Thuật toán sinh số nguyên tố tất định hiệu quả trên thiết bị nhúng

Trần Duy Lai, Hoàng Văn Thức, Trần Sỹ Nam

***Tóm tắt*— Trong bài báo này, chúng tôi giới thiệu thuật toán sinh số nguyên tố tất định dùng trong mật mã có thể cài đặt hiệu quả trên các thiết bị nhúng. Đóng góp chính của chúng tôi là làm tường minh về đảm bảo cơ sở lý thuyết và cài đặt thực tế thuật toán nêu trên.**

***Abstract*— In this paper, we introduce a provable prime generation algorithm, which is used in cryptography and can be implemented efficiently on embedded devices. Our main contribution is to make sure that the theoretical background is clear, correct and implement that algorithm.**

Từ khóa*— *số nguyên tố tất định; số nguyên tố chứng minh được; thuật toán CubeRoot.

I. GIỚI THIỆU

Các thuật toán sinh hiệu quả các số nguyên tố lớn đóng một vai trò quan trọng trong quá trình sinh tham số khóa cho các lược đồ, nguyên thủy mật mã hiện đại như: RSA, DSA, trao đổi khóa Diffie-Hellman.… Các thuật toán sinh số nguyên tố có thể được chia làm hai lớp: (1) thuật toán sinh số nguyên tố xác suất và (2) thuật toán sinh số nguyên tố tất định (còn gọi là phương pháp sinh số nguyên tố chứng minh được).

Đối với các số nguyên tố dùng trong lĩnh vực mật mã, yêu cầu đầu tiên là chúng phải được sinh bởi các thuật toán sinh số nguyên tố tất định. Yêu cầu này nhằm đảm bảo tính đúng đắn và bền vững của các nguyên thủy mật mã. Bên cạnh đó, việc sử dụng các thuật toán sinh số nguyên tố tất định sẽ giúp chúng ta dễ dàng điều khiển các tính chất an toàn mật mã khác liên quan tới các số nguyên tố.

Trong [1] và [4] đã đưa ra hai thuật toán sinh số nguyên tố tất định là thuật toán sinh số nguyên tố của Shawe-Taylor và thuật toán sinh số nguyên tố của Maurer. Cơ sở cho cả hai thuật toán trên đều dựa vào định lý Pocklington, được trình bày dưới đây:

Định lý 1 (Định lý Pocklington [2]). *Giả sử n =RF +1 với phân tích của F ra thừa số nguyên tố là:*

$$F = q_1 q_2 ... q_r, \ (q_1 \le q_2 \le ... \le q_r)$$

Nếu tồn tại số nguyên a sao cho

$a^{n-1} \equiv 1 \pmod{n}$ *và* $\left(a^{(n-1)/q_j}, n\right) = 1$ *với j = 1,... r*

thì mọi ước nguyên tố p của n sẽ đồng dư với 1 (mod F).

Tuy nhiên, cả hai thuật toán trên đều sử dụng phương pháp đệ quy (đối với thuật toán của Maurer thực hiện đệ quy ở bước thứ 3, còn thuật toán của Shawe - Taylor gọi đệ quy ở bước thứ 2), chi tiết xem trong [1]. Việc sử dụng phương pháp đệ quy dẫn đến khó khăn trong kiểm soát tài nguyên của hệ thống khi thực thi thuật toán. Do vậy, việc cài đặt thuật toán trên các thiết bị có tài nguyên hạn chế như PKI Token, SmartCard… có tính khả thi thấp.

Trong bài báo này, chúng tôi giới thiệu một thuật toán sinh số nguyên tố tất định có khả năng khắc phục được hạn chế trên, hướng tới ứng dụng trên các thiết bị có tài nguyên hạn chế mà vẫn đảm bảo yêu cầu an toàn.

Bố cục của bài báo bao gồm bốn mục, sau Mục mở đầu, Mục II trình bày thuật toán sinh số nguyên tố tất định, Mục III trình bày kết quả cài đặt thuật toán trên thiết bị nhúng và Mục cuối là kết luận.

II. THUẬT TOÁN SINH SỐ NGUYÊN TỐ TẤT ĐỊNH

A. Cơ sở lý thuyết

Định lý căn bậc 3 (Cube Root Theorem) được đề cập tới trong [3, Định lý 4]. Định lý này là cơ sở lý thuyết cho thuật toán sinh số nguyên tố sẽ được trình bày trong phần sau. Chính vì vậy, chúng ta cần tìm hiểu chứng minh của định lý này một cách tường minh.

Định lý 2 (Định lý Cube Root Theorem). *Giả sử p là một số nguyên tố lẻ, n = 2rp + 1 cùng với r là một số nguyên sao cho r < p^2 + 1. Nếu tồn tại một số nguyên a cùng với 2 ≤ a ≤ n, sao cho:*

(i) $a^{n-1} \equiv 1 \pmod{n}$ *và* $\gcd(a^{2r} - 1, n) = 1$

(ii) $r = up + s$, $1 \le s < p$ *đối với u lẻ*

thì n là số nguyên tố.

Trong [3] nói rằng, Định lý 2 như là một hệ quả của Định lý 3 dưới đây.

Định lý 3 (Brillhart-Lehmer-Selfridge-Tuckerman-Wagstaff [5]). *Giả sử n > 3 là một số nguyên lẻ, giả sử n = rF + 1 trong đó F là*

phân tích được hoàn toàn và gcd(r, F) = 1. Giả sử tồn tại một số nguyên a sao cho:

(i) $a^{n-1} \equiv 1 \pmod n$

(ii) $\gcd(a^{(n-1)/q} - 1, n) = 1$ *đối với mỗi thừa số nguyên tố q của F*

Giả sử $r = uF + s$, $1 \le s < F$, *và giả sử* $n < 2F^3 + 2F$, $F > 2$. *Nếu u lẻ hoặc nếu u là chẵn và* $s^2 - 4u$ *không là số chính phương, thì n là số nguyên tố.*

Nếu trong Định lý 3 ta lấy $F = p$ thì có $n = 2rp + 1 \le 2p^2 \cdot p + 1 < 2p^3 + 2p$. Nhưng khi đó $2r = 2up + 2s$ và $2u$ lại là số chẵn, nên việc áp dụng Định lý 3 như thế nào chưa xác định được.

Bên cạnh đó, trong khi tìm hiểu về kết quả được đưa ra trong Định lý 3, chúng tôi nhận thấy, trong tài liệu [6] có đề cập đến thuật ngữ "*Cube Root Theorem*" và trích dẫn đến [5, Định lý 11] hay chính là Định lý 3 đã được phát biểu ở trên. Chúng tôi cũng đã tìm hiểu tài liệu [7] của ba tác giả Brillhart, Lehmer và Selfridge, trong tài liệu này có phát biểu Định lý 4 dưới đây gần tương tự như Định lý 3 nêu trên.

Định lý 4. *Giả sử* $n - 1 = F_1R_1$, *trong đó* F_1 *là phần chẵn đã phân tích được của* $n - 1$, $R_1 > 1$ *và* $(F_1, R_1) = 1$. *Giả sử rằng, đối với mỗi số nguyên tố* p_i *là ước của* F_1 *tồn tại số* a_i *sao cho n là "giả nguyên tố" cơ sở* a_i *(tức là* $a_i^{n-1} \equiv 1 (\mathrm{mod}\, n), 1 < a_i < n-1$*) và* $\left(a_i^{(n-1)/p_i} - 1, n\right) = 1$). *Giả sử* $m \ge 1$. *Khi* $m > 1$, *giả sử tiếp theo rằng* $\lambda F_1 + 1 \nmid N$ *đối với* $1 \le \lambda < m$. *Nếu*

$$n < (mF_1 + 1)[2F_1^2 + (r - m)F_1 + 1],$$

trong đó r và s được định nghĩa bởi $R_1 = (n - 1)/F_1 = 2F_1s + r$, $1 \le r < 2F_1$, *thì n là nguyên tố khi và chỉ khi* $s = 0$ *hoặc* $r^2 - 8s \ne t^2$ ($r \ne 0$ *và* R_1 *là lẻ*).

Trong [7] có đưa ra chứng minh chi tiết cho Định lý 4, nhưng việc từ đó suy ra Định lý 3 hay Định lý 2 là không dễ thấy.

Vì vậy, nội dung còn lại của phần này, chúng tôi chứng minh trực tiếp Định lý 2.

Chứng minh Định lý 2.

Giả sử n không là số nguyên tố. Theo (i), theo Định lý 1, mọi ước số của n đều có dạng $mp+1$.

Ta thấy rằng, n không thể có quá 3 ước, vì mỗi ước đều có dạng $mp+1$ và $n \le 2p^3 + 1$.

Xét trường hợp n có 3 ước, gọi các ước đó là $m_1p + 1$, $m_2p + 1$ và $m_3p + 1$ với $m_1 \le m_2 \le m_3$.

Nếu $m_1 = m_2 = m_3 = 1$ thì ta có $(p + 1)^3 = n = 2rp + 1$. Suy ra: $p^2 + 3p + 3 = 2r$

Điều này là không thể, vì tính chẵn lẻ của hai vế trong biểu thức trên khác nhau.

Đối với các trường hợp còn lại ta có:

$2p^3 + 1 = 2p \cdot p^2 + 1 \ge 2rp + 1 = n =$

$(m_1p + 1)(m_2p + 1)(m_3p + 1) > m_1m_2m_3p^3 + 1$

$\ge 2p^3 + 1$ (điều này không thể xảy ra).

Giả sử rằng n có 2 ước, tức là:

$$2up^2 + 2sp + 1 = 2(up + s)p + 1 = 2rp + 1 = n$$
$$= (m_1p + 1)(m_2p + 1) = m_1m_2p^2 + (m_1 + m_2)p + 1$$

Do $n \le 2p^3 + 1$ nên $m_1m_2 < 2p$. (1)

Ta sẽ chứng minh rằng cũng có $m_1 + m_2 < 2p$. (2)

Thật vậy, giả sử ngược lại có $m_1 + m_2 \ge 2p$.

Khi đó: $2p > m_1m_2 \ge m_1 + m_2 - 1 \ge 2p - 1$ (3)

Hệ (3) chỉ xảy ra khi $m_1m_2 = m_1 + m_2 - 1 = 2p - 1$, tức là khi $m_1 = 1$ và $m_2 = 2p - 1$.

Khi đó, ta có:

$$2up^2 + 2sp + 1 = n = m_1m_2p^2 + (m_1 + m_2)p + 1 =$$
$$(2p\text{-}1)p^2 + 2p \cdot p + 1$$
$$2up + 2s = (2p - 1)p + 2s$$

tính chẵn lẻ của hai vế khác nhau, nên có (2).

Do $2up + 2s = m_1m_2p + (m_1 + m_2)$ (4), nên trường hợp cả m_1 và m_2 đều là số lẻ không thể xảy ra, vì như thế biểu thức trên có tính chẵn lẻ của 2 vế khác nhau. Tức là $m_1m_2 = 2k$.

Lúc đó (4) có thể được viết lại như sau:

$$u \cdot 2p + 2s = k \cdot 2p + (m_1 + m_2).$$

Do $2 \le 2s < 2p$ và (2) nên:

$$k = u \text{ và } (m_1 + m_2) = 2s$$

hay

$$m_1m_2 = 2k = 2u \text{ và } m_1 + m_2 = 2s.$$

Tức là tam thức bậc hai:

$$X^2 - (m_1 + m_2)X + m_1m_2 = X^2 - 2sX + 2u = 0$$

có 2 nghiệm là m_1 và m_2. Điều đó chỉ có thể xảy ra, khi:

$$\Delta' = s^2 - 2u = t^2$$

Nhưng $s^2 - t^2$ hoặc là số lẻ hoặc chia hết cho 4, trong khi đó $2u$ chỉ chia hết cho 2. Điều đó dẫn đến mâu thuẫn. □

B. Thuật toán sinh số nguyên tố tất định

Định lý 2 là cơ sở cho thuật toán sinh số nguyên tố tất định. Theo định lý này, ta sẽ nhân ba độ dài các số nguyên tố chứng minh được tại mỗi

lần lặp (thay cho việc nhân hai độ dài như trong phương pháp của Shawe-Taylor và Maurer).

Thuật toán CubeRoot() (Thuật toán 3.2, [3])

Đầu vào

Kích thước bit của số cần sinh ℓ_n và tích các số nguyên tố bé $\Pi = 2 \cdot 3 \cdot 5 \cdot \cdot p_t$

Đầu ra

n là số nguyên tố chứng minh được ℓ_n-bit

Thuật toán

1. $\ell \leftarrow \ell_n$
2. **while** $\ell > 31$ **do**
3. $\ell \leftarrow \lfloor \ell/3 \rfloor$
4. $\ell \leftarrow \ell + 1$
5. $n \leftarrow$ GenInitPrime(ℓ)
6. **while** $\ell < \ell_n$ **do**
7. $p \leftarrow n$
8. $\ell \leftarrow min(3\ell - 1, \ell_n)$
9. $I \leftarrow \left\lfloor \frac{2^{\ell-1}}{2p} \right\rfloor$
10. Chọn r ngẫu nhiên từ $[I + 1, 2I]$ sao cho $r = up + s$, $1 \leq s < p$ đối với u lẻ và $n \leftarrow 2rp + 1$ là nguyên tố cùng nhau với Π và chuyển tới 12
11. Cập nhật r trong $[I + 1, 2I]$ sao cho $r = up + s$, $1 \leq s < p$ đối với u lẻ và $n \leftarrow 2rp + 1$ là nguyên tố cùng nhau với Π
12. Nếu $\ell < 129$ thì
13. chọn một số nguyên a ngẫu nhiên từ $[2, n - 2]$
14. ngược lại
15. $a \leftarrow 2$
16. Nếu $a^{n-1} \bmod n \neq 1$ thì chuyển tới 11
17. Nếu $\gcd(a^{2r} - 1, n) \neq 1$ thì chuyển tới 11
18. Trả về n

Các chú ý:

- *Hàm GenInitPrime() ở bước 5*. Cách tiếp cận ở đây là áp dụng phép kiểm tra Miller-Rabin để sinh các số nguyên tố khởi đầu nhỏ hơn hoặc bằng 2^{32}. Kết quả nghiên cứu của Pomerance và các cộng sự trong [8], Jaeschke đã chứng minh trong [9] rằng, số nguyên bất kỳ nhỏ hơn 2^{32} được coi là số nguyên tố nếu nó vượt qua phép kiểm tra Miller-Rabin với 3 cơ số là 2, 7 và 61.
- *Lựa chọn và cập nhật của r và n.* Giải pháp để tìm được một r thích hợp tại bước 10 của Thuật toán bao gồm việc chọn ngẫu nhiên một giá trị đầu tiên $r \in [I + 1, 2I]$, sao cho $r = up+s$ với u lẻ và $1 \leq s < p$; đặt $n = 2rp + 1$ và sau đó tăng r lên 1 và n lên $2p$ cho đến khi các thặng dư môđun $(w_i = n \bmod p_i)_{i = 1, \ldots, t}$ tất cả đều khác 0. Mỗi w_i sau đó được tăng lên bởi $2p \bmod p_i$. Một cách thức nhằm tăng hiệu quả nằm ở việc tính các giá trị $2p \bmod p_i$ bằng cách gấp đôi môđun p_i các thặng dư w_i của lần lặp trước, vì giá trị trước đó của n tương ứng với giá trị mới của p ở lần lặp hiện tại. Tại Bước 11, cùng một cách cập nhật tăng dần của r và n được áp dụng để sinh ra dự tuyển tiếp theo nguyên tố cùng nhau với Π.
- *Sử dụng hằng số a=2.* Theo (ii) của [3, Định lý 2], chúng ta biết rằng xác suất mà một giá trị ngẫu nhiên a bác bỏ số nguyên tố n tại Bước 16 hoặc 17 là $1/p$. Giả sử, tỷ lệ của các số nguyên tố bị bác bỏ không thay đổi nhiều từ một giá trị của a tới giá trị khác, việc chọn một giá trị hằng số a có ảnh hưởng không đáng kể lên phân bố của các số nguyên tố được sinh ra khi kích thước bit ℓ là đủ lớn. Ví dụ, khi sinh một số nguyên tố 128-bit n = 2rp + 1 từ số nguyên tố chứng minh được p có 65-bit, ít hơn $1/2^{64}$ các số nguyên tố sẽ bị bác bỏ. Chúng ta chấp nhận mất mát entropy không đáng kể này và sử dụng $a = 2$ cho kiểm tra Fermat khi $\ell > 128$. Việc này dẫn tới các phép tính lũy thừa nhanh hơn cho các bước 16 và 17.

III. KẾT QUẢ CÀI ĐẶT

Chúng tôi đã thực hiện cài đặt thuật toán CubeRoot bằng ngôn ngữ C, sử dụng bộ tính toán số lớn BigNumber trên thiết bị nhúng ARMv7 Processor rev 2 (v7l). Kết quả chạy chương trình sinh các số nguyên tố có độ dài và thời gian trung bình được thống kê trong Bảng 1.

BẢNG 1. KẾT QUẢ CHẠY CHƯƠNG TRÌNH SINH CÁC SỐ NGUYÊN TỐ

Độ dài	512	768	1024	1536	1792	2048
Thời gian (ms)	550	1050	2560	12100	18200	36400

IV. KẾT LUẬN

Bài báo này đã giới thiệu thuật toán sinh số nguyên tố tất định có thể cài đặt hiệu quả trên thiết bị nhúng nhằm làm tường minh về cơ sở lý thuyết cho thuật toán, đồng thời cài đặt thực tế thuật toán trên một thiết bị nhúng cụ thể. Từ kết quả cài đặt thực tế cho thấy, thuật toán hoàn toàn có thể sử dụng cho mục đích sinh tham số cho các lược đồ, thuật toán như RSA, DSA, DH… trên các thiết bị có tài nguyên hạn chế như PKI Token, SmartCard….

TÀI LIỆU THAM KHẢO

[1]. ISO/IEC 18032: "Information technology - Security techniques – Prime number generation", first edition, 2005-01-15.

[2]. Rechard Crandall, Carl Pomerance, "Prime Numbers", Springer, 2005.

[3]. Christophe Clavier, Benoit Feix, Loïc Thierry and Pascal Paillier, "Generating Provable Primes Efficiently on Embedded Devices", PKC 2012, LNCS 7293, pp. 372–389, 2012.

[4]. FIPS PUB 186-3, "Digital Signature Standard (DSS)", 2009.

[5]. Brillhart, J., Lehmer, D.H., Selfridge, J.L., Tuckerman, B., Wagstaff Jr., S.S., "Factorization of $b^n \pm 1$, b = 2, 3, 5, 7, 10, 11, 12 Up to High Powers", Contemporary Mathematics vol. 22. American Mathematical Society, 1988.

[6]. Richard P. Brent, "Factorization of the tenth Fermat number", Mathematics of Computation, vol. 68, no. 225, January 1999, pp. 429-451.

[7]. John Brillhart, D. H. Lehmer, and J. L. Selfridge, "New Primality Criteria and Factorizations of $2^m \pm 1$", Math. Comp. 29 (1975), pp. 620-647.

[8]. Pomerance C., Selfridge C., Wagstaff, J.L., "The pseudoprimes to 25.10e9", Mathematics of Computation 35, pp. 1003–1026, 1990.

[9]. Jaechke, G., "On strong pseudoprimes to several bases", Mathematics of Computation 61, pp. 915–926, 1993.

SƠ LƯỢC VỀ TÁC GIẢ

TS. Trần Duy Lai

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

Email: tdlai@bcy.gov.vn

Tốt nghiệp chuyên ngành Xác suất - Thống kê, Đại học Matxcơva năm 1985. Nhận bằng Tiến sĩ ngành Toán ứng dụng, Đại học Bách khoa Hà Nội năm 1996.

Hướng nghiên cứu hiện nay: Toán, Khoa học - Công nghệ Mật mã, Bảo mật thông tin trên mạng máy tính.

TS. Hoàng Văn Thức

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

Email: thuchv@yahoo.com

Tốt nghiệp Kỹ sư năm 1998 và Thạc sĩ năm 2004 chuyên ngành Kỹ thuật Mật mã, Học viện Kỹ thuật Mật mã. Nhận bằng Tiến sĩ Toán học, Viện Khoa học - Công nghệ Quân sự năm 2012.

Hướng nghiên cứu hiện nay: Khoa học - Công nghệ Mật mã.

KS. Trần Sỹ Nam

Đơn vị công tác: Viện Khoa học - Công nghệ Mật mã, Ban Cơ yếu Chính phủ, Hà Nội.

E-mail: transynam1989@gmail.com

Tốt nghiệp chuyên ngành An toàn thông tin hệ thống viễn thông, Học viện FSO, Liên bang Nga năm 2013.

Hướng nghiên cứu hiện nay: Công nghệ mạng và bảo mật mạng.