BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Đặng Vân Trường

VỀ MỘT THUẬT TOÁN SINH SỐ GIẢ NGẪU NHIÊN DỰA TRÊN PHƯƠNG PHÁP TẠO DÃY PHI TUYẾN LỒNG GHÉP VỚI BẬC LỚN

Chuyên ngành: **Kỹ thuật Điện tử**
Mã số: **9.52.02.03**

TÓM TẮT LUẬN ÁN TIẾN SĨ KỸ THUẬT ĐIỆN TỬ

HÀ NỘI - 2022

Công trình được hoàn thành tại:
Học viện Công nghệ bưu chính viễn thông

Phản biện 1:……..
Phản biện 2: ……
Phản biện 3: ……

Luận án được bảo vệ trước Hội đồng chấm luận án cấp Học viện tại Học viện Công nghệ Bưu chính viễn thông, KM 10 đường Nguyễn Trãi, Hà Đông, Hà Nội.
Họp tại: ……
Vào hồi……giờ……ngày……tháng……năm 2022

Có thể tìm hiểu luận án tại thư viện Học viện Công nghệ Bưu chính viễn thông.

MỞ ĐẦU

Tính cấp thiết của đề tài luận án

Bài toán tạo ra các dãy số giả ngẫu nhiên (pseudo noise – PN) là bài toán luôn được quan tâm nghiên cứu phát triển trong những năm gần đây, phục vụ nhiều yêu cầu trong thực tế. Dãy giả ngẫu nhiên được sử dụng phổ biến nhất là m-dãy. Các bộ tạo m-dãy được S.W. Golomb đặt nền móng từ thập kỷ 1960, dựa trên lý thuyết trường Galois. Tiến sỹ Lê Chí Quỳnh đã đặt nền móng cho dãy lồng ghép từ năm 1986, tiếp sau đó TS Lê Minh Hiếu và TS Bùi Lai An tiếp tục nghiên cứu về dãy lồng ghép tam phân, dãy phi tuyến lồng ghép và dãy lồng ghép đa cấp, đưa ra ba phương pháp sinh dãy phi tuyến lồng ghép dựa trên m-dãy.

Trong hội nghị Asia Crypt 2004, chuyên gia mật mã Shamir đã chỉ rõ các lợi thế và hướng phát triển của mã dòng và m-dãy. Để có thể ứng dụng m-dãy trong các hệ mã dòng, một trong những tham số quan trọng nhất là độ lớn bậc của đa thức đặc trưng. Các phương pháp sinh dãy lồng ghép trước đây có độ phức tạp tính toán cao, khó có thể triển khai trong thực tế với bậc đa thức lớn. Trong luận án này, tác giả sẽ nghiên cứu đề xuất đưa ra thêm một phương pháp sinh dãy phi tuyến lồng ghép có thể sử dụng trong kỹ thuật mật mã, cùng với một thuật toán hiệu quả để sinh dãy phi tuyến lồng ghép với bậc lớn.

Mục tiêu nghiên cứu

Nghiên cứu các phương pháp sinh dãy phi tuyến lồng ghép, từ đó đề xuất một phương pháp khả thi trong thực hành để sinh dãy phi tuyến lồng ghép có thể ứng dụng trong kỹ thuật mật mã; Đề xuất một thuật toán hiệu quả để sinh dãy phi tuyến lồng ghép với bậc lớn, phân tích đánh giá thuật toán đã đề xuất.

Ý nghĩa khoa học và đóng góp

Các giải pháp được đề xuất trong luận án có thể giúp đưa các nghiên cứu về dãy phi tuyến lồng ghép vào ứng dụng trong kỹ thuật mật mã của ngành Cơ yếu Việt Nam. Luận án gồm có 02 đóng góp sau:

1) Đề xuất một giải pháp sinh dãy phi tuyến lồng ghép dựa trên kỹ thuật phân rã theo bước và kỹ thuật tính một phần thứ tự lồng ghép. Giải pháp này có thể ứng dụng trong thực tế để sinh ra một đoạn có kích thước tùy ý của dãy phi tuyến lồng ghép.

2) Đề xuất đề xuất một thuật toán hiệu quả để sinh dãy phi tuyến lồng ghép với bậc lớn. Độ phức tạp tính toán của thuật toán là cỡ $O(n^3)$ so với bậc n của đa thức sinh m-dãy. Độ phức tạp lưu trữ là m.n. Bằng cách khai thác một đặc điểm của tham số của dãy lồng ghép, thuật toán này có lợi thế lớn hơn so với thuật toán bình phương và nhân thông thường.

Bố cục của luận án

Nội dung luận án được trình bày trong 3 chương. Chương 1 trình bày tổng quan về bộ tạo dãy giả ngẫu nhiên dựa trên m-dãy, cùng với một số ứng dụng của m-dãy trong mật mã. Chương 2 trình bày các phương pháp xây dựng dãy phi tuyến lồng ghép dựa trên m-dãy và đề xuất giải pháp sinh từng phần dãy phi tuyến lồng ghép. Chương 3 đề xuất thuật toán sinh dãy phi tuyến lồng ghép với bậc lớn ứng dụng trong kỹ thuật mật mã với các đánh giá về độ phức tạp tính toán, độ phức tạp lưu trữ và đề xuất một hệ mã dòng sử dụng dãy phi tuyến lồng ghép có thể ứng dụng trong mật mã.

CHƯƠNG 1. TỔNG QUAN VỀ BỘ TẠO DÃY GIẢ NGẪU NHIÊN DỰA TRÊN M-DÃY

1.1. Khái niệm trường Galois

Trường Galois hay còn gọi là trường modulo theo đặc số p với p là số nguyên tố, ký hiệu là GF(p), bao gồm tập xác định gồm p số nguyên [0, 1, …, p-1] và hai phép toán ký hiệu là $\oplus$ và $\otimes$, là phép cộng và nhân số nguyên theo modulo p. Ta quan tâm tới trường hợp p>2, khi đó phép chia modulo (là phép toán ngược với phép $\otimes$) sẽ được thực hiện bằng cách áp dụng thuật toán Euclid mở rộng.

Phép mở rộng trường GF(p^n) cho phép mở rộng trường Galois cho một vector gồm n phần tử. Trạng thái của vector tại một thời điểm $A = \{a_0, a_1, \ldots, a_{n-1}\}$ có thể được biểu diễn như một đa thức với biến x:

$$A(x) = a_0 + a_1x + a_2x^2 + \ldots + a_{n-1}x^{n-1}.$$

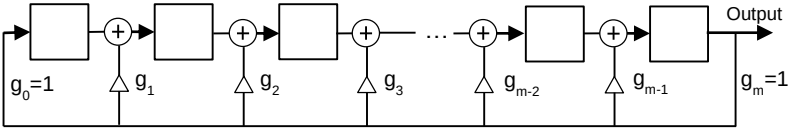
Từ trường GF(p^n) với một đa thức đặc trưng $g(x)$ bậc n là đa thức nguyên thủy, ta xây dựng nên một m-dãy, còn gọi là dãy có chu kỳ cực đại. Trạng thái m-dãy sau t bước bắt đầu từ trạng thái $S(x)$ là:

$$A(x) = S(x)*x^t/g(x)\ . \qquad (1.1)$$

Chú ý là phép modulo theo đa thức $g(x)$ sẽ tính theo nguyên tắc modulo theo hệ số bậc cao nhất, nghĩa là một vector $S(x)$ với biểu diễn thành đa thức có bậc không nhỏ hơn n sẽ cần chia cho $g(x)$ để lấy được phần dư là một đa thức kết quả $S'(x)$ có bậc nhỏ hơn n, gọi là modulo của $S(x)$ theo $g(x)$.

Có hai phương pháp xây dựng m-dãy trên GF(p^n):

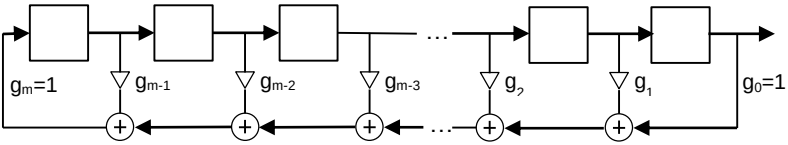
Phương pháp Galois áp dụng trực tiếp công thức xây dựng dãy trên trường GF(p^n) đã nêu ở trên.



Phương pháp Fibonacci áp dụng công thức tính truy hồi kiểu Fibonacci (còn gọi là thanh ghi dịch phản hồi tuyến tính – LFSR), ứng dụng tính chất phụ thuộc tuyến tính của m-dãy.

$$\sum_{i=0}^{n} A_i \otimes g_{n-i} = 0 \,. \qquad (1.2)$$

Phương pháp Fibonacci thường được sử dụng để cài đặt thuật toán sinh m-dãy trên máy tính và các hệ vi xử lý.



Nếu ta thực hiện thuật toán trên trường GF(2), khi đó phép nhân được thay bằng phép AND, phép cộng modulo thay bằng phép XOR, việc lấy nghịch đảo modulo và trừ modulo không cần thực hiện. Khi đó ta sẽ không còn các phép tính số học trên số nguyên mà chỉ còn các phép tính logic trên bit. Trong trường hợp tổng quát ta có thể phải thực hiện phép tính nghịch đảo module trong khi sinh 1 bit của dãy đầu ra:

$$A_n = -g_n^{-1} \sum_{i=0}^{n-1} A_i \otimes g_{n-1-i} \,. \qquad (1.3)$$

Để cài đặt thuật toán sinh m-dãy trên máy tính, thanh ghi A sẽ được lưu trữ trong một mảng số nguyên. Các hệ số của đa thức sinh g(x) cũng được lưu trữ trong mảng số nguyên tương ứng. Để sinh ra một bit đầu ra của dãy, ta thực hiện tính giá trị phản hồi theo công thức phụ thuộc tuyến tính, sau đó dịch toàn bộ thanh ghi về vị trí ô số 0. Giá trị còn trống trong thanh ghi sẽ được gán bằng giá trị phản hồi vừa tính được.

1.2. Ứng dụng của dãy giả ngẫu nhiên dựa trên m-dãy

Dãy giả ngẫu nhiên dựa trên m-dãy có rất nhiều ứng dụng trong hệ thống truyền thông hiện đại, máy tính và nhiều các thiết bị điện tử khác. Trong mạng di động, dãy giả ngẫu nhiên có ứng dụng với thuật toán A5 để mã hóa dữ liệu GSM, phân chia miền tần số cho các CDMA. Trong nhiều chuẩn truyền thông nối tiếp, dãy giả ngẫu nhiên được sử dụng cho việc ngẫu nhiên hóa dữ liệu trên kênh truyền như Bluetooth, USB… để dạng tín hiệu trên kênh gần giống như nhiễu trắng, sử dụng tính chất phân bố cân bằng của m-dãy.

Ứng dụng quan trọng của m-dãy trong mật mã là các hệ mã dòng dựa trên m-dãy, được sử dụng rộng rãi từ thập niên 1960. Cùng với sự phát triển của khoa học mật mã, có nhiều hệ mật hiện đại đã xuất hiện như mã khối, mã hóa khóa công khai song mã dòng vẫn giữ được chỗ đứng nhất định do tính đơn giản trong thiết kế và ứng dụng. Một dãy giả ngẫu nhiên từ một m-dãy độc lập có tính tuyến tính cao, do đó không thể sử dụng trực tiếp trong kỹ thuật mật mã. Giải pháp thường được sử dụng là kết hợp nhiều m-dãy với nhau để thiết kế lên một hệ mã dòng.

Tuy nhiên các yêu cầu đặt ra với mật mã dòng ngày càng tăng, do việc phát triển các kỹ thuật phân tích mã để tấn công các hệ mã dòng. Cùng với sự tăng trưởng không ngừng của công nghệ, sức mạnh của siêu máy tính mạnh nhất gần đây đã đạt tới 2^{60} phép tính/giây. Chính vì thế các hệ mã dòng sử dụng m-dãy thường yêu cầu các dãy thành phần có bậc tối thiểu là 128 để bảo đảm an toàn về mật mã.

1.3. Một số bộ tạo dãy giả ngẫu nhiên dựa trên m-dãy

Có 03 bộ tạo dãy bộ tạo dãy giả ngẫu nhiên có khả năng ứng dụng trong mật mã được xây dựng bằng cách kết hợp nhiều m-dãy thành phần với các tính chất tốt về mật mã.

Bộ tạo dãy Gold được xây dựng từ 2 m-dãy cùng chu kỳ kết hợp với nhau theo công thức cộng và dịch như sau:

$G(a,b) = \{a, b, a \oplus b, a \oplus Tb, a \oplus T^2b, \ldots, a \oplus T^{N-1}b\}$

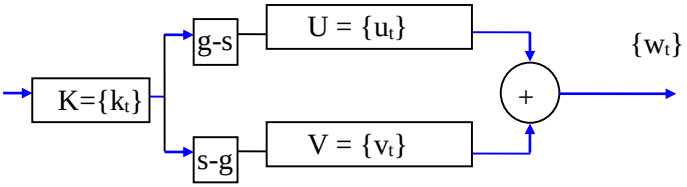
Trong đó, T là phép dịch dãy, $G(a,b)$ chứa $N + 2 = 2^m + 1$ dãy có chu kì $N = 2^m - 1$.

Dãy Gold tạo nên một tập hợp lớn các dãy có tính chất tương quan ACF và CCF tốt nhưng giá trị khoảng tuyến tính chưa cao.

Bộ tạo dãy tựa Gold (Gold-like) là một cải tiến của dãy Gold trong đó phép dịch dãy đơn giản được thay bằng ba phép dịch theo mẫu phức tạp hơn, từ đó dãy đầu ra có khoảng tuyến tính cao hơn

$$H(a,b) = \{a, a \oplus b^{(0)}, a \oplus T^1b^{(0)}, \ldots, a \oplus T^{N'-1}b^{(0)},$$
$$a \oplus b^{(1)}, a \oplus T^1b^{(1)}, \ldots, a \oplus T^{N'-1}b^{(1)},$$
$$a \oplus b^{(2)}, a \oplus T^1b^{(2)}, \ldots, a \oplus T^{N'-1}b^{(2)}\}$$

Bộ tạo dãy luân phiên (ASG) là sự kết hợp giữa hai m-dãy theo kiểu Stop – Go thông qua sự điều khiển của một dãy thứ ba.



Dãy điều khiển của dãy luân phiên là một dãy D'Brujin. Nếu đầu ra của dãy điều khiển là bit 1, dãy U sẽ chạy và dãy V dừng; Ngược lại nếu đầu ra của dãy điều khiển là bit 0, dãy V sẽ chạy và dãy U dừng.

Bộ tạo dãy luân phiên có các tính chất mật mã tốt, bao gồm chu kỳ, độ phức tạp tuyến tính, hàm phân bố, tính tương quan. Một lợi thế lớn của dãy luân phiên là dãy có tốc độ sinh bit khá nhanh so với các bộ tạo dãy kiểu kết hợp tương tự.

Dãy lồng ghép và dãy phi tuyến lồng ghép: Dãy lồng ghép (Interleaved sequence) là một kiến trúc riêng do nhóm nghiên cứu của TS. Lê Chí Quỳnh đề xuất, với mục tiêu xây dựng một dãy giả ngẫu nhiên từ một m-dãy ban đầu với các tham số lồng ghép được lựa chọn theo nguyên tắc riêng. Kế tiếp các nghiên cứu ban đầu về dãy lồng ghép, TS Lê Minh Hiếu tiếp tục nghiên cứu về dãy lồng ghép tam phân và dãy phi tuyến lồng ghép. Tiếp đó, tiến sỹ Bùi Lai An đã phát triển dãy lồng ghép đa cấp, đa chiều 2012.

Dãy phi tuyến lồng ghép là một phát triển của dãy lồng ghép, trong đó sử dụng 2 m-dãy ban đầu, song kết hợp với nhau theo phương pháp đặc trưng của dãy lồng ghép với mục tiêu đưa ra dãy đầu ra có tính phi tuyến cao hơn so với dãy lồng ghép. sử dụng tham số của hai dãy đầu vào có cùng bậc nhưng sinh bởi hai đa thức sinh khác nhau *f(x)* và *g(x)*. Tính chất "phi tuyến" được đề cập ở đây có nghĩa là dãy mới tạo ra có độ phức tạp tuyến tính lớn hơn nhiều so với dãy lồng ghép ban đầu.

Kết luận chương I

Trong chương này tác giả đã trình bày ngắn gọn và rõ ràng về trường Galois và mở rộng trường Galois bằng các khái niệm toán học đơn giản. Trong chương này đã giới thiệu một số ứng dụng của m-dãy và đi sâu phân tích về ứng dụng m-dãy trong các hệ mã dòng, cùng với một số bộ tạo dãy thông dụng dựa trên m-dãy.

CHƯƠNG 2: CÁC PHƯƠNG PHÁP SINH DÃY PHI TUYẾN LỒNG GHÉP DỰA TRÊN M-DÃY

2.1. Kiến trúc dãy lồng ghép

Biểu diễn biến đổi d của một chuỗi $\{b_n\}$ trên GF(p^n) được ký hiệu là D[b_n] và xác định bởi công thức:

$$\mathrm{D}[b_n] = F = \sum_{i=0}^{m} b_i d^i \,, b_i \in \{\mathrm{GF}(p)\}, \qquad (2.1)$$

Biến đổi d của chuỗi $\{b_n\}$ sinh ra từ bộ thanh ghi dịch phản hồi tuyến tính (LFSR) được xác định bởi công thức:

$$b(d) = \frac{S(d)}{g(d)} \qquad (2.2)$$

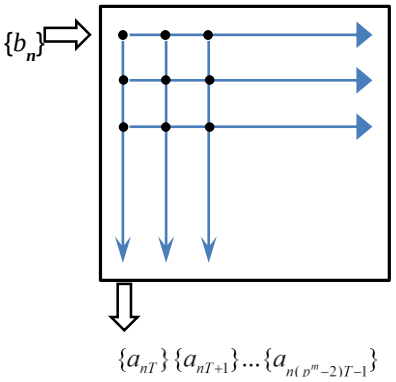
Trong đó *g(d)* có bậc *n* là đa thức sinh của LFSR và *S(d)* xác định giá trị ban đầu của thanh ghi. Khi *g(d)* là đa thức nguyên thủy, chuỗi sinh ra được gọi là m-dãy.

Kiến trúc dãy lồng ghép

Với một m-dãy $\{b_i\}$ được sinh bởi đa thức sinh *g(x)* trên trường GF(p^n). Trong trường hợp $n=m.l$, từ các giá trị $L = p^n-1$, $N = p^m-1$ ta tính ra bước lồng ghép $T = L/N$.

Ta xây dựng lên dãy lồng ghép $\{b_i\}$ bằng cách lồng ghép (T-1) dãy con thành phần, mỗi dãy có độ dài $N = q^m$-l. Các dãy con có được bằng cách áp dụng phép phân rã theo bước (decimation) trên dãy $\{b_i\}$ với bước nhảy T,

Do đó, xét trên miền thời gian, các dãy con này (sắp xếp theo cột) có thể được coi là ghép kênh theo bước thời gian T để đặt vào T khe thời gian như trong sơ đồ dưới đây:



Ta quan tâm tới một tham số quan trọng trong quá trình xây dựng dãy lồng ghép là tập thứ tự lồng ghép I_P^T (cũng là danh sách các bước dịch của dãy con) dưới dạng:

I_P^T = {4, 6, 6, 2, 5, "∞", 2, 0, 5, 6}.

Trong đó ∞ biểu diễn vị trí của dãy con chứa toàn phần tử 0,

Để xây dựng nên dãy lồng ghép, ta cần xác định tập thứ tự lồng ghép I_P^T trong pha chuẩn bị, từ đó có thể xây dựng nên toàn bộ dãy đầu ra mà không cần lập ma trận lồng ghép đầy đủ. Các phương pháp được trình bày sau đây tập trung vào việc xác định tập thứ tự lồng ghép này,

2.2. Các phương pháp để xây dựng dãy lồng ghép p-phân

Phương pháp mở rộng dãy sử dụng biến đổi d

Cho $\{b_n\}$ là một m-dãy sinh bởi đa thức sinh *g(d)* có bậc *n = l.m*

Bước lồng ghép T được lựa chọn như sau:

$$L = p^n - 1 = p^{l*m} - 1 = T.(p^m - 1) = T.N . \quad (2.3)$$

hay $T = (p^n - 1)/(p^m - 1)$.

Gọi $b(d)$ là biến đổi d của $\{b_n\}$, ta có thể biểu diễn $b(d)$ thành

$$b(d) = \sum_{i=0}^{T-1} d^i F_i(d^T) \quad (2.4)$$

với $F_i(d)$ là dãy con được sinh từ đa thức sinh $g_1(d)$ có bậc m

Các pha cụ thể của $\{F_n\}$ trong đó thứ tự lồng ghép có thể được xác định thông qua 3 bước:

Bước 1: mở rộng $F_i(d)$ lên T lần (chèn T-1 số 0 vào giữa hai bit liên tiếp của $F_i(d)$), trong biến đổi d, nó tương đương với việc thay thế d bằng d^T.

Bước 2: Biểu diễn theo biến đổi d của $\{b_n\}$ theo cách để xen kẽ các $F_i(d)$:

$$b(d) = \sum_{i=0}^{T-1} d^i F_i(d^T) = \sum_{i=0}^{T-1} d^i \frac{S_i(d^T)}{g_1(d^T)}. \quad (2.5)$$

Bước 3: Nhóm lại biến đổi d của $b(d)$ thành (với $d^T = D$):

$$b(d) = \sum_{i=0}^{T-1} d^i F_i(D). \quad (2.6)$$

So sánh phần $F_i(D)$ với bảng biến đổi d của từng phần của dãy, ta có thể xây dựng lên toàn bộ các bậc lồng ghép I_P^S.

Phương pháp phân rã m-dãy sử dụng hàm vết

Với *m, n* là các số nguyên dương mà *m* chia hết *n*, α là phần tử sinh của trường hữu hạn GF(p^n) và:

$$T = L/N = (p^n\text{-}1)/(p^m\text{-}1) \; . \qquad (2.7)$$

Khi đó, hàm vết $Tr_m^n(x)$ sẽ ánh xạ các phần tử của GF(p^n) vào GF(p^m) theo quan hệ sau:

$$Tr_m^n(x) = \sum_{k=0}^{n/m-1} x^{p^{mk}} \; . \qquad (2.8)$$

Bậc lồng ghép sẽ được tính theo công thức:

$$I_P^j = \begin{cases} i & if \;\; Tr_m^n(\alpha^j) = \alpha^{iT} \qquad i = 0,1,\dots,p^m - 1 \\ \infty & if \;\; Tr_m^n(\alpha^j) = 0 \qquad j = 0,1,\dots,T-1 \end{cases} \qquad (2.9)$$

Các tính toán thực tế trên một ví dụ cho thấy phương pháp phân rã sử dụng hàm vết cho kết quả trùng khớp với kết quả của phương pháp dùng biến đổi d .

Phương pháp thực hành để phân rã trực tiếp m-dãy

Trước hết ta sinh ra phần đầu của chuỗi $\{b_n\}$ với trạng thái ban đầu được cho trước, nhưng thay vì tạo chuỗi đầy đủ chu kỳ, ta chỉ cần tạo *m*T* giá trị đầu tiên.

Tiếp đó ta sẽ sắp xếp lại các giá trị này bằng cấu trúc xen kẽ như định nghĩa của dãy lồng ghép, từ đó ta có thể nhận được trực tiếp các trạng thái ban đầu của dãy con $F_i(d)$. Vị trí của mỗi trạng thái này trong dãy con chính là giá trị tương ứng trong I_P^T.

2.3. Xây dựng dãy phi tuyến lồng ghép

Trong kiến trúc dãy lồng ghép, dãy đầu ra $\{b_n\}$ về bản chất vẫn là một m-dãy và có tính tuyến tính với bậc tuyến tính rất nhỏ. Để tăng tính phi tuyến cho dãy đầu ra, nếu ta giữ nguyên thứ tự lồng ghép I_P^T nhưng thay các dãy con $\{a_n\}$ bằng các dãy con tương đương, ta sẽ có được một dạng m-dãy phi tuyến, còn gọi là dãy tựa – m. Đầu vào của dãy phi tuyến lồng ghép là hai lồng ghép có cùng bậc n và các tham số lồng ghép, với hai đa thức sinh tương ứng là

$g(d)$ và $f(d)$, hai đa thức sinh cho dãy con tương ứng $g_1(d)$ và $f_1(d)$. Sử dụng thứ tự lồng ghép I_P^T từ dãy lồng ghép thứ nhất nhưng lấy dãy con theo đa thức sinh $f_1(d)$ từ dãy lồng ghép thứ hai, dãy đầu ra $\{e_n\}$ là dãy phi tuyến lồng ghép.

Dãy phi tuyến lồng ghép được sinh ra được chứng minh rằng có giá trị hàm tự tương quan và hàm tương quan chéo có các tính chất tốt, hàm phân bố nhận giá trị phân bố đều,

Khoảng tương đương tuyến tính của các dãy phi tuyến lồng ghép

Khoảng tương đương tuyến tính (Equivalent Linear Span - ELS) của một dãy là bậc nhỏ nhất của đa thức sinh ra toàn bộ dãy đó. Trong luận án này, ta sẽ áp dụng biến đổi d để tính giá trị ELS. Sau khi xác định biến đổi d của dãy phi tuyến lồng ghép đầu ra, áp dụng thuật toán Euclid trên đa thức ta tính được gcd của hai đa thức, chính là đa thức biểu diễn ELS của dãy đầu ra. Tính toán thực nghiệm trên một ví dụ về dãy phi tuyến lồng ghép cụ thể, ta thấy giá trị ELS của dãy đầu ra cao hơn bậc của dãy lồng ghép tương ứng.

Tác giả đã xây dựng chương trình trên máy tính thực hiện các bước xây dựng dãy phi tuyến lồng ghép với bậc nhỏ. Chương trình đã thực hiện tính toán sinh ra tập thứ tự lồng ghép, từ đó sinh ra dãy phi tuyến lồng ghép đầu ra. Điểm khác biệt là chương trình được xây dựng cho nhiều giá trị đặc số p > 2, cụ thể là p=5 và p=7

2.4 Phương pháp phân rã theo bước để sinh dãy lồng ghép

Ngoài ba phương pháp sinh dãy phi tuyến lồng ghép đã được nghiên cứu, trong công bố [J2] tác giả luận án đã đưa ra thêm một phương pháp sinh dãy phi tuyến lồng ghép có tính ứng dụng cao.

Gọi A là một m-dãy với bậc n, chu kỳ 2^n-1 và phần tử sinh α. Ta sẽ sinh ra một dãy mới $A(T)$ bằng cách lấy các bit cách nhau T vị trí từ dãy A, bắt đầu từ bit đầu tiên. $A(T)$ được gọi là dãy phân

rã (hay dãy decimation) theo bước T từ dãy A. Giá trị T gọi là bước phân rã.

Nếu bước phân rã T và độ dài chu kỳ của m-dãy là nguyên tố cùng nhau, thì dãy phân rã theo bước T từ m-dãy ban đầu cũng là một m-dãy với phần tử sinh α^T, các tham số khác của m-dãy phân rã đều giống như dãy ban đầu. Nếu bước phân rã thỏa mãn $T = L / N$ theo các điều kiện của dãy lồng ghép, dãy phân rã chính là dãy con đầu tiên của dãy lồng ghép được trình bày trong phần 2.2 của trong chương này.

Giải pháp để xây dựng dãy phân rã một cách hiệu quả

Ta cần tính toán T trạng thái trong của m-dãy để tạo ra 1 bit của dãy phân rã đầu ra. Nếu ta sử dụng biến đổi - d để tính toán, ta có thể tính trực tiếp trạng thái trong sau bước T bằng trạng thái trong hiện tại nhân với d^T trên trường GF(q^n).

Khi sử dụng phương pháp Fibonacci, ta có thể thực hiện phân rã nhanh hơn bằng cách thực hiện các tính toán trước như sau:

Giả sử trạng thái trong của m-dãy được lưu trong một thanh ghi dịch ký hiệu $\{a_i\}$ (i = 0..n-1). Từ các bit của trạng thái ban đầu $S_{(0)} = \{a_0, a_1, \ldots, a_{n-1}\}$, sử dụng phương pháp Fibonacci để lập công thức cho cả T bit liên tiếp, tính từ n bit của $S_{(0)}$.

Như vậy trạng thái trong của thanh ghi sau T bước sẽ là:

$S_{(5)} = \{a_5, a_6, \ldots, a_{22}, \mathbf{a_{23}, a_{24}, a_{25}, a_{26}, a_{27}}\}$.

Nếu biểu diễn dưới dạng ma trận:

$$A = \begin{bmatrix}
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
&&&&&&&&&\vdots&&&&&&&&&&\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
1&0&0&0&0&1&0&0&1&1&0&0&1&0&1&0&0&0&1&0&0&1&0\\
0&1&0&0&0&0&1&0&0&1&1&0&0&1&0&1&0&0&0&1&0&0&1\\
0&0&1&0&0&1&0&1&1&1&1&1&1&0&0&0&1&0&1&0&1&1&0\\
0&1&0&1&0&0&1&0&1&1&1&1&1&1&0&0&0&1&0&1&0&1&1\\
1&0&1&0&1&1&0&1&1&0&1&1&0&1&0&0&0&0&0&0&1&1&1
\end{bmatrix}$$

Với biểu diễn ma trận này, ta có thể tính trạng thái trong của thanh ghi sau T bước là: $S_{(T)} = A\ S_{(0)}^{T}$. Sử dụng lặp lại công thức trên, ta có thể sinh ra toàn bộ dãy phân rã mà không cần tính các giá trị trung gian.

Khi áp dụng giải pháp nêu trên, độ phức tạp tính toán không thay đổi giữa phương pháp truyền thống và phương pháp tính phân rã theo trực tiếp. Tuy nhiên trong phương pháp truyền thống, ta cần dịch chuyển thanh ghi dịch *T* lần. Với phương pháp mới, ta chỉ cần một lần dịch chuyển thanh ghi.

Phương pháp xây dựng dãy lồng ghép từ phân rã theo bước

Áp dụng giải pháp xây dựng nhanh dãy phân rã đã nêu, ta có thể tính được giá trị *m* bit đầu tiên, từ đó xây dựng nên dãy con đầu tiên của dãy lồng ghép. Đồng thời trong khi tính toán ta sẽ lưu lại *m* trạng thái trong của m-dãy ban đầu tương ứng. Từ *m* trạng thái trong này, áp dụng công thức gốc để sinh m-dãy ban đầu, ta sẽ lần lượt tính được *m* bit khởi đầu của các dãy con tiếp theo. Các bộ *m* bit khởi đầu này được sử dụng để sinh ra các dãy con tiếp theo. Như vậy khi sử dụng phương pháp phân rã theo bước, ta có thể tính trực tiếp mọi bộ *m* bit khởi đầu.

Phương pháp này cần sử dụng dung lượng bộ nhớ để lưu *m* trạng thái của m-dãy ban đầu, tương ứng với kích thước *m.n*. Thông thường khi cần sinh một phần của dãy lồng ghép với kích thước cho trước, ta chỉ cần tính các thứ tự lồng ghép tương ứng với số dãy con cần thiết để sinh ra đủ lượng đầu ra với kích thước được yêu cầu.

Kết luận chương 2

Trong chương này, tác giả giới thiệu kiến trúc dãy lồng ghép, các phương pháp xây dựng dãy lồng ghép trên trường *p*-phân.

Đồng thời phân tích phương pháp xây dựng dãy phi tuyến lồng ghép từ hai dãy lồng ghép. Trong phần cuối chương đã đề xuất một phương pháp mới để sinh dãy lồng ghép sử dụng kỹ thuật phân rã theo bước. Phương pháp này có thể áp dụng một trong thực tế để sinh một phần đầu tiên của dãy lồng ghép với kích thước cho trước.

CHƯƠNG 3: THUẬT TOÁN SINH DÃY PHI TUYẾN LỒNG GHÉP BẬC LỚN ỨNG DỤNG TRONG KỸ THUẬT MẬT MÃ

Trong chương này sẽ phân tích về một số yêu cầu của dãy giả ngẫu nhiên dùng trong mật mã và đề xuất thuật toán sinh dãy phi tuyến lồng ghép bậc lớn cùng với ứng dụng trong mật mã.

3.1. Độ phức tạp tuyến tính của dãy giả ngẫu nhiên

Độ phức tạp tuyến tính $L_n(s)$ được xác định là số k-bé nhất sao cho dãy n-phần tử $s_1, s_2,... s_n$ trùng với n-số hạng đầu tiên của một dãy ghi dịch phản hồi tuyến tính bậc k.

Độ phức tạp tuyến tính của một dãy bất kỳ có thể được xác định bằng thuật toán Belekamp-Massey. Thuật toán này cũng đưa ra giá trị của đa thức sinh cho dãy ghi dịch phản hồi tuyến tính.

Thuật toán tổng hợp LFSR (Berlekamp-Massey)

1) $1 \to C(D)$ $\quad 1 \to B(D)$ $\quad 1 \to x$

$0 \to L$ $\quad 1 \to b$ $\quad 0 \to N$

2) Nếu N = n, dừng thuật toán. Ngược lại tính

$$d = s_N + \sum_{i=1}^{L} c_i s_{N-i}.$$

3) Nếu d = 0, thì $x + 1 \to x$*, chuyển đến bước 6)*

4) Nếu $d \neq 0$ *và* $2L > N$*, thì*

$C(D) - d.b^{-1}.D^x.B(D) \to C(D)$

$x + 1 \rightarrow x$

và chuyển đến bước 6).

5) Nếu $d \neq 0$ *và* $2L \leq N$, *thì*

$C(D) \rightarrow T(D)$ *[lưu giữ tạm thời của* $C(D)$*]*

$C(D) - d.b^{-1}.D^{x}.B(D) \rightarrow C(D)$

$N + 1 - L \rightarrow L$

$T(D) \rightarrow B(D)$

$d \rightarrow b$

$1 \rightarrow x$

6) $N + 1 \rightarrow N$ *và quay về bước 2).*

Thuật toán Belekamp – Massey có thể được sử dụng để phân tích mã đối với một dãy giả ngẫu nhiên được tạo bởi một m-dãy duy nhất, do đó các bộ tạo dãy giả ngẫu nhiên dựa trên m-dãy luôn được cấu thành từ nhiều m-dãy kết hợp phi tuyến với nhau để tránh tấn công này.

3.2. Tính chất tương quan địa phương của m-dãy

Trong thực tế khi thực hiện mã hóa trong hệ mã dòng, ta chỉ dùng một đoạn của dãy giả ngẫu nhiên có độ dài M tương ứng với độ dài bản rõ để thực hiện mã hóa. Ta quan tâm tới sự tương quan giữa các đoạn khóa độ dài M được lấy ngẫu nhiên đều từ bộ sinh dãy giả ngẫu nhiên. Bài toán khi này trở thành: Giả thiết ζ lấy ngẫu nhiên đều trên không gian M. Tìm phân bố của biến ngẫu nhiên wt(ζ), trọng số của $\zeta \in M$.

Mômen phân bố trọng số của m-dãy

Công thức tính mômen bậc 1:

$$S^1 = \frac{1}{N}\sum_{n=0}^{N-1} S_n = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{i=0}^{M-1} b_{n+i} = \frac{1}{N}\sum_{i=0}^{M-1}\sum_{n=0}^{N-1} b_{n+i} = -\frac{M}{N}. \quad (3.1)$$

Công thức tính mômen bậc 2:

$$S^2 = M + (2/N)\{\ C_M^2 .(-1)\} = M.[1- (M-1)/N]. \qquad (3.2)$$

Công thức tính mômen bậc 3:

$$S^3 = -\frac{M^3}{N} + 3!\frac{N+1}{N}.B_3 \qquad (3.3)$$

Công thức tính mômen bậc 4:

$$S^4 = M(3M-2) - \frac{M(M-1)^2(M+2)}{N} + 4!\frac{N+1}{N}.B_4 \qquad (3.4)$$

Từ đó ta có thể xây dựng thuật toán tính giá trị mômen bậc 3 và mômen bậc 4 của phân bố trọng số (gọi là B_3 và B_4).

Để kiểm tra tính tương quan địa phương của một bộ sinh dãy giả ngẫu nhiên, ta sinh ra một lượng bit đầu ra để thực thi được thuật toán tính B_3 và B_4. So sánh kết quả mômen trọng số tính được với bảng phân bố trọng số của dãy ngẫu nhiên lý tưởng, ta có thể đưa ra kết luận bộ sinh số giả ngẫu nhiên đang xét thỏa mãn tính ngẫu nhiên địa phương ở mức độ nào.

3.3. Đề xuất thuật toán sinh dãy giả ngẫu nhiên phi tuyến lồng ghép với bậc lớn

Các yêu cầu của dãy lồng ghép áp dụng trong kỹ thuật mật mã.

Từ hai phương pháp phân tích dãy giả ngẫu nhiên đã đề cập trong phần 3.1 và 3.2, trong hầu hết các trường hợp ta đều nhận được kết luận là dãy được sinh ra bởi m-dãy thành phần có bậc m. Chỉ khi đoạn dữ liệu đem phân tích có sự tiếp nối giữa hai dãy con, khi này độ phức tạp tuyến tính được tăng lên, song không vượt quá độ phức tạp chung của dãy gốc có bậc n. Việc áp dụng bài toán tương quan địa phương cũng đưa ra kết quả tương tự.

Phân tích chi tiết về độ phức tạp tính toán của 4 phương pháp sinh dãy lồng ghép phi tuyến đã nêu trong chương 2, ta thấy các phương pháp đều cần tính toán trên đa thức có bậc tương đương tham số lồng ghép T. Với m, n đủ lớn ta có thể coi $T \approx p^{n-m}$ do đó $T \geq p^{n/2}$. Với giới hạn công nghệ hiện nay cả về năng lực tính toán

cũng như năng lực lưu trữ, các tính toán với số phép tính khoảng 10^{10} có thể thực hiện được trong thời gian chấp nhận được, song khi số phép tính cỡ 10^{15} trở nên thì bài toán trở thành không khả thi về tính toán.

Với yêu cầu về kỹ thuật mật mã đã nêu trong phần 1.2, ta cần có yêu cầu về bậc: $n \geq 128$ hay $T \geq p^{64}$. Với giá trị yêu cầu này của T thì cả 4 phương pháp đã nêu đều khó có thể áp dụng trong thực tế. Để có thể sinh dãy giả ngẫu nhiên phi tuyến lồng ghép thỏa mãn yêu cầu của kỹ thuật mật mã, tác giả luận án đã đề xuất thuật toán để tính toán giá trị đa thức d^T trên trường GF(p^n) trong trường hợp T rất lớn, sử dụng trong quá trình xây dựng tập thứ tự lồng ghép.

Thuật toán sinh dãy giả ngẫu nhiên phi tuyến lồng ghép với bậc lớn

Ta chú ý tới giá trị tham số T:

$$T = {}^{N}/_{L} \text{ với } L = p^m\text{-}1 \text{ và } N = p^n\text{-}1 \qquad (3.5)$$

Chú ý là $n = m*l$, vậy ta có thể viết:

$$N = p^{m*l}\text{-}1 \text{ hay } N = (p^m)^l - 1 \,. \qquad (3.6)$$

Để đơn giản, đặt $Q = p^m$, ta có thể viết

$$T = \frac{Q^l - 1}{Q - 1} = Q^{l\text{-}1} + Q^{l\text{-}2} + \ldots + Q + 1 \,. \qquad (3.7)$$

Nói một cách khác, nếu biểu diễn giá trị T theo cơ số Q là

(l số 1)

$$T = 111 \ldots 111_{\mathrm{Q}} \,. \qquad (3.8)$$

Nếu biểu diễn T theo cơ số p, biểu diễn của T cũng chỉ có l số 1, chen giữa các số 1 là m-1 số 0:

$$T = 100...00100...00100 \ldots 001_{\mathrm{p}} \qquad (3.9)$$

Ta sẽ tìm cách tính nhanh giá trị đa thức

$$U_T(d) = \frac{d^T}{g(d)} \,. \qquad (3.10)$$

Trong đó chia thành các bước tính $U_Q(d)$ và $U_{Q^k}(d)$.

Để tính $U_Q(d)$ ta sẽ tính lần lượt từng bước như sau. Trước hết

tính trực tiếp đa thức:

$$U_p(d) = \frac{d^p}{g(d)} \quad . \tag{3.11}$$

Chú ý rằng:

$$d^{p^k} = d^{p^{k-1} \times p} = \left(d^{p^{k-1}}\right)^p \quad . \tag{3.12}$$

Vì thế ta có:

$$U_{p^k}(d) = \left(U_{p^{k-1}}(d)\right)^p . \tag{3.13}$$

Ta có thể tính $U_{p^k}(d)$ theo công thức sau:

$$U_{p^k}(d) = \frac{\prod_{j=1}^{p} U_{p^{k-1}}(d)}{g(d)} . \tag{3.14}$$

$$U_Q(d) = U_{p^m}(d) . \tag{3.15}$$

Để tính $U_{Q^k}(d)$ từ $U_{Q^{k-1}}(d)$ Ta sẽ sử dụng phương pháp tương tự

Chú ý rằng $U_{p^0 Q^{k-1}}(d) = U_{Q^{k-1}}(d)$ nên:

$$U_{p^i Q^{k-1}}(d) = \frac{\prod_{j=1}^{p} U_{p^{i-1} Q^{k-1}}(d)}{g(d)} . \tag{3.16}$$

$$U_{Q^k}(d) = U_{p^m Q^{k-1}}(d) . \tag{3.17}$$

Cuối cùng, sử dụng các giá trị tính bởi (3.17) ta tính được đa thức (3.10). Như vậy bằng cách áp dụng các công thức truy hồi, ta có thể tính được giá trị $U_T(d)$ với tham số lồng ghép T có độ lớn bất kỳ.

Áp dụng phương pháp bình phương và nhân

Một số công thức trong thuật toán trên cần tính phép nhân tích lũy *p* giá trị giống nhau (dạng x^p). Khi ta có biểu diễn của *p* dưới dạng nhị phân thành tập các bit $\{p_i\}$ với i=0..r, sau đó sử dụng phương pháp bình phương và nhân ta có thể tính đa thức kết quả của công thức (3.40) bằng phương pháp sau:

Đặt $U^* = 1$

$V_{tmp} = U_{p^k}(d)$

Với i chạy từ 0 tới r, ta lần lượt tính

Nếu $p_i = 1$

$U^* = U^* \times V_{tmp}$

$V_{tmp} = \frac{(V_{tmp})^2}{g(d)}$

Sau r bước ta có $U_{p^k}(d) = \mathrm{U}^*$.

Bằng phương pháp tương tự ta tính được đa thức kết quả của công thức (3.42) như sau:

Đặt $U^* = 1$ *và* $V_{tmp} = U_{p^{i-1}Q^{k-1}}(d)$

Với i chạy từ 0 tới r, ta lần lượt tính

Nếu $p_i = 1$

$U^* = U^* \times V_{tmp}$

$V_{tmp} = \frac{(V_{tmp})^2}{g(d)}$

Sau r bước ta có $U_{p^iQ^{k-1}}(d) = \mathrm{U}^*$.

Thuật toán sinh dãy phi tuyến lồng ghép dựa trên m-dãy

Từ trạng thái khởi đầu bất kỳ của dãy, áp dụng thuật toán nêu trên ta tính được trạng thái của dãy sau T bước. Tiếp tục áp dụng phương pháp sinh dãy lồng ghép sử dụng phân rã theo bước ta tìm ra được m trạng thái của dãy gốc giá trị tại m vị trí đầu tiên trong cột 1 của ma trận lồng ghép, đồng nghĩa với việc ta có thể xây dựng được n cột đầu tiên của ma trận lồng ghép (do mỗi trạng thái có n bit). Nếu cần xây dựng các cột tiếp theo, ta tìm thứ tự lồng ghép tiếp theo bằng cách sử dụng từng trạng thái trong $S_{(kT)}$ để xác định các trạng thái trong $S_{(kT+1)}$ qua công thức sinh m-dãy, từ đó có được giá trị khởi đầu của cột n+1. Như vậy ta có thể xây dựng dãy lồng ghép có độ dài bất kỳ mà không cần tính trước toàn bộ bảng thứ tự lồng ghép.

Để sinh dãy phi tuyến lồng ghép, ta áp dụng các bước tương tự như với dãy lồng ghép đối với dãy đầu vào thứ nhất, song riêng

việc sinh ra giá trị các cột lại sử dụng dãy con từ dãy đầu vào thứ hai. Với dãy thứ hai ta không cần tính thứ tự lồng ghép.

Đánh giá độ phức tạp của thuật toán sinh dãy giả ngẫu nhiên phi tuyến lồng ghép với bậc lớn

Để tính được $U_Q(d)$ ta cần (m-1).(p-1) phép nhân đa thức trên trường GF(p^n) để tính công thức (3.17). Trong trường hợp tính bằng phương pháp bình phương và nhân, số phép nhân cần tính là (m-1).log$_2p$ (Giả sử thời gian tính phép bình phương và phép nhân đa thức là tương đương nhau).

Để tính được $U_{Q^k}(d)$ từ $U_{Q^{k-1}}(d)$ ta cần tính các phép nhân đa thức trên trường GF(p^n). Xét trường hợp sử dụng phương pháp bình phương và nhân thay cho (3.14) và (3.16) thì số phép nhân đa thức là:

$$v_{mulq} = (m\text{-}1).\log_2 p \qquad (3.18)$$

Để tính được $U_T(d)$ theo (3.10) ta cần sử dụng thêm (l-1) phép nhân đa thức. Tổng số phép nhân đa thức trên trường GF(p^n) cần tính là:

$$v_q = (l\text{-}1).(m\text{-}1).\log_2 p + (l\text{-}1). \qquad (3.19)$$

Chú ý là khi n $\to \infty$, m và l đều có cỡ tương đương n. Ta có thể coi $log_2 p$ là hằng số, xét phép nhân đa thức trên trường GF(p^n) có độ phức tạp tính toán là n thì độ phức tạp tính toán của v_q là cỡ O(n^3).

So sánh với phương pháp bình phương và nhân áp dụng trực tiếp cho giá trị số mũ T bằng một thuật toán tương tự như *Thuật toán 1*. Ta sẽ biểu diễn giá trị của T thành $log_2 T$ bit $\{t_i\}$ để áp dụng bình phương và nhân.

Số phép bình phương đa thức cần tính là

$$v_{mul2} = log_2 T = (n\text{-}m)*log_2 p = m.(l\text{-}1).log_2 p \qquad (3.20)$$

Số phép nhân cần tính sẽ có giá trị trung bình là v/2 (do phân

bố bit 0/1 trong biểu diễn nhị phân của T là đều nhau).

Vậy tổng số phép nhân đa thức cần tính là

$$v_2 = {}^3/_2 \cdot v_{mul2},$$

$$v_2 = {}^3/_2 \; m.(l\text{-}1).log_2 p. \qquad (3.21)$$

Phương pháp tính theo biểu diễn cơ số p có được lợi thế hơn phương pháp biểu diễn nhị phân bởi vì trong biểu diễn cơ số p của T có rất nhiều phần tử bằng 0 theo (3.31), trong khi biểu diễn nhị phân của T không có được lợi thế này.

Để so sánh cụ thể số bước tính toán giữa hai trường hợp tính toán với cơ số p và tính toán theo phương pháp bình phương và nhân trực tiếp (trên biểu diễn cơ số 2 của T) ta tính toán các giá

Bảng 3.2 Số bước tính toán tiền xử lý cho dãy lồng ghép

TT	p	n	m	v_{mulq}	$n_{bit1\text{-}q}$	v_q	T	v_{mul2}	$n_{bit1\text{-}2}$	v_2
1	2	24	8	17	3	**20**	65 793	17	3	**20**
2	3	24	8	32	3	**35**	43 053 283	26	12	**38**
3	7	24	8	48	3	**51**	33 232 936 334 403	45	24	**69**
4	7	28	14	42	2	**44**	678 223 072 850	40	23	**63**
5	13	12	3	36	4	**40**	10 609 328 380	34	17	**51**
6	13	18	9	36	2	**38**	10 604 499 374	34	17	**51**
7	29	12	3	45	4	**49**	14 507 740 823 580	44	19	**63**
8	31	12	3	45	4	**49**	26 440 509 694 144	45	12	**57**

trị cụ thể của vq và v2 trong một số trường hợp như trong bảng 3.2. Ta tính chính xác giá trị T, sau đó chuyển đổi sang nhị phân và đếm số bit để có $n_{bit1\text{-}2}$, do đó trong một số trường hợp số bit 1 không đúng là giá trị $½log_2T$.

So sánh dữ liệu trong bảng trên, ta thấy phương pháp tính toán với cơ số *p* có hiệu quả tốt hơn phương pháp tính toán bình phương và nhân trực tiếp.

Với cơ số $p = 2$, hai phương pháp có số bước giống nhau do cùng là tính toán trên cơ số 2. Với cơ số $p = 3$ là một giá trị rất nhỏ, phương pháp tính toán với cơ số *p* chỉ giúp tăng một phần nhỏ hiệu quả so với phương pháp tính toán bình phương và nhân trực tiếp. Trong trường hợp cơ số *p* có giá trị lớn, phương pháp tính toán trên cơ số *p* có hiệu quả tốt hơn hẳn so với phương pháp tính toán bình phương và nhân trực tiếp, cụ thể là số bước tính toán ít hơn khoảng 25%.

3.4 Đề xuất phương pháp sinh dãy giả ngẫu nhiên an toàn sử dụng dãy phi tuyến lồng ghép

Bộ tạo dãy luân phiên phi tuyến lồng ghép:

Sử dụng bộ tạo dãy luân phiên đã trình bày trong phần 1.3.3, trong đó ta chọn dãy thành phần thứ nhất là một dãy phi tuyến lồng ghép, trong khi dãy thành phần thứ 2 vẫn giữ nguyên là m-dãy và dãy điều khiển vẫn giữ nguyên dãy D'Brujin.

$K=\{ k_t\}_{t\geq 0}$ là dãy D' Bruijn bậc k; $U=\{u_t\}$ là m-dãy bậc L, $V=\{v_t\}$ là dãy phi tuyến lồng ghép bậc M, trong đó L và M nguyên tố cùng nhau.

Các tính chất của bộ tạo dãy luân phiên phi tuyến lồng ghép

So sánh với dãy luân phiên ban đầu, dãy luân phiên phi tuyến lồng ghép có chu kỳ tương đương, lực lượng bộ tạo dãy lớn hơn do có

thêm giá trị tham số lồng ghép T. Song xét về độ an toàn, cụ thể là tính tương quan thì dãy luân phiên phi tuyến lồng ghép có tính tương quan phụ thuộc vào dãy con của dãy lồng ghép, do đó các tính chất tương quan cũng bị suy giảm tương ứng với độ dài của dãy con so với độ dài dãy ban đầu.

Kết luận chương 3

Trong chương này đã đề xuất một thuật toán mới để sinh dãy phi tuyến lồng ghép với bậc lớn. Thuật toán này có độ phức tạp tính toán tiệm cận với $O(n^3)$ với n là bậc của đa thức sinh m-dãy. Bằng cách khai thác một đặc điểm của tham số của dãy lồng ghép, thuật toán này có lợi thế lớn hơn thuật toán bình phương và nhân thông thường. Tác giả cũng đề xuất bộ tạo dãy luân phiên phi tuyến lồng ghép là bằng việc ứng dụng dãy phi tuyến lồng ghép vào bộ tạo dãy luân phiên.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

A-Hai đóng góp khoa học của luận án

1) Đề xuất một giải pháp sinh dãy phi tuyến lồng ghép dựa trên kỹ thuật phân rã theo bước và kỹ thuật tính một phần thứ tự lồng ghép. Giải pháp này có thể ứng dụng trong cài đặt thực tế để sinh ra một đoạn có kích thước tùy ý của dãy phi tuyến lồng ghép.

2) Đề xuất một thuật toán hiệu quả để sinh dãy phi tuyến lồng ghép với bậc lớn. Độ phức tạp tính toán của thuật toán là cỡ $O(n^3)$ so với bậc n của đa thức sinh m-dãy. Bằng cách khai thác một đặc điểm của tham số của dãy lồng ghép, thuật toán này có lợi thế lớn hơn so với thuật toán bình phương và nhân thông thường.

B-Hướng phát triển tiếp theo của đề tài

Với những đóng góp khoa học nêu trên, luận án là cơ sở để tác giả có thể đề xuất một hệ mã dòng có thể ứng dụng trong kỹ thuật mật mã đáp ứng nhu cầu bảo mật thông tin trong Ban Cơ yếu. Việc đề xuất một thuật toán mật mã mới cần phải xem xét rất kỹ về tính an toàn của thuật toán trên nhiều khía cạnh trước khi có thể đưa vào sử dụng thực tế, cần có các nghiên cứu sâu về việc phân tích mã đối với dãy lồng ghép và phi tuyến lồng ghép, cũng như dãy luân phiên phi tuyến lồng ghép

Một công việc khác cần tiếp tục nghiên cứu là giải pháp để cài đặt hiệu quả các dãy trên $GF(p^n)$ với số p nguyên tố lớn ($p>2$) trên cả hai môi trường: phần mềm máy tính và các thiết bị xử lý trực tiếp bằng phần cứng. Ta cũng cần nghiên cứu về việc sử dụng hiệu quả dãy đầu ra trên $GF(p^n)$, có thể là một phương pháp chuyển đổi dữ liệu giữa hệ q-phân và hệ nhị phân.

DANH MỤC CÔNG TRÌNH KHOA HỌC ĐÃ CÔNG BỐ

[J1] Hieu Le Minh, **Truong Dang Van**, Binh Nguyen Thanh and Quynh Le Chi, "Design and Analysis of Ternary m-sequences with Interleaved Structure by d-Transform", Journal of Information Engineering and Applications, vol.5, no.8, pp.93-101, 2015.

[J2] **Truong Dang Van**, Quynh Le Chi, "Applying M-sequences Decimation to Generate Interleaved Sequence", Journal of Science and Technology on Information security, No 2.CS (14) 2021, pp. 85-88

[J3] **Đặng Vân Trường, "**Một phương pháp hiệu quả để sinh dãy giả ngẫu nhiên kiểu lồng ghép phi tuyến với bậc lớn**",** Tạp chí Khoa học Công nghệ Thông tin và Truyền thông (Journal of Science and Technology on Information and Communications), 2022