MỘT SỐ MỞ RỘNG CHO DẠNG BIỂU DIỄN NAF CỦA SỐ NGUYÊN DƯƠNG

Phạm Văn Lực[*], **Lều Đức Tân**[+]
[*] Học Viện Công Nghệ Bưu Chính Viễn Thông
[+] Học Viện Kỹ thuật mật mã, Ban Cơ yếu Chính phủ

Tóm tắt: Trong bài báo này, chúng tôi đưa ra một thuật toán cải tiến mới cho việc tìm dạng biểu diễn không liền kề NAF của số nguyên dương k. Tính đúng đắn của thuật toán được phân tích chi tiết cùng với một số đánh giá hiệu quả của thuật toán đề xuất. Cuối cùng, chúng tôi đề xuất một dạng biểu diễn mới nhằm tăng hiệu quả thực thi của một số phép tính số học, như phép tính lũy thừa trên trường hữu hạn hay phép nhân điểm trên đường cong elliptic.

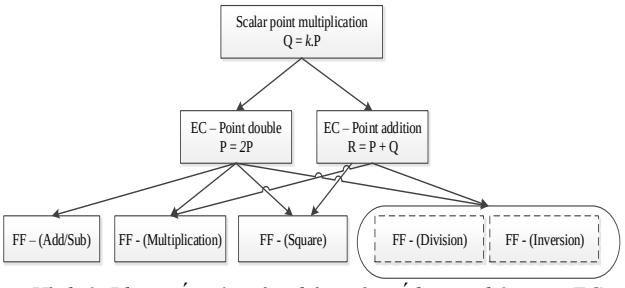
Từ khóa: Dạng biểu diễn không liên kề, phép tính số học, đường cong elliptic, trường hữu hạn.

I. GIỚI THIỆU CHUNG

Các thiết bị IoT, di động (thoại thông minh, máy tính bảng) đang trở nên phổ biến. Mặc dù cấu hình của các thiết bị này tương đối mạnh, nhưng chúng vẫn bị hạn chế ở một số khía cạnh như điện năng tiêu thụ, tốc độ xử lý. Do đặc điểm của những thiết bị này thường là không dây, vấn đề bảo mật dữ liệu đường truyền để ngăn chặn tấn công nghe lén và rò rỉ thông tin cá nhân là rất quan trọng. Vì vậy, việc nghiên cứu triển khai phần mềm mật mã hiệu quả trên các thiết bị này trở nên rất đáng quan tâm. Các giải pháp bảo mật thường dựa trên nền tảng mật mã khóa công khai và mật mã khóa đối xứng; đặc biệt, các lược đồ trên đường cong elliptic khóa công khai thường được sử dụng do có hiệu quả cao như: Thuật toán chữ ký số trên đường cong elliptic (ECDSA) và lược đồ thỏa thuận khóa Diffie Hellman trên đường cong elliptic (ECDH).

Phép nhân vô hướng là phép toán trọng tâm và tốn thời gian nhất trong nhiều hệ mật khóa công khai dựa vào đường cong elliptic như các hệ mật đường cong elliptic ECC, đường cong Hyperelliptic HECC và các hệ mật dựa vào cặp. Cấu trúc thuật toán và tính toán của nó là chủ đề được nghiên cứu nhiều trong những năm gần đây nhằm làm giảm thời gian thực thi cũng như yêu cầu về năng lực tính toán và bộ nhớ của nó. Như vậy, làm cho khả năng cài đặt phù hợp với vô số các ứng dụng mới trong rất nhiều thiết bị như smartcard, cellphone, thẻ RFID và các mạng cảm biến không dây.

An toàn của hệ mật đường cong elliptic (ECC) dựa trên bài toán khó là Bài toán tìm logarit rời rạc trên đường cong elliptic (ECDLP – Elliptic Curve Discrete Logarithm Problem), tức là cho một đường cong E định nghĩa trên trường hữu hạn $\mathbb{F}_q$, một điểm $P \in E(\mathbb{F}_q)$ có cấp n và một điểm $Q \in E(\mathbb{F}_q)$, tìm một số nguyên (nếu tồn tại) $k \in [0, n-1]$ sao cho $Q = kP$. Thuật toán nhân điểm được thực hiện bởi phép cộng và phép nhân đôi, ví dụ $7P = 2((2P) + P) + P$. Phép tính $Q = kP$ được gọi là phép nhân điểm vô hướng và phép tính này tiêu thụ thời gian thực hiện chính trong cài đặt ECC. Phân cấp của các phép toán số học cơ bản trên EC được thể hiện như hình dưới đây.



Hình 1. Phân cấp của các phép toán số học cơ bản trên EC

Phân cấp toán học của phép nhân vô hướng điểm trên đường cong elliptic bao gồm 3 mức:

- Mức 1: Tính toán vô hướng kP;
- Mức 2: Tính toán điểm bao gồm phép cộng điểm $P+Q$, phép nhân đôi điểm $2P$,…;
- Mức 3: Tính toán trên trường hữu hạn bao gồm phép cộng, trừ, bình phương, nhân, nghịch đảo trên trường F_p.

Theo hình trên, các phép toán trên EC được thực hiện từ sự kết hợp của các phép toán cơ bản trên trường hữu hạn (finite field – FF). Phép nhân điểm vô hướng là phép toán trọng tâm và tốn thời gian nhất trong nhiều hệ mật khóa công khai dựa trên đường cong elliptic.

Ký hiệu các chữ hoa in nghiêng *M, S* và *A* tương ứng là chi phí tính toán của phép nhân, phép bình phương, và phép cộng hoặc trừ trên trường. Để đơn giản hóa các phân tích chi phí chúng tôi đưa ra chú ý rằng các cài đặt dựa trên phần mềm thì phép bình phương nhanh hơn phép nhân. Trong các tài liệu thường dùng biểu thức $1S = 0.8M$.

Bảng 1: Chi phí của một số phép toán.

Phép toán	Chi phí
Nhân đôi điểm nhanh [12]	$2M + 8S$
Nhân đôi điểm trước đó	$4M + 6S$
Cộng điểm tổng quát nhanh [12]	$11M + 5S$
Cộng điểm tổng quát trước đó	$12M + 4S$
Cộng điểm hỗn tạp nhanh [12]	$7M + 4S$

Tác giả liên hệ: Phạm Văn Lực,
Email: pvluc@bcy.gov.vn
Đến tòa soạn: 18/9/2020, chỉnh sửa: 21/11/2020, chấp nhận đăng: 24/12/2020.

Cộng điểm hỗn tạp trước đó	$8M + 3S$
Nhân ba điểm nhanh [12]	$6M + 10S$
Nhân ba điểm trước đó [14]	$10M + 6S$
Nhân đôi-cộng hiệu quả [13]	$11M + 7S$
Nhân đôi-cộng trước đó [3]	$12M + 9S$

Trong bài báo này, chúng tôi nghiên cứu về phép nhân vô hướng điểm trên đường cong elliptic ở mức 1. Các thuật toán truyền thống để cài đặt phép toán này thường được dựa vào khai triển nhị phân của các số (ví dụ như NAF, *w*NAF), phương pháp sau dựa vào thực thi thành công của các phép toán điểm cơ bản, tức là nhân đôi và cộng.

Cho đến nay dạng biểu diễn không liền kề, ký hiệu là NAF (non-adjacent form), là dạng đạt được tính chất có giá trị w nhỏ nhất trong các biểu diễn theo (1), tuy nhiên giá trị ℓ lại không phải là nhỏ nhất. Dạng biểu diễn này được giới thiệu bởi Booth [1] và Reitweisner [10]. Trong năm 1989, Jedwab và Michel [4] đã đề xuất ứng dụng mật mã khi áp dụng dạng NAF để giảm số lượng các phép nhân trong phép nhân điểm trên đường cong elliptic. Morain và Olivos [8] năm 1989 đề xuất áp dụng NAF để tăng tốc độ nhân vô hướng trên đường cong elliptic. Joye và Yen [6] vào năm 2000 đã phát triển một thuật toán ghi mã dạng NAF từ trái sang phải; hơn nữa, các tác giả này trong [7] vào năm 2002 đã đề xuất một biểu diễn có ít nhất các chữ số khác 0 của bất kỳ dạng sửa đổi nào. Ngoài ra, chúng ta còn có thể kể một số công trình dành riêng cho các biểu diễn dạng không liền kề như Fong, Hankerson và Menezes [2], Okeya và Takagi [9], Zhang và Wang [11] và Joye và Tymen [5].

Các thuật toán truyền thống để cài đặt phép toán này thường được dựa vào khai triển nhị phân của các số (ví dụ như NAF, wNAF). Tuy nhiên, sự phát triển của các phép toán phức tạp hơn thúc đẩy sự phát triển của các phép nhân vô hướng sử dụng các cơ số khác 2 (radix-r NAF [15]) hoặc tổ hợp của các cơ số khác nhau (các phương pháp tam phân/nhị phân [17], hệ hai cơ số DB [16] mà cho phép giảm độ dài của khai triển vô hướng và dẫn đến nếu các phép toán phức tạp đủ hiệu quả thì sẽ rút gọn được thời gian cần thiết thực hiện phép nhân vô hướng.

II. ĐẶT VẤN ĐỀ

Trong các ứng dụng mật mã chúng ta thường gặp một số tính toán có chi phí lớn như tính lũy thừa số mũ k (ở đây k là một số nguyên dương) trên các trường hữu hạn hay phép nhân điểm với k trên các đường cong elliptic. Phương pháp chủ yếu để thực hiện việc làm trên là "bình phương-nhân" (hay tương tự "gấp đôi-cộng") và được thể hiện trong thuật toán sau.

Biết rằng nếu $k = \sum_{i=0}^{\ell-1} k_i 2^i$ với $k_{\ell-1} \neq 0$ và $k_i \in (0, \pm 1)$ thì để tính $g^k \bmod p$ ta có thể thực hiện theo thuật toán sau.

Thuật toán 1. (tính $g^k \bmod p$ theo phương pháp "bình phương-nhân")

Input: Số nguyên dương $k = \sum_{i=0}^{\ell-1} k_i 2^i$ với $k_i \in (0, \pm 1)$, (1)

$g \in GF^*(p), g' = g^{-1} \bmod p$.

Output: $g^k \bmod p$.

1. $a \leftarrow 1$.
2. For i from $\ell - 1$ downto 0 do
   2.1 $a \leftarrow a^2 \bmod p$.
   2.2 if $k_i = 1$ then $a \leftarrow a * g \bmod p$.
   2.3 if $k_i = -1$ then $a \leftarrow a * g' \bmod p$.
3. return a.

Theo thuật toán trên thì chi phí tính toán sẽ bằng đúng $\ell - 1$ phép bình phương và $w - 1$ phép nhân, trong đó w là số các ký tự $k_i \neq 0$ trong công thức 1. Như vậy bài toán tìm biểu diễn của k theo công thức (1) sao cho $\ell + w$ bé nhất được đặt ra một cách tự nhiên nhằm giảm chi phí tính toán cho thuật toán 1.

Trong bài viết này sau khi giới thiệu lại khái niệm NAF và các kết quả đã có về nó trong mục 3 thì tại mục 4 chúng tôi đưa ra một thuật toán mới để tính NAF(k). Thuật toán mới đưa ra được đánh giá qua mệnh đề 4 là hiệu quả hơn thuật toán đã có. Cuối cùng, tại mục 5, đưa ra một khái niệm mới cho dạng biểu diễn, ký hiệu là aNAF, với các ý nghĩa quan trọng đó là:

- Tổng $w + \ell$ của aNAF(k) luôn không vượt quá của NAF(k).
- Số các giá trị k trong $[2^{\ell-1}, 2^{\ell})$ với $\ell > 1$ là không dưới $\frac{2^{\ell-1}}{8}$.

III. DẠNG KHÔNG LIỀN KỀ (NAF)

Tại mục này chúng tôi sẽ trình bày lại khái niệm NAF và các kết quả đã có về NAF theo (xem trang 98 trong [3])

A. Khái niệm NAF và các tính chất của NAF(k)

Định nghĩa 1. (xem [3]) Dạng NAF (non-adjacent form) của một số nguyên dương k là biểu thức $k = \sum_{i=0}^{\ell-1} k_i 2^i$ ở đây $k_{\ell-1} \neq 0, k_i \in (0, \pm 1)$ và không có hai ký tự k_i liền nhau nào đều khác 0. Khi này ℓ được gọi là độ dài của NAF còn $k_{\ell-1} k_{l-2} \ldots k_0$ được gọi là biểu diễn NAF của k, ký hiệu là NAF(k).

Định lý 1. (xem [3]) (tính chất của các NAF)

(i) Với mọi k có duy nhất NAF(k).

(ii) NAF(k) có số ký tự khác 0 là nhỏ nhất trong mọi biểu diễn nhị phân có dấu của k.

(iii) Nếu $2^{\ell-1} \leq k < 2^{\ell}$ thì độ dài NAF(k) bằng $\ell + \delta$ với $\delta \in \{0,1\}$.

(iv) Nếu độ dài NAF(k) = ℓ thì $\frac{2^{\ell}}{3} < k < \frac{2^{\ell+1}}{3}$.

(v) Số các ký tự khác 0 trung bình trong tất cả các NAF độ dài ℓ là xấp xỉ $\frac{\ell}{3}$.

B. Thuật toán tính NAF(k)

Cũng trong [3], thuật toán tìm NAF(k) được trình bày như sau.

Thuật toán 2. (xem [3])

Input: Số nguyên dương k.

Output: NAF(k).

1. $\ell \leftarrow 0$.
2. While $k \geq 1$ do
   2.1 if k is odd then: $k_{\ell} \leftarrow 2 - (k \bmod 4)$, $k \leftarrow k - k_{\ell}$;

2.2 else: $k_\ell \leftarrow 0$.

2.3 $k \leftarrow \frac{k}{2}, \ell \leftarrow \ell + 1$.

3. return $k_{\ell-1}k_{\ell-2} \dots k_0$.

IV. THUẬT TOÁN MỚI TÌM NAF(k)

Trong mục này chúng tôi đưa ra thêm một thuật toán mới (thuật toán 3) để tính NAF(k).

A. Thuật toán tìm NAF(k) mới

Thuật toán 3.

Input: Số nguyên dương k.

Output: NAF(k).

1. $k_{\ell-1}k_{\ell-2} \dots k_0 \leftarrow$ Binary(k).
2. $j \leftarrow 0$.
3. While $j < \ell$ do:

3.1 While $k_j = 0$ do $j \leftarrow j + 1$.

3.2 if ($j = \ell - 1$) or ($k_{j+1} = 0$) then $j \leftarrow j + 1$;

3.3 else:

3.3.1 $k_j \leftarrow -1$.

3.3.2 $j \leftarrow j + 1$.

3.3.3 while ($k_j = 1$) do: $k_j \leftarrow 0, j \leftarrow j + 1$.

3.3.4 $k_{j+1} \leftarrow 1$.

3.3.5 If ($j = \ell$) $\ell \leftarrow \ell + 1$.

4. return $k_{\ell-1}k_{\ell-2} \dots k_0$.

Ở trên Binary(k) là biểu diễn nhị phân không dấu của k.

B. Tính đúng đắn của thuật toán 3

Tính đúng đắn của thuật toán 3 được cho bởi kết quả sau.

Mệnh đề 2. *Thuật toán 3 là đúng đắn tức là dãy $k_{\ell-1}k_{\ell-2} \dots k_0$ thu được tại bước 4 chính là NAF(k).*

Chứng minh. Trước hết ta có dãy $k_{\ell-1}k_{\ell-2} \dots k_0$ thu được sau bước 1 thỏa mãn $k_j \in \{0,1\}$ với mọi $j = 0, \dots, \ell - 1$ và sự thay đổi các giá trị này chỉ xảy ra tại 3.3.1, 3.3.3 và 3.3.4 mà các giá trị được thay đổi tương ứng trong các bước này là – 1, 0 và 1 nên dãy $k_{\ell-1}k_{\ell-2} \dots k_0$ thu được sau bước 3 thỏa mãn $k_j \in \{0, \pm 1\}$. Như vậy để chứng minh mệnh đề này ta chỉ cần chỉ ra rằng không tồn tại hai ký tự khác 0 liền nhau của dãy đầu ra.

Biết rằng trong dãy đầu ra thì ký tự $k_j \neq 0$ chỉ xuất hiện ngay sau bước 3.1. Ta có:

Nếu điều kiện của bước 3.2 được thỏa mãn thì:

- Hoặc là ta đã xét đến $j = \ell - 1$ và khi này dãy $k_{\ell-1}k_{\ell-2} \dots k_0 = 10k_{\ell-3} \dots k_0$.
- Hoặc là đoạn $\dots k_{j+1}k_jk_{j-1} \dots k_0 = \cdots 010 \dots k_0$ và do k_{j+1} sẽ vẫn bằng 0 theo bước 3.1 của vòng sau nên ký tự $k_j = 1$ này không có ký tự bên cạnh nào cũng khác 0.

Ngược lại, theo các bước của 3.3 thì nếu có $t > 1$ ký tự liên tiếp (tính từ k_j) bằng 1 tức là $k_{j+t-1}k_{j+t-2} \dots k_j = 11 \dots 1$ thì bước này sẽ biến đổi đoạn dãy trên thành $k_{j+t}k_{j+t-1} \dots k_j = 1\underbrace{0 \dots 0}_{t-1} - 1$ như vậy trước ký tự $k_j = -1$ có ít nhất $t - 1$ ký tự bằng 0. Lại do nếu $j = 0$ thì không tồn tại ký tự sau k_j còn ngược lại thì theo 3.1 thì ký tự ngay trước k_j phải bằng 0. Tóm lại ta cũng có k_j không liền kề với một ký tự khác 0 nào.

Tóm lại mệnh đề đã được chứng minh.

Ngoài ra chúng ta dễ dàng kiểm tra được rằng thuật toán 3 có tính chất sau.

Tính chất 3. *Với mọi j, sau khi xác định được ký tự thứ j của NAF(k) thì tất cả các ký tự k_i với $i > j$ vẫn được giữ nguyên chỉ trừ ra j là vị trí cuối cùng của vòng lặp 3.3 (khi này k_{j+1} phải là 0 và được đổi thành 1).*

C. So sánh tính hiệu quả của hai thuật toán 2 và 3

Việc so sánh tính hiệu quả giữa hai thuật toán 2 và 3 được cho trong kết quả sau.

Mệnh đề 4. *Thuật toán 3 là hiệu quả hơn thuật toán 2.*

Chứng minh. Hai thuật toán đều phải thực hiện xác định ℓ giá trị k_j ($j = 0, \dots, \ell - 1$).

Đối với thuật toán 2 cần:

- Một phép tính k mod 4, một phép trừ 2 cho k mod 4 ($k_j = 2 - (k \bmod 4)$), một phép trừ k cho k_j trong trường hợp k lẻ.
- Một phép chia k cho 2 cho mọi trường hợp.

Đối với thuật toán 3 cần:

- Một phép chia k cho 2 cho mọi trường hợp (để tìm khai triển nhị phân của k).
- Một phép đặt $k_j = -1$ khi j bắt đầu vào vòng 3.3, một phép đảo bít (0 thành 1 hoặc ngược lại) khi j ở bước tiếp theo cho đến khi ra ngoài vòng 3.3.

Như vậy nếu $k_j = 0$ thì cả hai thuật toán đều có chi phí tính toán như nhau. Ngược lại thì thuật toán 2 cần thực hiện thêm ba phép tính số học so với thuật toán 3. Tóm lại mệnh đề đã được chứng minh.

D. Ví dụ

Tính NAF(348).

Các bước thực hiện theo thuật toán 2.

j	k_j	$k = (k - k_j)/2$	j	k_j	$k = (k - k_j)/2$
0	0	174	5	−1	6
1	0	87	6	0	3
2	−1	44	7	−1	2
3	0	22	8	0	1
4	0	11	9	1	0

Các bước thực hiện theo thuật toán 3.

Bước 1. Binary(348) = 101011100, $\ell = 9$.

Bước 3.

	Các bước thực hiện	$k_{\ell-1}k_{\ell-2}\dots k_0$	j	ℓ
Vòng 1	3.1	1 0 1 0 1 1 1 0 0	2	
	3.3	1 0 1 1 0 0–1 0 0	5	9
Vòng 2	3.3	1 1 0–1 0 0–1 0 0	7	9
Vòng 3	3.3	1 0–1 0–1 0 0–1 0 0	8	10
Vòng 4	3.2	1 0–1 0–1 0 0–1 0 0	10	10

V. DẠNG BIỂU DIỄN HẦU KHÔNG LIỀN KỀ

Trong phần này, chúng tôi đưa ra một dạng biểu diễn mới cho một số nguyên dương k, được gọi hầu không liền kề, được kí hiệu là aNAF(k). Chú ý, trong dạng biểu diễn này, sẽ giữ nguyên dạng biểu diễn của NAF của k chỉ có một sự điều chỉnh khi dạng biểu diễn NAF(k) có mẫu $k_{l-1}k_{l-2}k_{l-3}$ có dạng mẫu 1 0 -1 khi đó được thay thế bởi 11. Cụ thể, được định nghĩa như sau:

Định nghĩa 2. *Cho k là một số nguyên dương. Khi đó ta gọi khai triển NAF mở rộng của k, ký hiệu là $aNAF(k)$, được xác định theo công thức sau*

$$aNAF(k) = \begin{bmatrix} 11k_{\ell-4}\dots k_0 & khi\ NAF(k) = 10-1k_{\ell-4}\dots k_0 \\ NAF(k) & trong\ trường\ hợp\ còn\ lại \end{bmatrix}. \quad (2)$$

Như vậy, dạng aNAF(k) sẽ không là NAF(k) khi $k_{l-1}k_{l-2}k_{l-3}$ có dạng mẫu $10-1$. Tuy nhiên, ta vẫn thấy có quan hệ 1-1 trong dạng biểu diễn aNAF và NAF. Hơn nữa, ta có tính chất như sau.

Định lý 5.

(i) *Với mọi số nguyên dương k thì số ký tự khác 0 của aNAF(k) và NAF(k) là bằng nhau.*

(ii) *Với $2^{\ell-1} \le k < 2^{\ell}(\ell > 1)$ thì số các aNAF(k) có độ dài nhỏ hơn độ dài của NAF(k) là không dưới $\frac{2^{\ell-1}}{4}$.*

Chứng minh. Từ định nghĩa 2 ta thấy kết quả (i) là hiển nhiên.

Với $\ell = 2$. Ta có 2 giá trị k thỏa mãn $2^{\ell-1} \le k < 2^{\ell}$ đó là 2 và 3.

NAF(2) = 1 0, NAF(3) = **1 0 –1** . Dãy sau có độ dài lớn hơn dãy aNAF, trong khi $\frac{2^{\ell-1}}{4} = \frac{2}{4} < 1$ vậy (ii) đã thỏa mãn.

Với $\ell = 3$. Ta có 4 giá trị k thỏa mãn $2^{\ell-1} \le k < 2^{\ell}$ đó là 4, 5, 6 và 7.

k	4	5	6	7
NAF(k)	1 0 0	1 0 1	**1 0 –1 0**	1 0 0 –1

Số các NAF(k) có độ dài lớn hơn dãy aNAF(k) là 1, trong khi $\frac{2^{\ell-1}}{4} = \frac{2^2}{4} = 1$ vậy (ii) đã thỏa mãn.

Với $\ell = 4$. Ta có 4 giá trị k thỏa mãn $2^{\ell-1} \le k < 2^{\ell}$ đó là 8, 9, 10, 11, 12, 13, 14 và 15.

k	8	9	10	11
NAF(k)	1 0 0 0	1 0 0 1	1 0 1 0	**1 0 –1 0 –1**
k	12	13	14	15
NAF(k)	**1 0 –1 0 0**	**1 0 –1 0 1**	1 0 0 –1 0	1 0 0 0 –1

Số các NAF(k) có độ dài lớn hơn dãy aNAF(k) là 3, trong khi $\frac{2^{\ell-1}}{4} = \frac{2^3}{4} < 3$ vậy (ii) đã thỏa mãn.

Với $\ell > 4$.

Xét tập hai tập sau

$$R(\ell) = \{k: 2^{\ell-1} \le k < 2^{\ell} \text{ và Binary(k)} = 1011k_{\ell-5}\dots k_0\}. \quad (3)$$

$$S(\ell) = \{k: 2^{\ell-1} \le k < 2^{\ell} \text{ và Binary(k)} = 1100k_{\ell-5}\dots k_0\}. \quad (4)$$

Ta có hai tập trên đều có đúng $\frac{2^{\ell-1}}{8}$ phần tử cho nên nếu ta chỉ ra được mọi phần k tử thuộc các tập trên đều thỏa mãn NAF(k) có độ dài lớn hơn dãy aNAF(k) thì (ii) đã được chứng minh.

Nếu $k \in R$, tức là $\text{Binary(k)} = 1011k_{\ell-5}\dots k_0$. Theo tính chất 3 thì khi sau khi xác định được $k_{\ell-5}$ thì do ký tự $k_{\ell-4}$ (vốn bằng 1) nên bước $\ell - 5$ chỉ xảy ra một trong hai khả năng sau:

- Là bước trung gian của vòng 3.3 , điều này chỉ xảy ra với $k_{\ell-5} = 1$. Khi này bước cuối cùng của 3.3 sẽ ứng với $j = \ell - 2$ và sau vòng này dãy trở thành $1100k_{\ell-5}\dots k_0$. Tiếp đây sẽ là bước tiếp theo của 3 với vòng lặp 3.3 để thu được dãy đầu ra là $10-100k_{\ell-5}\dots k_0$.
- Là bước cuối của 3.1, điều này chỉ xảy ra với $k_{\ell-5} = 0$. Khi này dãy chính là $10110k_{\ell-6}\dots k_0$ và bước tiếp theo của 3 chính là 3.3 chúng sẽ chuyển dãy trên trở thành $10-101k_{\ell-6}\dots k_0$.

Tóm lại cho dù k thuộc R hay S thì độ dài NAF(k) đều lớn hơn độ dài aNAF(k).

Tương tự, nếu $k \in S$, tức là $\text{Binary(k)} = 1100k_{\ell-5}\dots k_0$. Theo tính chất 3 thì khi sau khi xác định được $k_{\ell-5}$ thì ký tự $k_{\ell-4}$ (vốn bằng 0) nên trong mọi trường hợp nó sẽ nhận giá trị trong {0, 1} cho nên từ $k_{\ell-3} = 0$ giá trị $k_{\ell-4}$ và $k_{\ell-3}$ sẽ không thay đổi từ đó vòng lặp tiếp theo của bước 3 sẽ bắt đầu từ 3.1 cho đến khi $j = \ell - 2$. Tiếp đến sẽ là bước lặp 3.3 (điều kiện của 3.2 không được thỏa mãn) sẽ biến $k_{\ell-1}k_{\ell-1} = 1\ 1$ thành 1 0 –1 tức là độ dài NAF(k) dài hơn độ dài aNAF(k).

Qua trên ta có số các giá trị k có độ dài NAF(k) lớn hơn độ dài aNAF(k) là không dưới $\#R + \#S = 2\frac{2^{\ell-1}}{8} = \frac{2^{\ell-1}}{4}$ vậy định lý đã được chứng minh.

VI. KẾT LUẬN

Tóm lại, chúng tôi đã đưa ra thuật toán 3 cho việc tìm dạng biểu diễn không liền kề NAF của một số nguyên k cho trước. Thuật toán này hoạt động dựa trên dạng nhị phân của số đó, trong quá trình tính toán thuật toán 3 tiết kiệm được ba phép tính số học so với thuật toán 2 đã có. Cuối cùng, với dạng biểu diễn hầu không liền kề aNAF mới của số k, chúng ta đã rút gọn một phép tính bình phương khi thực hiện phép "bình phương – nhân" đối với phép tính lũy thừa k trên các trường hữu hạn, hoặc tương ứng một phép tính gấp đôi khi thực hiện phép "gấp đôi - cộng" trong

phép nhân điểm trên đường cong elliptic.

VII. TÀI LIỆU THAM KHẢO

[1]. Andrew D Booth, *A signed binary multiplication technique.* The Quarterly Journal of Mechanics and Applied Mathematics, 1951. **4**(2): p. 236-240.

[2]. Kenny Fong, Darrel Hankerson, Julio López, and Alfred Menezes, *Field inversion and point halving revisited.* IEEE Transactions on Computers, 2004. **53**(8): p. 1047-1059.

[3]. Darrel Hankerson, Alfred J Menezes, and Scott Vanstone, *Guide to elliptic curve cryptography*. 2006: Springer Science & Business Media.

[4]. Jonathan Jedwab and Chris J Mitchell, *Minimum weight modified signed-digit representations and fast exponentiation.* Electronics Letters, 1989. **25**(17): p. 1171-1172.

[5]. Marc Joye and Christophe Tymen. *Compact encoding of non-adjacent forms with applications to elliptic curve cryptography*. in *International Workshop on Public Key Cryptography*. 2001. Springer.

[6]. Marc Joye and Sung-Ming Yen, *Optimal left-to-right binary signed-digit recoding.* IEEE Transactions on Computers, 2000. **49**(7): p. 740-748.

[7]. Marc Joye and Sung-Ming Yen. *New minimal modified radix-r representation with applications to smart cards*. in *International Workshop on Public Key Cryptography*. 2002. Springer.

[8]. François Morain and Jorge Olivos, *Speeding up the computations on an elliptic curve using addition-subtraction chains.* RAIRO-Theoretical Informatics and Applications, 1990. **24**(6): p. 531-543.

[9]. Katsuyuki Okeya and Tsuyoshi Takagi. *The width-w NAF method provides small memory and fast elliptic scalar multiplications secure against side channel attacks*. in *Cryptographers' Track at the RSA Conference*. 2003. Springer.

[10]. George W Reitwiesner, *Binary arithmetic*, in *Advances in computers*. 1960, Elsevier. p. 231-308.

[11]. Jing Zhang and Pingan Wang. *Non-adjacent form recursive algorithm on elliptic curves cryptograph*. in *2010 International Conference on Computational Intelligence and Security*. 2010. IEEE.

[12]. P. Longa and A. Miri, "*Fast and Flexible Elliptic Curve Point Arithmetic over Prime Fields*," in *IEEE Transactions on Computers*, Vol. 57, No 3, pp. 289-302, 2008.

[13] P. Longa and A. Miri, "*New Composite Operations and Precomputation Scheme for Elliptic Curve Cryptosystems over Prime Fields*," *International Conference on Practice and Theory in Public Key Cryptography* (PKC 2008), LNCS Vol. 4939, pp. 229-247, Springer, 2008.

[14] V. Dimitrov, L. Imbert and P.K. Mishra, "*Efficient and Secure Elliptic Curve Point Multiplication using Double-Base Chains*," Advances in Cryptology (ASIACRYPT'05), LNCS Vol. 3788, pp. 59–78, Springer-Verlag, 2005.

[15]. T. Takagi, S-M. Yen and B-C. Wu, "*Radix-r Non-Adjacent Form*" in International Conference on Information Security (ISC'04), LNCS Vol. 3225, pp. 99-110, Springer-Verlag, 2004.

[16]. V. Dimitrov, L. Imbert and P.K. Mishra, "*Efficient and Secure Elliptic Curve Point Multiplication using Double-Base Chains*" Advances in Cryptology (ASIACRYPT'05), LNCS Vol. 3788, pp. 59–78, Springer-Verlag, 2005.

[17]. M. Ciet, M. Joye, K. Lauter and P. L. Montgomery, "*Trading Inversions for Multiplications in Elliptic Curve Cryptography*" in Designs, Codes and Cryptography, Vol. 39, No 2, pp.189-206, 2006.

SOME EXTENDED RESULT FOR NON-ADJACENT FORM OF POSITIVE INTEGER NUMBERS

Abstract: In this paper, we introduce a new modification algorithm for finding the non-adjacent form (NAF) of the positive integer k. The correctness of the algorithm is analyzed in detail along with some evaluation of the effectiveness of the proposed algorithm. Finally, we propose a modification form of NAF to increase the implementation performance of some arithmetic operators, such as exponent operator over finite fields or point multiplication on elliptic curves.

Phạm Văn Lực, nhận học vị Thạc sỹ năm 2008. Hiện công tác Viện Khoa học – Công nghệ mật mã. Lĩnh vực nghiên cứu: Công nghệ mật mã và an toàn thông tin

Lều Đức Tân, nhận học vị Tiến sĩ năm 1992. Nguyên cán bộ nghiên cứu tại Học viện Kỹ thuật mật mã. Lĩnh vực nghiên cứu: Khoa học mật mã.