mục này chỉ ra kết luận sau:

Kết luận 3. *Nếu (p, q, g) là bộ tham số an toàn theo Định nghĩa 1, thì tấn công Lim-Lee để tìm khóa mật x_B là không thể (theo nghĩa người tấn công không thể thực hiện được trong thời gian thực).*

Chứng minh

Biết rằng trong các ứng dụng mật mã có độ an toàn dựa trên tính khó giải của bài toán logarit trên $GF(p)$ thì tham số q cần được chọn sao cho việc giải bài toán logarit rời rạc trên $GF(p)$ là không thể. Trong đó việc thực hiện $\lceil \sqrt{q} \rceil$ đối với người giải là không thể.

Các tham số khóa mật x_B và khóa ngắn hạn theo phiên k_B luôn được chọn đủ lớn, ít nhất là không thể tìm được theo phương pháp vét cạn (việc thực hiện x_B phép toán cơ bản là không thể đối với người giải).

Với (p, q, g) là an toàn thì chỉ xảy ra hai khả năng đối với u đó là:

- u = 2. Khi đó, theo (4) thì tấn công Lim-Lee có chi phí thực hiện tính toán là $\lceil\sqrt{q/2}\rceil \approx \lceil\sqrt{q}\rceil$ nên sẽ không thể.
- $u \neq 2$. Theo điều kiện (2) thì $u \geq q$ nên $x_B \bmod u = x_B$, vì vậy tấn công Lim-Lee cũng không thể.

Như vậy, Kết luận 3 đã được chứng minh.■

III. THUẬT TOÁN SINH CÁC SỐ NGUYÊN TỐ AN TOÀN

Trong phần này đưa ra một thuật toán sinh các cặp số nguyên tố p, q có các độ dài tương ứng $len(p) = L, len(q) = N$ sao cho $q \mid (p - 1)$ và p thỏa mãn yêu cầu về số nguyên tố an toàn đã đưa ra trong Định nghĩa 1.

A. Một số ký hiệu

Cho $S \to \{a_j\}_{j=1,\ldots,m}$ là một tập hữu hạn. Khi đó:

- Việc lấy ngẫu nhiên một phần tử a trong S ký hiệu $a = Random(S)$ (hoặc $a \in_R S$).
- Hàm $Next: S \to S$ được xác định như sau

$$Next_S(a_j) = \begin{cases} a_{j+1} & \text{nếu} \quad j < m \\ a_1 & \text{nếu} \quad j = m \end{cases}$$

Cho S là tập các số tự nhiên. Khi đó:

- Tập các số nguyên a thỏa mãn $A \leq a \leq B$ ký hiệu là $[A, B]$.
- Tập các số nguyên tố trong S ký hiệu là $Prime(S)$, đặc biệt nếu $S = [A, B]$ thì tập $Prime([A, B])$ được viết gọn lại là $Prime(A, B)$.
- Hàm $Next_S: S \to S$ được xác định như sau:

$$\text{Hàm } Next_{[A,B]}(a) = \begin{cases} a + 1 & \text{nếu} \quad a < B \\ A & \text{nếu} \quad a = B \end{cases}$$

- Hàm $Random(Prime(A, B))$ được thực hiện theo thuật toán sau:

Thuật toán 1.

Input: $[A, B]$.

Output: $p \in Prime(A, B)$.

1. $p \in_R [A, B]$
2. while (p is not prime)

   $p \leftarrow Next_{[A,B]}(p)$.
3. return p

- Hàm $Next_{\text{Prime(A,B)}}()$ được thực hiện theo thuật toán sau:

Thuật toán 2.

Input: $p \in Prime(A, B)$.

Output: $q \in Prime(A, B)$ thỏa mãn định nghĩa của hàm $Next$ với việc đánh số các số nguyên tố trong $Prime(A, B)$ theo thứ tự tăng dần.

1. $q \leftarrow Next_{[A,B]}(p)$
2. while (q is not prime)

   $q \leftarrow Next_{[A,B]}(q)$
3. return q

Chú ý 2.

a) Do phải duyệt toàn bộ các hợp số giữa số nguyên tố đầu vào đến số nguyên tố đầu ra đối với Thuật toán 2, trong khi đối với Thuật toán 1 chỉ cần duyệt từ một số cho đến số nguyên tố đầu ra cho nên chi phí trung bình để thực hiện hàm $\text{Next}_{\text{Prime(A,B)}}(.)$ là gấp đôi chi phí trung bình để thực hiện hàm $Random(Prime(A, B))$. Như vậy nếu ký hiệu: $\rho = \frac{\#[A,B]}{\#\text{Prime(A,B)}}$ và gọi là khoảng cách trung bình giữa hai số nguyên tố trong $[A, B]$ thì việc thực hiện hàm $Random(Prime(A, B))$ chỉ cần trung bình $\frac{\rho}{2}$ phép kiểm tra tính nguyên tố, còn hàm $Next_{Prime(A,B)}(.)$ cần trung bình ρ phép kiểm tra tính nguyên tố.

b) Đối với Thuật toán 1, nếu $Prime(A, B) = \emptyset$ thì sẽ không dừng. Để tránh lỗi trên ta có thể kiểm tra việc đã duyệt toàn bộ các số nguyên trong $[A, B]$ hay chưa, chẳng hạn có thể sửa Thuật toán 1 thành Thuật toán 1a như sau.

Thuật toán 1a.

Input: $[A, B]$.

Output: $p \in Prime(A,B)$ nếu $Prime(A,B) \neq \emptyset$. Ngược lại đưa ra thông báo "$Prime(A,B) = \emptyset$".

1. $p \in_R [A,B]$; stop $\leftarrow p$; ok $\leftarrow 0$
2. ok $\leftarrow$ (p is prime).
3. if (ok) then return p
4. else then:

4.1 $p \leftarrow Next_{[A,B]}(p)$

4.2 ok $\leftarrow$ (p is prime)

4.3 while (ok $= 0$) and (stop $\neq p$)

4.3.1 $p \leftarrow Next_{[A,B]}(p)$

4.3.2 ok $\leftarrow$ (p is prime)

5. if (ok $= 0$) then return "$Prime(A,B) = \emptyset$"
6. else return p

Kỹ thuật trên cũng có thể áp dụng khi sử dụng hàm $Next_{Prime(A,B)}$ nhiều lần với thêm vào đầu ra thông báo "Đã duyệt hết các phần tử trong $Prime(A,B)$" trong trường hợp đầu ra của hàm $Next_{Prime(A,B)}$ ở lần thực hiện thứ *j* nào đó đúng bằng đầu vào của hàm này trong lần thực hiện đầu tiên.

B. Thuật toán sinh các số nguyên tố an toàn

Thuật toán 3.

Input: L, N là hai số nguyên $N > 1$ và $L \geq 2(N+1)$.

Output: p, q là hai số nguyên tố, thỏa mãn $len(p) = L$, $len(q) = N$ và p là số nguyên tố an toàn theo Định nghĩa 1.

[*Sinh số nguyên tố* q]

1. $q \leftarrow Random\big(Prime(2^{N-1}, 2^N - 1)\big)$.
2. $f_0 \leftarrow 2 * q$, $M_0 \leftarrow len(f_0)$.

[*Xác định số ước* q_i]

3. $k \in_R \left[1, \left\lfloor \frac{L - M_0 - 1}{N} \right\rfloor\right]$.

//Từ $L \geq 2(N+1)$ và $M_0 = N + 1$ nên $L - M_0 - 1 \geq N$.

[*Sinh* $k-1$ *số nguyên tố* q_i *với* $i = 1, \dots, k-1$]

4. $i \leftarrow 1$.
5. while ($i < k$) do

5.1 $N_i \in_R [N, L - M_{i-1} - (k-i) * N - 1]$

5.2 if ($N_i = N$) then

$q_i \leftarrow Random(Prime(q, 2^N - 1))$;

5.3 else

$q_i \leftarrow Random(Prime(2^{N_i - 1}, 2^{N_i} - 1))$

5.4 $f_i \leftarrow f_{i-1} * q_i$; $M_i \leftarrow len(f_i)$

5.5 $i \leftarrow i + 1$

[*Sinh số nguyên tố* p]

6. $A \leftarrow \left\lceil \frac{2^{L-1}}{f_{k-1}} \right\rceil$; $B \leftarrow \left\lfloor \frac{2^L - 1}{f_{k-1}} \right\rfloor$
7. $q_k \leftarrow Random(Prime[A,B])$
8. $p \leftarrow f_{k-1} * q_k + 1$
9. while p is not prime do:

9.1 $q_k \leftarrow Next_{Prime[A,B]}(q_k)$

9.2 $p \leftarrow f_{k-1} * q_k + 1$

10. return (p, q)

Chú ý 3. *Theo phần a) của Chú ý 2 thì bước 9.1 của thuật toán có thể được thay bằng* $q_k \leftarrow Random(Prime[A,B])$ *sẽ hiệu quả hơn. Tuy nhiên trong trường hợp không tồn tại số nguyên tố* $p = f_{k-1} * q_k + 1$ *với mọi* $q_k \in Prime[A,B]$ *thì ta sẽ không có giải pháp như đã nêu trong phần b) của Chú ý 2 để quyết định dừng thuật toán với đầu ra là thông báo "Null".*

IV. PHÂN TÍCH THUẬT TOÁN 3

A. Một số kết quả hỗ trợ

Bổ đề 4. *Với mọi* $1 \leq i < k$ *thì*

$$L - M_{i-1} - 1 \geq (k - i + 1)N \qquad (5)$$

Chứng minh

Trước tiên theo bước 2 và bước 5.4 thì:

$M_0 = len(2 * q)$

và $M_i = len(2 * q * q_1 * \dots * q_i)$. (6)

Còn theo các bước 5.2 và 5.3 thì

$len(q_i) = N_i$. (7)

Tiếp theo sẽ chứng minh bất đẳng thức (5) bằng phương pháp quy nạp như sau.

Với $i = 1$, theo công thức tính k tại bước 3 là $k = Random\left[1, \left\lfloor\frac{L-M_0-1}{N}\right\rfloor\right]$ tức là $k \le \left\lfloor\frac{L-M_0-1}{N}\right\rfloor$ hay $L - M_{i-1} - 1$

$= L - M_0 - 1 \ge kN$

$= (k - i + 1)N.$

Chứng tỏ bất đẳng thức (5) đã đúng với $i = 1$.

Giả thiết quy nạp là (5) đã đúng đến i với $1 \le i < k - 1$,

xét $L - M_{(i+1)-1} - 1 = L - M_i - 1.$ (8)

Từ (6) và (7) ta có:

$M_i = len(2 * q * q_1 * ... * q_i)$

$\le len(2 * q * q_1 * ... * q_{i-1}) + len(q_i)$

$= M_{i-1} + N_i.$ (9)

Mặt khác, theo bước 5.1 thì

$N_i \le L - M_{i-1} - 1 - (k - i)N$

hay $L - M_{i-1} - N_i - 1 \ge (k - i)N$

$= (k - (i + 1) + 1)N.$ (10)

Thay (9) vào vế phải của (8) thì vế phải này trở thành vế trái của (10), nên theo bất đẳng thức này có:

$L - M_{(i+1)-1} - 1 \ge L - M_{i-1} - N_i - 1$

$\ge (k - (i + 1) + 1)N.$

Bất đẳng thức trên cho thấy (5) đã đúng với $i + 1$, vậy bổ đề đã được chứng minh.■

Nhắc lại định lý Gauss và định lý Drichlet được trình bày tại [10] về các số nguyên tố.

Định lý Gauss.

Ký hiệu $\pi(x)$ là số các số nguyên tố $\le x$. Ta có:

$$\pi(x) \sim \frac{x}{lnx}$$

theo nghĩa $\lim_{x\to\infty} \frac{x}{\pi(x)lnx} = 1.$

Như vậy với x đủ lớn, ta có thể viết

$$\pi(x) \approx \frac{x}{lnx}$$

Định lý Drichlet.

Cho A và B là hai số tự nhiên thỏa mãn $gcd(A, B) = 1$. Ký hiệu $\pi_{A,B}(x)$ là số các số nguyên tố $p = t.A + B \le x$ $(t = 0, 1, 2, ...)$.

Ta có: $\pi_{A,B}(x) \sim \frac{1}{\varphi(A)}\pi(x).$

Như vậy, với x đủ lớn có thể viết:

$$\pi_{A,B}(x) \approx \frac{1}{\varphi(A)}\pi(x).$$

Từ Bổ đề 4, thu được một số hệ quả sau.

Hệ quả 5. *Các tập sử dụng trong bước 5.1 là khác rỗng, tức là*

$[N, L - M_{i-1} - (k - i) * N - 1] \ne \emptyset.$

Chứng minh

Theo (5) thì $L - M_{i-1} - 1 \ge (k - i + 1)N$

$\Leftrightarrow L - M_{i-1} - (k - i)N \ge N$

Suy ra, có tập

$[N, L - M_{i-1} - (k - i) * N - 1] \ne \emptyset.$■

Hệ quả 6. *Mọi số nguyên tố q_k đều thỏa mãn $q_k \ge q$.*

Chứng minh

Do $q_k \in Prime[A, B]$ nên chỉ cần chứng minh được $A \ge q$ thì hệ quả là hiển nhiên.

Do $f_{k-1} = f_{k-2}.q_{k-1}$ nên

$M_{k-1} \le len(f_{k-2}) + len(q_{k-1}) = M_{k-2} + N_{k-1}.$ (11)

Theo cách xác định N_{k-1} thì

$N_{k-1} \le L - M_{(k-1)-1} - \big(k - (k - 1)\big)N - 1$

$= L - M_{k-2} - N - 1.$

Hay $M_{k-2} + N_{k-1} \le L - N - 1.$ (12)

Từ (11) và (12) thu được

$A = \left\lceil\frac{2^{L-1}}{f_{k-1}}\right\rceil \ge \frac{2^{L-1}}{f_{k-1}} \ge 2^{L-1-M_{k-1}} \ge 2^{L-1-(L-N-1)} = 2^N > q.$

Đây là điều cần chứng minh.■

Hệ quả 7. *Với A và B xác định tại bước 7 của Thuật toán 3 thì:*

a) $B < 2^{L-N}$ (13)

b) $B - A + 1 \ge 2^N$ (14)

Chứng minh

Từ $B = \left\lfloor\frac{2^{L-1}}{f_{k-1}}\right\rfloor$ nên B này lớn nhất khi $k = 1$, khi đó $f_{k-1} = f_0 = 2q$ là số (N+1)-bit nên $f_{k-1} \ge 2^N$. Cho nên:

$B = \left\lfloor \frac{2^L-1}{f_{k-1}} \right\rfloor \le \frac{2^L-1}{f_{k-1}} < \frac{2^L}{2^N} = 2^{L-N}.$

Vậy (13) đã được chứng minh.

Theo (12) thì $len(f_{k-1}) = M_{k-1} \le M_{k-2} + N_{k-1} \le L - N - 1$, hay $f_{k-1} < 2^{L-N-1}$. Nên

$$\begin{aligned} B - A + 1 = & \left\lfloor \frac{2^L-1}{f_{k-1}} \right\rfloor - \left\lceil \frac{2^{L-1}}{f_{k-1}} \right\rceil + 1 \\ \ge & \left(\frac{2^L-1}{f_{k-1}} - 1\right) - \left(\frac{2^{L-1}}{f_{k-1}} + 1\right) + 1 \\ = & \frac{2^{L-1}}{f_{k-1}} - \frac{1}{f_{k-1}} - 1 > \frac{2^{L-1}}{2^{L-N-1}} - 2 \\ & = 2^N - 2. \end{aligned}$$

Như vậy, Hệ quả 7 đã được chứng minh.■

Bổ đề 8. *Với mọi số tự nhiên N đủ lớn hơn thì:*

a) Khoảng cách trung bình giữa 2 số nguyên tố trên tập các số nguyên N-bit, ký hiệu $\rho(N)$, được đánh giá theo biểu thức sau:

$$\rho(N) < N. \qquad (15)$$

b) Khoảng cách trung bình giữa 2 số nguyên tố trên tập các số nguyên N-bit cùng dạng $t.f + 1$ với f là số tự nhiên lớn hơn 1, ký hiệu $\rho_f(N)$, được đánh giá theo biểu thức sau

$$\rho_f(N) \le \frac{\varphi(f)}{f} . \frac{N(2^{N-1}+f-1)}{2^{N-1}}.$$

c) *Hơn nữa nếu f là chẵn và nhỏ hơn 2^{N-1} thì*

$$\rho_f(N) \le N. \qquad (16)$$

Chứng minh

Số các số nguyên tố N-bit, theo định lý Gauss là:

$$\begin{aligned} \pi(2^N) - \pi(2^{N-1}) &\approx \frac{2^N}{Nln2} - \frac{2^{N-1}}{(N-1)ln2} \\ &= \frac{2^{N-1}(N-2)}{N(N-1)ln2}. \end{aligned}$$

Còn số các số nguyên N-bit là 2^{N-1} nên:

$$\rho(N) = \frac{2^{N-1}}{\pi(2^N)-\pi(2^{N-1})} = \frac{N(N-1)ln2}{(N-2)}.$$

Do $\lim\limits_{N\to\infty} \frac{(N-1)ln2}{(N-2)} = ln2 < 1$ nên với N đủ lớn, ta có vế phải trên $< N$ hay (15) đã được chứng minh.

Trước hết, số các số nguyên N-bit có dạng $t.f + 1$, ký hiệu là $D_f(N)$ đúng bằng số các số nguyên t thỏa mãn $\left\lceil \frac{2^{N-1}}{f} \right\rceil - 1 \le t \le \left\lfloor \frac{2^N-1}{f} \right\rfloor - 1$ nên có ước lượng như sau:

$$\begin{aligned} D_f(N) &= \left(\left\lfloor \frac{2^N-1}{f} \right\rfloor - 1\right) - \left(\left\lceil \frac{2^{N-1}}{f} \right\rceil - 1\right) + 1 \\ &\le \frac{2^N-1}{f} - \frac{2^{N-1}}{f} + 1 = \frac{2^{N-1}+f-1}{f}. \end{aligned}$$

Theo định lý Drichlet thì số các số nguyên tố N-bit có dạng $t.f + 1$ là:

$$\pi_f(2^N) - \pi_f(2^{N-1}) \approx \frac{1}{\varphi(f)}\left(\frac{2^N}{Nln2} - \frac{2^{N-1}}{(N-1)ln2}\right) > \frac{1}{\varphi(f)} . \frac{2^{N-1}}{N}.$$

Do đó, $\rho_f(N) = \frac{D_f(N)}{\pi_f(2^N)-\pi_f(2^{N-1})}$

$$\begin{aligned} &\le \left(\frac{2^{N-1}+f-1}{f}\right) : \left(\frac{1}{\varphi(f)} . \frac{2^{N-1}}{N}\right) \\ &= \frac{\varphi(f)}{f} . \frac{N(2^{N-1}+f-1)}{2^{N-1}}. \end{aligned}$$

Như vậy đã chứng minh xong bất đẳng thức trong phần b) của Bổ đề 8.

Từ giả thiết f là số chẵn nên $\frac{\varphi(f)}{f} < \frac{1}{2}$, còn từ $f < 2^{N-1}$ nên $\frac{(2^{N-1}+f-1)}{2^{N-1}} \le 2$, do đó khi thay vào vế phải của bất đẳng thức trên có ngay bất đẳng thức (16). Bổ đề 8 đã được chứng minh.■

B. Các đánh giá về Thuật toán 3

Đánh giá 1. *Thuật toán 3 là đúng đắn, hơn thế nữa nếu $\frac{2^N}{(L-N)L} > 1$ thì thuật toán luôn sinh được bộ (p, q) an toàn theo Định nghĩa 1.*

Chứng minh

Biết rằng với mọi số nguyên dương a thì $Prime(a, 2a) \ne \emptyset$ nên với mọi N ta có $Prime(2^{N-1}, 2^N - 1) = Prime(2^{N-1}, 2^N) \ne \emptyset$, ngoài ra do q là số nguyên tố N-bit nên $Prime(q, 2^N - 1) \ne \emptyset$ nên các bước 1, 5.2 và 5.3 luôn thực hiện được và các số nguyên tố q_i tìm được trong các bước này đều thỏa mãn $q_i \ge q$.

Từ giả thiết $L \ge 2(N + 1)$ và từ $M_0 = len(2q) = N + 1$ nên:

$\left\lfloor \frac{L-M_0-1}{N} \right\rfloor \ge \left\lfloor \frac{2(N+1)-(N+1)-1}{N} \right\rfloor = 1$ hay $\left[1, \left\lfloor \frac{L-M_0-1}{N} \right\rfloor\right] \ne \emptyset$.

Vậy bước 3 là thực hiện được.

Hệ quả 5 chính là điều kiện để bước 5.1 thực hiện được.

Muốn chứng tỏ bước 7 cũng thực hiện được ta cần chỉ ra $Prime(A,B) \neq \emptyset$.

Quả thật, do $\frac{B}{A} = \left\lfloor \frac{2^{L}-1}{f_{k-1}} \right\rfloor : \left\lceil \frac{2^{L-1}}{f_{k-1}} \right\rceil \leq \frac{2^{L}-1}{f_{k-1}} : \frac{2^{L-1}}{f_{k-1}} < 2$ nên $[A,B]$ sẽ chỉ gồm những số nguyên cùng T-bit hoặc gồm hai loại số nguyên T-bit và (T–1)-bit (ở đây $T = len(B)$).

Trong trường hợp thứ nhất, theo phần a) của Bổ đề 8 thì:

$$\#\, Prime(A,B) \geq \frac{B-A+1}{T}.$$

Còn trong trường hợp thứ hai, cũng theo Bổ đề 8 thì:

$$\#\, Prime(A,B)$$
$$= \#\, Prime(A, 2^{T-1}-1) + \#\, Prime(2^{T-1}, B)$$
$$\geq \frac{2^{T-1}-A}{T-1} + \frac{B-2^{T-1}+1}{T} \geq \frac{B-A+1}{T}.$$

Mặt khác, giá trị B ứng với trường hợp $k = 1$ là lớn nhất, khi này $f = 2q$, nên $B < 2^{L-N}$.

Bất đẳng thức trên có nghĩa $T = len(B) < L - N$ và kết hợp với bất đẳng thức (6) trong Hệ quả 7, thu được bất đẳng thức sau:

$$\#\, Prime(A,B) > \frac{2^{N}}{L-N}.$$

Cuối cùng, xét đến bước 9 của thuật toán. Trong bước này các số được duyệt là có dạng $p \leftarrow f_{k-1} * q_k + 1$ với $q_k \in Prime(A,B)$. Chúng gồm không dưới $\frac{2^{N}}{(L-N)}$ các số L-bit và do f_{k-1} là số chẵn và nhỏ hơn 2^{L}, nên theo phần c) của Bổ đề 8 số các số nguyên tố trong số này là:

$$\#\frac{Prime(A,B)}{\rho_{f_{k-1}}(L)} \geq \frac{2^{N}}{(L-N)L}.$$

Từ bất đẳng thức trên, ta thấy rằng nếu vế phải của bất đẳng thức lớn hơn 1 thì thuật toán sẽ tìm được cặp (p,q). Tính an toàn của (p,q) được cho bởi Hệ quả 6.

Kết quả tiếp theo sẽ trình bày đánh giá về chi phí tính toán của Thuật toán 3. Do trong thuật toán sử dụng đến việc kiểm tra tính nguyên tố của một số nguyên mà việc làm này có chi phí phụ thuộc vào phương pháp kiểm tra (chẳng hạn chi phí theo phương pháp như Miller và Rabin là $O(N^3)$ còn theo AKS là $O(N^6)$) nên tác giả dùng ký hiệu $T_{Test}(N)$ để đại diện cho chi phí kiểm tra tính nguyên tố của một số nguyên N-bit.

Đánh giá 2. *Chi phí để sinh được 1 cặp (p,q) an toàn với $p-1$ có $k+2$ ước nguyên tố kể cả bội, ký hiệu là $T_{Gen}(k)$, được cho bởi công thức sau.*

$$T_{Gen}(k) \leq \frac{k.N}{2}.T_{Test}(N) + L(T_{Test}(L) + M.T_{Test}(M)). \quad (17)$$

Với $M = L - kN + k - 1$.

Chứng minh

Từ giả thiết $p-1$ có $k+2$ ước nguyên tố kể cả bội, bỏ qua hai ước nguyên tố là 2 và q thì còn thêm đúng k ước nguyên tố nữa (đương nhiên theo thuật toán 3 thì $1 \leq k \leq \left\lfloor \frac{L-M_0-1}{N} \right\rfloor$). Ngoài ra $T_{Test}(N)$ luôn có bậc cao nhất trong các phép tính số học dùng trong thuật toán nên ta chỉ quan tâm đến chi phí trên với thực tế nó là một đại lượng đơn điệu tăng theo N.

Theo phần a) của Bổ đề 8 thì chi phí trung bình cho việc sinh một số nguyên tố N-bit (thực hiện $q \leftarrow Random\big(Prime(2^{N-1}, 2^{N}-1)\big)$ theo Thuật toán 1) sẽ là $\frac{N}{2}.T_{Test}(N)$.

Trong Thuật toán 3, cần đến việc sinh số nguyên tố N-bit q ở bước 1 và $k-1$ số nguyên tố N_i-bit q_i $(i = 1,..,k-1)$ ở bước 5. Như vậy cho đến hết bước 5 cần một chi phí tính toán là:

$$T_1 = \frac{1}{2}\big(N.T_{Test}(N) + \textstyle\sum_{i=1}^{k-1} N_i.T_{Test}(N_i)\big).$$

Bước 7 và bước 8 thực hiện sinh 1 số nguyên tố q_k và kiểm tra tính nguyên tố của số L-bit p. Tương tự mỗi lần lặp ở bước 9, thì 9.1 thực hiện tìm số nguyên tố tiếp theo (thực hiện $q_k \leftarrow Next_{Prime[A,B]}(q_k)$ theo Thuật toán 2) và kiểm tra tính nguyên tố của số L-bit p (điều kiện dừng cho vòng $while$). Do các giá trị $p = f_{k-1} * q_k + 1$ được kiểm tra trong bước 9 không phải là các số liền nhau trong tập các số dạng $p = f_{k-1} * t + 1$, nên số lần lặp trung bình của bước này sẽ là $\rho_{f_{k-1}}(L) < L$.

Như vậy, chi phí cho các bước 7, 8 và 9 trung bình là:

$$T_2 = L(T_{Test}(L) + N_k.T_{Test}(N_k)) \quad (18)$$

với $N_k = len(q_k)$.

Do $k < L$ nên $T_{Gen}(k) = T_1 + T_2$ sẽ lớn nhất khi N_k lớn nhất, từ quan hệ:

$$2^{L-1} < f_{k-1} * q_k + 1 \le 2^L - 1$$

nên điều trên xảy ra khi f_{k-1} bé nhất tức là $f_{k-1} = 2.q^k$.

Khi đó:

$T_1 = \frac{k.N}{2}.T_{Test}(N)$.

Do $q > 2^{N-1}$ nên:

$f_{k-1} > 2^{k(N-1)+1} \Rightarrow q_k < 2^{L-(k(N-1)+1)} = 2^M$

$\Rightarrow N_k < M.$ (19)

Thay (19) vào (18) ta được:

$T_{Gen}(k) = T_1 + T_2 < \frac{k.N}{2}.T_{Test}(N) + L(T_{Test}(L) + M.T_{Test}(M))$.

Và đây là điều cần chứng minh.■

Đánh giá 3. *Thuật toán 3 không thể sinh toàn bộ các cặp (p, q) an toàn.*

Chứng minh

Xét Thuật toán 3 với cặp đầu vào $(L, N) = (8, 2)$. Với bước 3 của Thuật toán 3 thì số các ước nguyên tố lẻ của $p - 1$, không kể q, tối đa là:

$$k = \left\lfloor \frac{L - M_0 - 1}{N} \right\rfloor = \left\lfloor \frac{8-3-1}{2} \right\rfloor = 2.$$

Như vậy Thuật toán 3 này không thể sinh được bộ $(p, q) = (163, 3)$ do $163 = 2.3^4 + 1$ tương ứng với $k = 3$.

V. Một số kết quả của cài đặt thuật toán

Sử dụng bộ công cụ lập trình Visual Studio 2013 kết hợp với thư viện mật mã Miracl để cài đặt Thuật toán 3, kết quả thu được các bộ tham số dùng cho các giao thức thỏa thuận khóa DH-KE an toàn, chống lại được các tấn công như đã chỉ ra ở trên.

Ví dụ về bộ tham số được sinh bởi chương trình:

Bộ tham số:

Số nguyên tố p (2048 bit):

```
8BF76C050D3DFB10C9FB37D722F986388
DE867CA00499826C96562867844430F74
BB0E04E141BED83E4930ECDCB268C8852
6E6A1F2E37D43543D60F9775A0F83F17D
523A8DF64A3A12CBC667E78F9BD0F4019
599D1E186AE9AAF6E56BD93CA053DE0E7
D6066A466D0E60DC96991B9006DC18EA1
B9C70726EDF99028DAD6E632B9A1B6E4B
070BA1C975250B986E6993EB58853A2AE
CC2B9F2CBE903338752ED080222192C08
1D757AED91E30DC6C5AC904AF07BFD723
87335331C279701849D2F652DD0A9EB35
ACD8C3F644B71D20635D34940FF7F9AC8
0E7A72AAF60A11D53FA8B08D4A8336749
CE9A723C5545461E11FDA4B7A9556B768
07F81948652F677A7
```

Số nguyên tố q (224 bit):

```
8BF59EC7A4FA0349F4D76BF6D26BBE668
6D07B8B2DAFB3283D397337
```

Các số nguyên tố p_i:

p_0 (692 bit):

```
DAC8F4EDD3FC57287873DC63AB8B6AD83
7722E0D211194CA80CEA267C24233FAA8
94E90DD62C89DCFA81663B83477468E8E
8280758A1507E36DF0876C07F498BD6B0
A54DF7E57B7B013DBC651AD019B1494E3
4A213581
```

p_1 (1132 bit):

```
95C7B7C6166A3AB9CC497D086A82E87CF
32E2153FF491771BE334F45EE678C7B6F
E062DD86C07232E6B0CF29A53A1ACFC83
C870AD062335ECEB18D2350E6DE107A67
265B6E02923DFA621B19CAF96444D61D1
0EE946A6806342D6E97733683A108C4B0
F97B62828145C8AF53FA0AB44DA0F3EA3
DDCEA50B2229CBE583EB89570CDB2C8E2
1E3E0892C9A056616C5
```

VI. Kết luận

Đối với việc triển khai các giao thức trao đổi khóa kiểu DH-KE trong các ứng dụng bảo mật thông tin, thì việc sinh được các bộ tham số an toàn là rất quan trọng, nó đảm bảo cho giao thức DH-KE chống lại được một số tấn công để tìm ra khóa bí mật của người dùng (cụ thể là các tấn công liên quan đến nhóm con nhỏ). Bài báo đã đề xuất, xây dựng được một thuật toán sinh số nguyên tố an toàn và các đánh giá, phân tích về tính đúng đắn, độ phức tạp của thuật toán về mặt toán học. Đây là ưu điểm chính của thuật toán đề xuất so với thuật toán sinh số nguyên tố an toàn của Lim-Lee, vì thuật toán đó chỉ chú trọng vào việc sinh thực nghiệm cho những số nguyên tố dạng này nhưng không có đảm bảo chắc chắn về mặt toán học cho việc sinh chúng. Bài báo cũng đưa ra kết quả cài đặt thực tế của thuật toán để minh chứng cho tính khả thi của thuật toán đưa ra.

TÀI LIỆU THAM KHẢO

[1] S. C. Pohlig and M. E. Hellman (1978), An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Trans. Inform. Theory, IT-24 (1), pp.106-110.

[2] C. Lim and P. Lee (1997), A Key Recovery Attack on Discrete Log-based Schemes Using a Prime Order Subgroup, EUROCRYPT 1997.

[3] J.M.Pollard (1978), Monte Carlo methods for index computation (rood p), Math. Comp., 32(143), pp.918-924.

[4] FIPS PUB 186-3 (2009), Digital Signature Standard (DSS), https://csrc.nist.gov/csrc/media/publications/fips/186/3/archive/2009-06-25/documents/fips_186-3.pdf, Accessed on 10/9/2020.

[5] T. Matsumoto, Y. Takashima and H. Imai (1986), On seeking smart public-key distribution systems, The Transactions of the [EICE of Japan, E69, pp.99-106.

[6] FSF, Gnu privacy guard, http://www.gnupg.org/, Accessed on 10/9/2020.

[7] Gutmann. P, cryptlib, https://www.cs.auckland.ac.nz/~pgut001/cryptlib/, Accessed on 10/9/2020.

[8] PGP. I, OpenPGP, https://www.openpgp.org/, Accessed on 10/9/2020.

[9] MIRACL, MIRACL Cryptographic SDK, https://github.com/miracl/MIRACL, Accessed on 10/9/2020.

[10] Rechard Crandall, Carl Pomerance (2005), Prime Numbers: A Computational Perspetive, Springer, https://www.springer.com/gp/book/9780387252827, Accessed on 10/9/2020.

[11] Nguyễn Quốc Toàn, Đỗ Đại Chí, Triệu Quang Phong (2016), Về một tiêu chuẩn tham số cho bài toán logarithm rời rạc, Nghiên cứu Khoa học và Công nghệ trong lĩnh vực An toàn thông tin, ISSN 2615-9570. No 02. Vol 01. 2016.

SƠ LƯỢC VỀ TÁC GIẢ

ThS. Nguyễn Thanh Sơn

Đơn vị công tác: Cục Quản lý mật mã dân sự và Kiểm định sản phẩm mật mã, Ban Cơ yếu Chính phủ.

Email: vuphongthanhson@gmail.com

Quá trình đào tạo: Tốt nghiệp Kỹ sư Kỹ thuật mật mã (1993-1998); Thạc sĩ Kỹ thuật mật mã (2002-2005) tại Học viện Kỹ thuật mật mã.

Hướng nghiên cứu hiện nay: An toàn bảo mật thông tin, Kỹ thuật mật mã.