(12) BẢN MÔ TẢ SÁNG CHẾ THUỘC BẰNG ĐỘC QUYỀN SÁNG CHẾ

(19) **Cộng hòa xã hội chủ nghĩa Việt Nam (VN)**
CỤC SỞ HỮU TRÍ TUỆ

(11) **1-0052469**

(51)[2022.01] **G06F 21/00**

(13) **B**

(21) 1-2023-05066

(22) 28/07/2023

(45) 27/10/2025 451

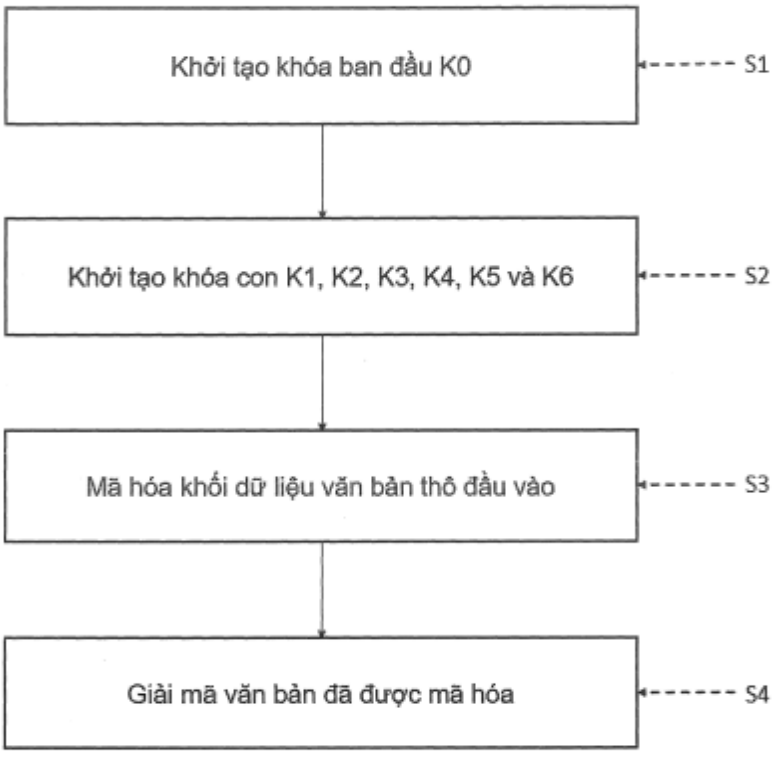
(43) 27/11/2023 428A

(76) Phùng Văn Hà (VN)
306, KNƠ, Bắc Linh Đàm mở rộng, Phường Hoàng Liệt, Quận Hoàng Mai, thành phố Hà Nội

(54) PHƯƠNG PHÁP MÃ HÓA VÀ GIẢI MÃ KHỐI DỮ LIỆU CÓ KÍCH THƯỚC BIẾN ĐỔI THEO ĐỘ DÀI KHÓA

(21) 1-2023-05066

(57) Sáng chế đề xuất phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa bao gồm: khởi tạo khóa ban đầu K0, tạo các khóa con một chiều để sử dụng cho mã hóa và giải mã dữ liệu; mã hoá dữ liệu theo các khoá con và hàm S-box; giải mã dữ liệu sử dụng khoá con và hàm I-Sbox.



Hình 1

Lĩnh vực kỹ thuật được đề cập

Sáng chế đề cập đến phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa. Cụ thể, sáng chế đề cập đến phương pháp giải mã và mã hóa khối dữ liệu sử dụng kiến trúc mạng mật mã xử lý kích thước khối dữ liệu biến đổi theo độ dài khóa.

Tình trạng kỹ thuật của sáng chế

Ngành mật mã học nói chung và các kỹ thuật mật mã nói riêng có lịch sử ra đời và phát triển hàng nghìn năm nay, được tạo ra do nhu cầu bảo mật thông tin trong quá trình phát triển của loài người, đặc biệt trong các thời kỳ chiến tranh. Ngành này được chia ra thành các thời kỳ: mật mã học cổ điển xuất hiện vào các thời kỳ Ai Cập cổ đại, Hy Lạp cổ đại, và trong các nền văn minh Ấn-Hằng; mật mã học trung cổ xuất hiện vào các thời kỳ La Mã, thời kỳ Phục Hưng tại Châu Âu; mật mã học giai đoạn thế chiến I, II; và mật mã học hiện đại với sự ra đời của lý thuyết mật mã của Claude Shannon năm 1949, đặt nền móng cho các kỹ thuật mật mã toán học đang được sử dụng rộng rãi cho tới ngày nay. Tuy nhiên, do sự phát triển như vũ bão của công nghệ điện toán, và nhu cầu trao đổi thông tin từ xa thông qua các hệ thông mạng truyền dẫn dữ liệu, cần phải chuẩn hóa một kỹ thuật mật mã chung, dễ thực hiện, và bảo mật cao để áp dụng cho nhiều nền tảng điện toán khác nhau. Đáp ứng yêu cầu này, vào năm 1977, tiêu chuẩn mã hóa dữ liệu DES, được phát hành bởi Cục Tiêu chuẩn Quốc gia Mỹ - NBS (nay là Viện Công nghệ và Tiêu chuẩn Quốc gia - NIST) theo đề xuất của IBM, đã được chấp thuận áp dụng.

Cũng vậy, đi cùng với sự phát triển và chuẩn hóa kỹ thuật mật mã, các kỹ thuật phân tích mật mã cũng phát triển theo, chủ yếu dưới các hình thức tấn công phá khóa, điển hình là: tấn công kiểu thác lũ (*Brute Force Attack*) tấn công sử dụng phân tích mã tuyến tính (*Linear Analysis Attack*), tấn công sử dụng phân tích vi sai (*Differential Anlysis Attack*), tấn công Boomerang (*Boomerang Attack*), tấn công sử dụng các trường đại số (*Algebraic Attack*), tấn công sử dụng phân tích công suất vi sai (*Differential Power Analysis Attack*), tấn công sử dụng phân tích thời gian (*Timing Analysis Attack*), tấn công thử nghiệm không gian con và vi sai rút gọn (*Truncated Differential and Subspace Trail Attack*), tấn công khối lập phương (*Cube or Cube-like Attack*), tấn công MITM, và nhiều hình thức tấn công khác. Sự phát triển của các hình thức tấn công phá khóa dẫn đến nhiều kỹ thuật mã hóa đã bị bẻ gãy, bao gồm cả DES. Do đó, vào năm 1999, NIST phát động một cuộc thi tìm kiếm một giải thuật mật mã mới để chuẩn hóa thay thế cho DES.

Cuộc thi đã thu hút nhiều nhiều cá nhân, tổ chức đăng ký các giải thuật của họ tham gia, có thể kể đến các giải thuật mật mã khối như: RIJNDAEL (do hai nhà nghiên cứu mật mã người Bỉ là Joan Daemen and Vincent Rijmen đề xuất), MARS (do IBM đề xuất), RC6 (do nhóm nghiên cứu gồm Ron Rivest, Matt Robshaw, Ray Sidney, Yiqun Lisa Yin đề xuất), SERPENT (do nhóm nghiên cứu gồm Ross Anderson, Eli Biham, Lars Knudsen đề xuất), TWOFISH (do nhà nghiên cứu người Mỹ Bruce Schneier đề xuất), CAST-256 (do nhóm nghiên cứu gồm Carlisle Adams, Stafford Tavares, Howard Heys, Michael Wiener đề xuất), và nhiều giải thuật khác như DFC, E2, LOKI-97, MAGENTA, SAFER+,…

Với mục tiêu giải thuật đáp ứng cần có độ bảo mật cao, chống lại hiệu quả các kỹ thuật tấn công phá khóa, đồng thời dễ thực hiện trên cả phần cứng và phần mềm, sau vòng thi cuối cùng, giải thuật RIJNDAEL đã được NIST chọn lựa trở thành tiêu chuẩn mã hóa mới tiên tiến AES vào năm 2000, và chính thức thay thế DES vào năm 2001. Giải thuật AES tập trung trên các độ dài khóa 128-bit, 192-bit, và 256-bit, với kích thước khối dữ liệu 128-bit, thuộc thể loại mật mã khối, khóa đối xứng. Hiện nay, các giải thuật mật mã mới vẫn đang tiếp tục được nghiên cứu và công bố, bởi vì các giải thuật đã và đang được sử dụng, bao gồm cả AES vẫn tồn tại những hạn chế nhất định. Cụ thể là:

Điểm hạn chế (1): Trên lý thuyết, mọi giải thuật mật mã đều bị bẻ gãy với tấn công kiểu thác lũ (Brute Force Attack). Trong kiểu tấn công này, mỗi khóa trong tổ hợp 2^n khóa sẽ được thử, vấn đề ở đây là thời gian thực hiện có thể lên đến hàng trăm năm với độ dài khóa n = 256-bit và với công suất tính toán của hệ thống máy tính tại thời điểm AES được công bố. Tuy nhiên, với công suất tính toán của thế hệ máy tính lượng tử hiện nay, thời gian tấn công kiểu thác lũ đã giảm rõ rệt. Phương pháp duy nhất chống lại tấn công kiểu thác lũ là tăng độ dài khóa càng lớn càng tốt. Tuy nhiên, việc tăng độ dài khóa đối với các giải thuật mật mã khối đang sử dụng hiện nay đối mặt với hai trở ngại: thời gian mã hóa/giải mã tăng; dung lượng bộ nhớ yêu cầu tăng.

Điểm hạn chế (2): Trong hai kiến trúc mạng mật mã SPN (Substitution-Permutation Network) và Feistel, thành phần quan trọng đảm bảo tính phi tuyến cao cho dữ liệu mã hóa là hộp thay thế S-box. Nhiều phương pháp tấn công ở trên (điển hình là *Differential Anlysis Attack* và *Linear Anlysis Attack*) tập trung vào khai các thuộc tính bảo mật của S-box để khôi phục ngược khóa bảo mật. Do các giải thuật mật mã ở trên sử dụng các S-box chọn trước cố định để tính trước các bảng tìm kiếm thay thế khi thực hiện mã hóa/giải mã, quy trình tính các bảng tìm kiếm này phức tạp và yêu cầu nhiều thời gian thực hiện, do đó trong quá trình áp dụng, việc thay thế S-box khác có thuộc tính bảo mật tương đương (nhằm tăng độ

khó chống lại các phương pháp tấn công bẻ khóa vì phải thử các S-box khác nhau, về lý thuyết sẽ có một tập gồm N chỉnh hợp chập 256 của 256 đối với loại S-box 8-bit) sẽ không được thực hiện dễ dàng. Như với AES tiêu chuẩn, việc thay thế S-box thì dễ dàng nhưng lại gặp hạn chế về thời gian mã hóa/giải mã tăng lên, còn AES thực hiện nhanh như RIJNDAEL thì lại gặp khó khăn thì thay thế S-box do phải tính toán lại 10 bảng tìm kiếm có kích thước mỗi bảng là 256 x 32-bit và yêu cầu kiến thức chuyên môn sâu về toán học trường nhị phân GF (2).

Điểm hạn chế (3): Như đã đề cập với các giải thuật mật mã đã và đang được sử dụng ở trên, để giảm thời gian mã hóa, ngoài việc sử dụng các bảng tìm kiếm thay thế đã tính toán trước, các giải thuật này còn sử dụng các phép toán trên cơ sở số 32-bit và các hàm Intrinsic (thường chỉ được hỗ trợ trên ngôn ngữ lập trình C). Các tính toán 32-bit tận dụng được các ưu điểm của các bộ xử lý 32/64-bit đa nhân, tuy nhiên sẽ gặp vấn đề về tăng kích thước mã lệnh và tốc độ xử lý chậm đi đáng kể trên các hệ vi xử lý 8/16/32-bit đơn nhân. Đây là vấn đề lớn khi hiện nay các thiết bị IoT thường phải xử dụng các chip 8/16/32-bit đơn nhân để giảm giá thành.

Điểm hạn chế (4): Để giải quyết vấn đề tốc độ và bộ nhớ, một số giải thuật mật mã đã được cứng hóa nghĩa là sử dụng một phần cứng chuyên biệt để thực hiện giải thuật. Việc cứng hóa yêu cầu phải được thực hiện trước từ khâu thiết kế ASIC, dẫn đến chi phí sản xuất chip tăng lên, và không mềm dẻo khi cần thay đổi hoặc cập nhật các thuộc tính bảo mật của giải thuật. Ngoài ra, các chip IoT giá rẻ hầu như không có các phần cứng chuyên biệt này. Để cứng hóa, chỉ các giải thuật được chấp nhận bởi các tổ chức tiêu chuẩn quốc tế như NIST mới được sử dụng. Vì thế, hiện nay, hầu như chỉ giải thuật AES được chọn để thực hiện cứng hóa. Tuy nhiên, việc cứng hóa chỉ giải quyết được vấn đề tốc độ, chứ không cải tiến được khả năng bảo mật và tính mềm dẻo khi ứng dụng như khả năng thay thế S-box tương đương. Ví dụ, tấn công vi sai Bicliques có thể bẻ gãy khóa 128/192/256-bit khi biết khóa con (subkey) tại vòng 8 đối với AES-128, và vòng 9 đối với AES-192/AES-256 dù giải thuật được thực hiện bằng phần cứng hay phần mềm.

Điểm hạn chế (5): Các giải thuật mật mã như AES tiêu chuẩn (AES), AES nhanh (RIJNDAEL), MARS, RC6, SERPENT, TWOFISH, và nhiều giải thuật khác hoạt động trên các khối dữ liệu có độ dài cố định 128-bit với số lượng các độ dài khóa nhỏ (hầu như chỉ là các độ dài khóa 128-bit, 192-bit, và 256-bit), vì thế các cuộc tấn công luôn biết trước được độ dài khối và độ dài khóa để áp dụng số lượng cặp dữ liệu vào/ra có sai khác cố định từ đó tìm ra thuộc tính có tính nhất quán với xác suất cao nhất và nội suy ra khóa, điển hình là các mô hình tấn công

như Chosen plaintext attack, Chosen ciphertext attack, Known plaintext attack. Để chống lại hoặc làm khó khăn thêm các cho các cuộc tấn công dạng này, kích thước khối dữ liệu và độ dài khóa cần được thay đổi linh hoạt và không dễ để dự đoán trước, khi những kẻ tấn công thu giữ được các văn bản mã hóa (ciphertext). Điều này không dễ dàng và đơn giản để thực hiện một cách nhanh chóng (on-the-fly) với các mô hình mạng SPN và Feistel hiện hành, vì sẽ phải tính toán lại S-box, tính toán lại các bảng tìm kiếm, tính toán lại số vòng thực hiện, tính toán lại các cơ chế thực hiện mã hóa/giải mã, và sau cùng là phải kiểm tra các thuộc tính ngẫu nhiên của dữ liệu mã hóa để đảm bảo giải thuật đạt được các yêu cầu về bảo mật. Đặc biệt khi kích thước khối dữ liệu xử lý và độ dài khóa càng lớn thì các công việc này lại càng khó khăn, phức tạp và tốn nhiều thời gian thực hiện.

Các giải thuật mật mã đã biết luôn tập trung tối đa vào việc tạo ra các quan hệ phi tuyến cao giữa các biến vào và biến ra, sử dụng các phép nhân, chia đa thức, và logarit trong trường nhị phân GF(2) dẫn đến sự phức tạp về tập lệnh thực hiện và yêu cầu bộ nhớ lớn.

Từ những hạn chế trên, sáng chế đề xuất phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa khắc phục các nhược điểm nêu trên.

Bản chất kỹ thuật của sáng chế

Sáng chế đề xuất phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa giải quyết các hạn chế nêu ra ở trên.

Cụ thể, sáng chế đề xuất phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa bao gồm các bước:

(S1) khởi tạo khóa ban đầu K0;

(S2) khởi tạo sáu tập khóa con một chiều K1, K2, K3, K4, K5 và K6 từ khoá ban đầu K0;

(S3) mã hóa khối dữ liệu văn bản thô đầu vào bằng cách sử dụng lần lượt ba khoá con K1-K3 và K4-K6 thu được từ (S2); và

(S4) giải mã văn bản đã được mã hóa thu được ở (S3) bằng cách sử dụng lần lượt ba khoá con K6-K4 và K3-K1 thu được từ (S2).

Phương pháp theo sáng chế sử dụng kiến trúc mạng mật mã mới gọi là mạng phương trình phi tuyến đa ẩn được cấu tạo với 4 hệ phương trình phi tuyến thuận cho bước mã hóa (S3) và 4 hệ phương trình phi tuyến ngược cho giải mã (S4), trong đó các hệ phương trình phi tuyến này được sắp xếp theo cơ chế chuỗi nối tiếp, nghĩa là đầu ra của hệ phương trình này là dữ liệu cho việc khởi tạo đầu vào của hệ phương trình kế tiếp. Bên cạnh đó, dữ liệu cần mã hoá được chia thành các

khối dữ liệu có độ dài bằng độ dài khóa K0 có độ dài nằm trong khoảng từ 128-bit tới 8192-bit với mỗi bước thay đổi là 16-bit, theo cùng một cơ chế thống nhất với các vòng lặp các mã lệnh xử lý, không sử dụng các bảng tìm kiếm thay thế 32-bit, ngoài các Sbox 8-bit (nghĩa là độ dài của Sbox đã quy định là 2^8 = 256), phương pháp theo sáng chế có thể sử dụng tới 3 Sbox 8-bit khác nhau. Cơ chế thống nhất này cho phép đạt kết quả tương đồng về tốc độ thực hiện, về hiệu quả nhầm lẫn và khuếch tán, và phân bố ngẫu nhiên thống kê của dữ liệu mã hóa theo các phép kiểm tra đánh giá của NIST cho mọi kích thước độ dài khóa.

Phương pháp theo sáng chế có độ tuỳ chọn độ dài khoá cao hơn, thay vì chỉ được chọn các mốc độ dài khoá cụ thể như 128-bit, 192 và 256; người dùng có thể lựa chọn các độ dài khóa khác mà không ảnh hưởng tới chất lượng của văn bản được mã hoá và giải mã. Kiến trúc cân bằng và đối xứng của mạng phương trình phi tuyến đa ẩn cho phép thực hiện số tập lệnh mã máy (cũng như tập lệnh ngôn ngữ lập trình bậc cao) đơn giản, ổn định, không biến đổi, giống nhau cho mọi độ dài khóa K0, không phải sử dụng các phép nhân/chia đa thức, các hàm logarit thông qua các phép quay bit dữ liệu và bảng tìm kiếm. Do tính phi tuyến đa ẩn, các bước mã hóa/giải mã chỉ sử dụng hai tác vụ chính là XOR-bit và thay thế S-box. Đặc trưng này khiến độ khó của phương pháp tăng lên rất nhiều so với các tấn công kiểu thác lũ (Brute Force Attack) do việc tăng k-bit khóa sẽ làm tăng thêm 2^k tổ hợp khóa cần thử.

Phương pháp theo sáng chế không phụ thuộc vào hộp thay thế S-box đã tính toán trước. Kiến trúc cân bằng và đối xứng của mạng phương trình phi tuyến đa ẩn cũng cho phép thay đổi S-box có thuộc tính bảo mật tương đương hoặc cao hơn với S-box mặc định của giải thuật mà không ảnh hưởng tới chất lượng giải thuật như bộ nhớ, tốc độ, hiệu quả nhầm lẫn và khuếch tán, và phân bố thống kê ngẫu nhiên.

Phương pháp theo sáng chế truyền sự thay đổi của một biến vào tới mọi biến ra. Kiến trúc mạng phi tuyến đa ẩn cho phép 2 đặc trưng phi tuyến quan trọng được thực hiện đồng thời: (1) một biến ra phụ thuộc phi tuyến vào tất cả các biến vào; (2) một biến vào ảnh hưởng phi tuyến tới mọi biến ra.

Phương pháp theo sáng chế dễ dàng triển khai trên mọi nền tảng bộ vi xử lý mà không ảnh hưởng tới các hiệu quả bảo mật, tạo các khóa con (subkey) một chiều có mức độ độc lập tuyến tính lẫn nhau cao, tối thiểu vòng lặp kiến trúc mạng và tối đa số lượng khóa con có thể sử dụng trên mỗi vòng lặp. Cụ thể hơn, cơ chế mã hóa và giải mã sử dụng 2 vòng lặp kiến trúc mạng và 3 khóa con riêng biệt cho mỗi vòng lặp.

Mô tả vắn tắt các hình vẽ

Hình 1 là lưu đồ thể hiện phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa theo một phương án của sáng chế;

Hình 2 là lưu đồ thể hiện bước (S2) tạo các khóa con một chiều sử dụng cho mã hóa và giải mã dữ liệu theo một phương án của sáng chế;

Hình 3a là sơ đồ khối mô tả bước (S3) mã hóa dữ liệu đầu vào theo một phương án của sáng chế;

Hình 3b là lưu đồ thể hiện bước (S3) mã hóa dữ liệu theo một phương án của sáng chế;

Hình 4a là sơ đồ khối mô tả bước (S4) giải mã dữ liệu theo một phương án của sáng chế;

Hình 4b là lưu đồ thể hiện bước (S4) giải mã dữ liệu theo một phương án của sáng chế.

Mô tả chi tiết sáng chế

Dưới đây, các ưu điểm, hiệu quả và bản chất của sáng chế có thể được hiểu rõ hơn thông qua việc mô tả chi tiết các phương án ưu tiên thực hiện có dựa vào các hình vẽ kèm theo. Trên các hình vẽ, các số chỉ dẫn giống nhau được dự định để biểu thị các thành phần hoặc chi tiết giống nhau hoặc tương đương và được sử dụng thống nhất trong toàn bộ mô tả, do vậy trên một số các hình vẽ hoặc một số phần trên hình vẽ có thể không xuất hiện một hoặc một số các số chỉ dẫn nhằm mục đích làm cho hình vẽ trở nên đơn giản và thuận tiện trong việc thể hiện các thành phần cấu tạo hoặc nguyên lý hoạt động khác nhau của sáng chế, trong trường hợp như vậy, mối tương quan giữa các thành phần hoặc chi tiết cụ thể với các số chỉ dẫn biểu thị nó có thể được minh họa rõ khi tham chiếu tới các hình vẽ khác hoặc các phần khác trên hình vẽ. Bên cạnh đó, các thành phần và chi tiết được thể hiện trên hình vẽ là không theo kích thước và hình dạng thực tế, một số thành phần hoặc chi tiết sẽ được phóng đại lên và có thể được biểu thị bởi các khối giản lược nhằm mục đích minh họa và thuận tiện cho việc mô tả. Vì vậy, cần hiểu rằng các phương án được mô tả trong phần mô tả chỉ với mục đích làm ví dụ giúp cho việc hiểu rõ hơn về bản chất và các ưu điểm của sáng chế, mà không giới hạn phạm vi của sáng chế theo các phương án được mô tả này.

Mã hoá khối (Block Cipher) thực hiện tính toán trên từng khối bit có kích thước cố định. Vì vậy khi đầu vào bản cần mã hoá (plaintext) có độ dài không là bội số của khối, cần thực hiện thao tác đệm (padding) sao cho số bit của đầu vào phải là bội số của khối.

Trong lĩnh vực mã hoá, SBox (Substitution-box Hộp thay thế) là thành phần cơ bản của các thuật toán khóa đối xứng để thực hiện thay thế. Trong mã hoá khối,

SBox thường được sử dụng để che đi mối quan hệ giữa khóa và bản mã (dữ liệu sau khi mã hoá-ciphertext) - thuộc tính nhầm lẫn của Shannon. Trong nhiều trường hợp, các SBox được lựa chọn cẩn thận để chống các giải thuật. Nói chung, SBox lấy *a* bit đầu vào và biến thành *b* bit đầu ra. Theo sáng chế, *a* bằng *b,* Sbox là hàm thay thế để biến tổ hợp tuyến tính hoặc phi tuyến của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra có *m* phần tử.

XOR là một phép toán logic căn bản trong lĩnh vực điện tử và là phép toán mã lệnh căn bản trong các ngôn ngữ lập trình. Theo sáng chế, thay vì phải sử dụng các phép toán nhân, chia, logarit phức tạp để tạo tập dữ liệu phi tuyến đầu ra, chỉ các phép toán XOR và thay thế S-box được sử dụng.

Khóa con một chiều K1-K6 khi được sử dụng trong sáng chế được hiểu là cần biết dữ liệu đầu vào tạo khóa mới tìm ra được khóa con, trong khi nếu biết được khóa con thì không nội suy được dữ liệu đầu vào. Cụ thể là, biết khóa khởi tạo K0 mới tạo ra được K1, biết K1 mới tạo ra được K2, biết K2 mới tạo ra được K3, biết K3 mới tạo ra được K4, biết K4 mới tạo ra được K5, biết K5 mới tạo ra được K6. Quá trình nêu trên không thể đảo ngược, ví dụ biết K6 thì không thể nội suy ra K5.

Trên Hình 1 thể hiện phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa theo một phương án của sáng chế.

Như thể hiện trên Hình 1, phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa bao gồm các bước:

(S1) khởi tạo khóa ban đầu K0;

(S2) khởi tạo sáu tập khóa con K1, K2, K3, K4, K5 và K6 có số lượng phần tử bằng nhau và bằng m;

(S3) mã hóa khối dữ liệu văn bản thô đầu vào;

(S4) giải mã văn bản đã được mã hóa.

Ở bước S1, khởi tạo khóa ban đầu K0, K0 có độ dài *m* do người sử dụng thiết lập, trong đó m là số chẵn byte có độ dài khóa từ 16-byte đến 256-byte, tương đương 128-bit đến 8192-bit;

Ở bước S2, như thể hiện ở Hình 2, tạo các khóa con một chiều để sử dụng cho mã hóa và giải mã dữ liệu bao gồm:

khởi tạo tập khóa con K1, bao gồm:

S211 tạo ba khóa phụ GS1, sử dụng khóa K0 để tạo ba khóa phụ Ks11, Ks12 và Ks13 có độ dài bằng của K0, trong đó:

khởi tạo: $\text{idx} = \text{Sbox}(K0_0 \hat{} K0_1)$ (1.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks11}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j \text{\^{}} \text{K0}_{j1})}{256} \right\rfloor \\ \text{Ks12}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks12}_i \text{\^{}} \text{K0}_{j1})}{256} \right\rfloor \\ \text{Ks13}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{với } 0 \le i \le m-1 \quad (1.2)$$

S211' sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, bao gồm bốn tập dữ liệu đầu vào K0, Ks11, Ks12, Ks13, mỗi tập có *m* phần tử, để tạo ra tập khóa con K1 có *m* phần tử;

khởi tạo tập khóa con K2, bao gồm:

S212 tạo ba khóa phụ GS2, sử dụng khóa K0 và khóa con K1 để tạo ra 3 khóa phụ khác cùng độ dài Ks21, Ks22, Ks23, trong đó:

khởi tạo: $\text{idx} = \text{Sbox}(\text{K0}_0 \text{\^{}} \text{K1}_1)$ (2.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks21}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j \text{\^{}} \text{K1}_{j1})}{256} \right\rfloor \\ \text{Ks22}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks22}_i \text{\^{}} \text{K1}_{j1})}{256} \right\rfloor \\ \text{Ks23}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{với } 0 \le i \le m-1 \quad (2.2)$$

S212' sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, với 4 tập dữ liệu đầu vào K1, Ks21, Ks22, Ks23, mỗi tập có *m* phần tử, để tạo ra tập khóa con K2 có *m* phần tử;

khởi tạo tập khóa con K3, bao gồm:

S213 tạo 3 khóa phụ GS3 sử dụng khóa K0 và khóa con K2 để tạo ra ba khóa phụ khác cùng độ dài Ks31, Ks32, Ks33, trong đó:

khởi tạo: idx = $\mathrm{Sbox}(\mathrm{K0}_0 \hat{} \mathrm{K2}_1)$ (3.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \mathrm{Sbox}(\mathrm{idx}) \\ j = \left\lfloor \frac{i+(m \gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \mathrm{Ks31_i} = \mathrm{temp} \\ idx = \left\lfloor \frac{idx+1+(\mathrm{K0_j} \hat{} \mathrm{K2_{j1}})}{256} \right\rfloor \\ \mathrm{Ks32_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\mathrm{Ks32_i} \hat{} \mathrm{K2_{j1}})}{256} \right\rfloor \\ \mathrm{Ks33_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \mathrm{i} \le \mathrm{m}-1 \quad (3.2)$$

S213' sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn, với 4 tập dữ liệu đầu vào K2, Ks31, Ks32, Ks33, mỗi tập có *m* phần tử, để tạo ra tập khóa con K3 có *m* phần tử;

khởi tạo tập khóa con K4, bao gồm:

S214 tạo ba khóa phụ GS4 sử dụng K0 và khóa con K3 để tạo ra ba khóa phụ khác cùng độ dài Ks41, Ks42, Ks43, trong đó:

khởi tạo: idx = $\mathrm{Sbox}(\mathrm{K0}_0 \hat{} \mathrm{K3}_1)$ (4.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \mathrm{Sbox}(\mathrm{idx}) \\ j = \left\lfloor \frac{i+(m \gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \mathrm{Ks41_i} = \mathrm{temp} \\ idx = \left\lfloor \frac{idx+1+(\mathrm{K0_j} \hat{} \mathrm{K3_{j1}})}{256} \right\rfloor \\ \mathrm{Ks42_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\mathrm{Ks42_i} \hat{} \mathrm{K3_{j1}})}{256} \right\rfloor \\ \mathrm{Ks43_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \mathrm{i} \le \mathrm{m}-1 \quad (4.2)$$

S214' sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, với 4 tập dữ liệu đầu vào K3, Ks41, Ks42, Ks43, mỗi tập có *m* phần tử, để tạo ra tập khóa con K4 có *m* phần tử;

khởi tạo tập khóa con K5, bao gồm:

S215 tạo ba khóa phụ GS5 sử dụng K0 và khóa con K4 để tạo ra ba khóa phụ khác cùng độ dài Ks51, Ks52, Ks53, trong đó:

khởi tạo: idx = Sbox($K0_0$^$K4_1$) (5.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m \gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks51}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j \text{^K4}_{j1})}{256} \right\rfloor \\ \text{Ks52}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks52}_i \text{^K4}_{j1})}{256} \right\rfloor \\ \text{Ks53}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{với } 0 \le i \le m-1 \quad (5.2)$$

S215' sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, với bốn tập dữ liệu đầu vào K4, Ks51, Ks52, Ks53, mỗi tập có *m* phần tử, để tạo ra tập khóa con K5 có *m* phần tử;

khởi tạo tập khóa con K6, bao gồm:

S216 tạo ba khóa phụ GS6 sử dụng K0 và khóa con K5 để tạo ra ba khóa phụ khác cùng độ dài Ks61, Ks62, Ks63, trong đó:

khởi tạo: idx = Sbox($K0_0$^$K5_1$) (6.1)

lặp lại *m* lần các bước sau, trong đó *m* là độ dài khóa K0 tính theo byte:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m \gg 1)}{m} \right\rfloor; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks61}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j \text{^K5}_{j1})}{256} \right\rfloor \\ \text{Ks62}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks62}_i \text{^K5}_{j1})}{256} \right\rfloor \\ \text{Ks63}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{với } 0 \le i \le m-1 \quad (6.2)$$

(S216') tạo ra tập khóa con K6 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận với bốn tập dữ liệu đầu vào K5, Ks61, Ks62, Ks63, mỗi tập có *m* phần tử;

Ở bước S3, mã hóa khối dữ liệu văn bản thô đầu vào, chia văn bản thô đầu vào thành các khối dữ liệu X có độ dài bằng độ dài khóa K0, trong đó nếu văn bản

thô này không chia hết cho *m*, khối dữ liệu X cho phần không chia hết này sẽ bao gồm các byte dữ liệu gốc và các byte đệm 0 hoặc 1 để khối dữ liệu này có độ dài *m*.

Như thể hiện ở Hình 3a mã hóa khối dữ liệu X, trong đó bao gồm các bước:

S301 sử dụng khối dữ liệu X và ba tập khóa con K1, K2, K3,

S302 sử dụng kết quả đầu ra của S301 và ba tập khóa con K4, K5, K6,

kết quả đầu ra của S302 là khối dữ liệu đã được mã hóa Y, có độ dài bằng khối dữ liệu X, tập hợp các khối dữ liệu Y để được một văn bản mã hóa của văn bản thô đầu vào, trong đó:

S301 sử dụng khối dữ liệu X và ba tập khóa con K1, K2, K3, bao gồm bảy bước thực hiện như sau:

Như thể hiện trên Hình 3b, S301 có bảy bước thực hiện từ E1S1 đến E1S7, khối dữ liệu X và ba tập khóa con K1, K2 và K3 được định dạng thành các tập hợp 8-bit (1 byte), gọi *n* là độ dài theo bit, *m* là độ dài theo byte thì *m=n/8*.

S311 chuyển dạng phi tuyến thuận E1S1 thực hiện *m* hàm chuyển dạng phi tuyến của 2 tập biến số X và K1, mỗi tập có *m* phần tử, trong đó X11 được tính như sau:

$$X11 = Sbox(X + K1) = \{Sbox(X_i + K1_i)\} \text{ với } 1 \le i \le m \quad (3.1)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X11 có *m* phần tử.

S312, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S2 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X11 để tạo ra một kết quả của tập X12, X12 gồm *m* phần tử kết quả, trong đó X12 được tính như sau:

$$X12 = F(X11) = \{F_i(X11_i)\} \text{ với } 1 \le i \le m \quad (3.2)$$

Biểu diễn toán học của biểu thức (3.2) như sau:

$$\begin{cases} X12_1 = A_{1,1} * X11_1 + A_{1,2} * X11_2 + \cdots + A_{1,m} * X11_m \\ X12_2 = A_{2,1} * X11_1 + A_{2,2} * X11_2 + \cdots + A_{2,m} * X11_m \\ \cdots \\ X12_m = A_{m,1} * X11_1 + A_{m,2} * X11_2 + \cdots + A_{m,m} * X11_m \end{cases} \quad (3.3)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số A là một ma trận vuông *m* hàng và *m* cột $A_{m,m}$ với các phần tử nhị phân, có các đặc điểm sau: chỉ phương trình đầu tiên phụ thuộc vào *m* phần tử đầu vào, các phương trình còn lại phụ thuộc vào m-1 phần tử đầu vào, ma trận A luôn tồn tại một hàng (hàng đầu tiên) và một cột bất kỳ có các hệ số luôn khác 0, ký hiệu cột có các hệ số luôn khác 0 là cột thứ T.

S313, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S3 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E1S3 gồm *m* phần tử của tập X12 và *m* phần tử của tập khóa con K2, trong đó X13 được tính như sau:

$$X13 = F(X12, K2) = \{F_i(X12_i + K2_i)\} \text{ với } 1 \le i \le m \quad (3.4)$$

biểu diễn toán học của biểu thức (3.4) như sau:

$$\begin{cases} X13_1 = B_{1,1} * (X12_1 + K2_1) + B_{1,2} * (X12_2 + K2_2) + \cdots + B_{1,m} * (X12_m + K2_m) \\ X13_2 = B_{2,1} * (X12_1 + K2_1) + B_{2,2} * (X12_2 + K2_2) + \cdots + B_{2,m} * (X12_m + K2_m) \\ \cdots \\ X13_m = B_{m,1} * (X12_1 + K2_1) + B_{m,2} * (X12_2 + K2_2) + \cdots + B_{m,m} * (X12_m + K2_m) \end{cases}$$

(3.5)

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulhai trong trường nhị phân GF (2),

ma trận hệ số B là một ma trận vuông *m* hàng và *m* cột $B_{m,m}$ với các phần tử nhị phân, ma trận B bao gồm các đặc điểm sau: phương trình đầu tiên phụ thuộc phi tuyến vào *m* phần tử đầu vào, mỗi phương trình trong $m - 1$ phương trình còn lại sẽ phụ thuộc vào một phần tử đầu vào bất kỳ. Ma trận B sẽ luôn có một và chỉ một cột có m-1 phần tử là 0, kí hiệu thứ tự cột này là Z. Do X12 là tập hợp *m* các phần tử phi tuyến, nên X13 là một tập gồm *m* phần tử phi tuyến.

(S314) chuyển dạng phi tuyến thuận E1S4 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X13 và K1, mỗi tập có *m* phần tử, theo phương trình sau:

$$X14 = Sbox(X13 + K1) = \{Sbox(X13_i + K1_i)\} \text{ với } 1 \le i \le m \quad (3.6)$$

trong đó,

phép cộng là phép toán XOR-bit, do X13 là tập hợp *m* các phần tử phi tuyến, nên X13 + K1 cũng là một tập gồm *m* phần tử phi tuyến,

Sbox là hàm thay thế để biến tổng phi tuyến của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X14 có *m* phần tử.

S315, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S5 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X14 để tạo ra một kết quả của tập X15, X15 gồm *m* phần tử kết quả, trong đó X15 được tính như sau:

$$X15 = F(X14) = \{F_i(X14_i)\} \text{ với } 1 \le i \le m \quad (3.7)$$

Biểu diễn toán học của biểu thức (3.7) như sau:

$$\begin{cases} X15_1 = C_{1,1} * X14_1 + C_{1,2} * X14_2 + \cdots + C_{1,m} * X14_m \\ X15_2 = C_{2,1} * X14_1 + C_{2,2} * X14_2 + \cdots + C_{2,m} * X14_m \\ \cdots \\ X15_m = C_{m,1} * X14_1 + C_{m,2} * X14_2 + \cdots + C_{m,m} * X14_m \end{cases} \quad (3.8)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số C là một ma trận vuông *m* hàng và *m* cột $C_{m,m}$ với các phần tử nhị phân, ma trận hệ số C có các đặc điểm: chỉ phương trình thứ *m* phụ thuộc vào *m* phần tử đầu vào, các phương trình còn lại phụ thuộc vào m-1 phần tử đầu vào.

Các phần tử của ma trận C là đối xứng theo trục ngang với các phần tử của ma trận A hay ma trận C là phép hoán đổi đối xứng các hàng của ma trận A theo trục hoành để tạo sự xáo trộn ngẫu nhiên đối xứng, biểu diễn của ma trận C theo ma trận A như sau:

$$C_{k,i} = A_{m-k+1,i} \text{ với } 1 \le k \le m, 1 \le i \le m \quad (3.9)$$

Cũng giống như ma trận A, ma trận C cũng luôn tồn tại một hàng (hàng cuối cùng) và một cột bất kỳ có các hệ số luôn khác 0, ký hiệu cột có các hệ số luôn khác 0 là cột thứ T.

S316, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S6 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E1S6 gồm *m* phần tử của tập X15 và *m* phần tử của tập khóa con K2, trong đó X16 được tính như sau:

$$X16 = F(X15, K2) = \{F_i(X15_i + K2_i)\} \text{ với } 1 \le i \le m \quad (3.10)$$

Biểu diễn toán học của biểu thức (3.10) như sau:

$$\begin{cases} X16_1 = D_{1,1} * (X15_1 + K2_1) + D_{1,2} * (X15_2 + K2_2) + \cdots + D_{1,m} * (X15_m + K2_m) \\ X16_2 = D_{2,1} * (X15_1 + K2_1) + D_{2,2} * (X15_2 + K2_2) + \cdots + D_{2,m} * (X15_m + K2_m) \\ \cdots \\ X16_m = D_{m,1} * (X15_1 + K2_1) + D_{m,2} * (X15_2 + K2_2) + \cdots + D_{m,m} * (X15_m + K2_m) \end{cases} \quad (3.11)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số D là một ma trận vuông *m* hàng và *m* cột $D_{m,m}$ với các phần tử nhị phân, các phần tử của D là đối xứng theo trục ngang với các phần tử của ma trận B hay ma trận D là phép hoán đổi đối xứng các hàng của ma trận B theo trục hoành để tạo sự xáo trộn ngẫu nhiên đối xứng, do đó ma trận D cũng sẽ luôn có một cột có m-1 phần tử là 0, kí hiệu thứ tự cột này là Z, biểu diễn toán học của D theo B như sau:

$$D_{k,i} = B_{m-k+1,i} \text{ với } 1 \leq k \leq m, 1 \leq i \leq m \quad (3.12)$$

Do X15 là tập hợp *m* các phần tử phi tuyến, nên X15 + K2 cũng là một tập gồm *m* phần tử phi tuyến, do đó mỗi phương trình trong biểu thức (3.11) là một phương trình phi tuyến.

(S317) chuyển dạng phi tuyến thuận E1S7 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X16 và K3, mỗi tập có *m* phần tử, X17 được tính như sau

$$X17 = Sbox(X16) + K3 = \{Sbox(X16_i) + K3_i\} \text{ với } 1 \leq i \leq m \quad (3.13)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X17 có *m* phần tử.

Do X16 là tập hợp *m* các phần tử phi tuyến, nên Sbox(X16) + K3 cũng là một tập gồm *m* phần tử phi tuyến.

ở bước S302 có bảy bước thực hiện từ E2S1 đến E2S7, khối dữ liệu X17 và ba tập khóa con K4, K5 và K6, khối dữ liệu X17 có độ dài là tập *m* phần tử, trong đó các bước được thực hiện như sau:

(S321) chuyển dạng phi tuyến thuận E2S1 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X17 và K4, mỗi tập có *m* phần tử, X21 được tính như sau:

$$X21 = Sbox(X17 + K4) = \{Sbox(X17_i + K4_i)\} \text{ với } 1 \le i \le m \quad (3.14)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X21 có *m* phần tử.

S322, mã hoá sử dụng hệ phương trình phi tuyến thuận E2S2 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X21 để tạo ra một kết quả của tập X22, X22 cũng cũng gồm *m* phần tử kết quả, X22 được tính như sau:

$$X22 = F(X21) = \{F_i(X21_i)\} \text{ với } 1 \le i \le m \quad (3.15)$$

Biểu diễn toán học của biểu thức (3.15) như sau:

$$\begin{cases} X22_1 = A_{1,1} * X21_1 + A_{1,2} * X21_2 + \cdots + A_{1,m} * X21_m \\ X22_2 = A_{2,1} * X21_1 + A_{2,2} * X21_2 + \cdots + A_{2,m} * X21_m \\ \cdots \\ X22_m = A_{m,1} * X21_1 + A_{m,2} * X21_2 + \cdots + A_{m,m} * X21_m \end{cases} \quad (3.16)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số A là ma trận có chức năng như đã mô tả ở S312, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S2.

S323, mã hoá sử dụng hệ phương trình phi tuyến thuận E2S3 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E2S3 gồm *m* phần tử của tập X22 và *m* phần tử của tập khóa con K5, X23 được tính như sau:

$$X23 = F(X22, K5) = \{F_i(X22_i + K5_i)\} \text{ với } 1 \le i \le m \quad (3.17)$$

Biểu diễn toán học của biểu thức (3.17) như sau:

$$\begin{cases} X23_1 = B_{1,1} * (X22_1 + K5_1) + B_{1,2} * (X22_2 + K5_2) + \cdots + B_{1,m} * (X22_m + K5_m) \\ X23_2 = B_{2,1} * (X22_1 + K5_1) + B_{2,2} * (X22_2 + K5_2) + \cdots + B_{2,m} * (X22_m + K5_m) \\ \cdots \\ X23_m = B_{m,1} * (X22_1 + K5_1) + B_{m,2} * (X22_2 + K5_2) + \cdots + B_{m,m} * (X22_m + K5_m) \end{cases} \quad (3.18)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận B là ma trận có chức năng như đã mô tả ở S313, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S3.

(S324) chuyển dạng phi tuyến E2S4 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X23 và K4, X24 có *m* phần tử, X24 được tính như sau:

$$X24 = Sbox(X23 + K4) = \{Sbox(X23_i + K4_i)\} \text{ với } 1 \le i \le m \qquad (3.19)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X24 có *m* phần tử.

Do X23 là tập hợp *m* các phần tử phi tuyến, nên X24 cũng là một tập gồm *m* phần tử phi tuyến.

(S325) mã hoá sử dụng hệ phương trình phi tuyến thuận E2S5 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X24 để tạo ra một kết quả của tập X25, X25 gồm *m* phần tử kết quả, X25 được tính như sau:

$$X25 = F(X24) = \{F_i(X24_i)\} \text{ với } 1 \le i \le m \qquad (3.20)$$

Biểu diễn toán học của biểu thức (3.20) như sau:

$$\begin{cases} X25_1 = C_{1,1} * X24_1 + C_{1,2} * X24_2 + \cdots + C_{1,m} * X24_m \\ X25_2 = C_{2,1} * X24_1 + C_{2,2} * X24_2 + \cdots + C_{2,m} * X24_m \\ \cdots \\ X25_m = C_{m,1} * X24_1 + C_{m,2} * X24_2 + \cdots + C_{m,m} * X24_m \end{cases} \qquad (3.21)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán modul hai trong trường nhị phân GF (2),

ma trận C là ma trận có chức năng như đã mô tả ở S315, mã hoá sử dụng hệ phương trình phi tuyến thuận E1S5.

S326, mã hoá sử dụng hệ phương trình phi tuyến thuận E2S6 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E2S6 gồm *m* phần tử của tập X25 và *m* phần tử của tập khóa con K5, X26 được tính như sau:

$$X26 = F(X25, K5) = \{F_i(X25_i + K5_i)\} \text{ với } 1 \le i \le m \qquad (3.22)$$

Biểu diễn toán học của biểu thức (3.22) như sau:

$$\begin{cases} X26_1 = D_{1,1} * (X25_1 + K5_1) + D_{1,2} * (X25_2 + K5_2) + \cdots + D_{1,m} * (X25_m + K5_m) \\ X26_2 = D_{2,1} * (X25_1 + K5_1) + D_{2,2} * (X25_2 + K5_2) + \cdots + D_{2,m} * (X25_m + K5_m) \\ \cdots \\ X26_m = D_{m,1} * (X25_1 + K5_1) + D_{m,2} * (X25_2 + K5_2) + \cdots + D_{m,m} * (X25_m + K5_m) \end{cases} \quad (3.23)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận D là ma trận có chức năng như đã mô tả ở "(S316) mã hoá sử dụng hệ phương trình phi tuyến thuận E1S6".

(S327) chuyển dạng phi tuyến E2S7 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X26 và K6, mỗi tập có *m* phần tử, khối dữ liệu mã hóa Y được tính như sau:

$$Y = Sbox(X26 + K6) = \{Sbox(X26_i + K6_i)\} \text{ với } 1 \leq i \leq m \quad (3.24)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y có *m* phần tử.

Do X26 là tập hợp *m* các phần tử phi tuyến, nên khối dữ liệu mã hóa Y cũng là một tập gồm *m* phần tử.

Ở bước S4, giải mã văn bản đã được mã hóa, chia văn bản mã hóa thành các khối dữ liệu Y có độ dài bằng độ dài khóa K0, như thể hiện ở Hình 4a, giải mã khối dữ liệu Y trong đó bao gồm các bước:

S401 sử dụng Y và ba tập khóa con K4, K5, K6,

S402 sử dụng đầu ra của S401 và ba tập khóa con K1, K2, K3,

kết quả đầu ra của S402 là khối dữ liệu thô X, có độ dài bằng độ dài của khối dữ liệu mã hoá Y, tập hợp các khối dữ liệu X để được văn bản thô ban đầu.

Trong đó, chi tiết S401 và S402 được thể hiện ở Hình 4b, cụ thể như sau: S401 có bảy bước từ từ D1S1 tới D1S7, S402 có bảy bước từ D2S1 tới D2S7, đầu vào của S402 là kết quả đầu ra của S401. Khối dữ liệu mã hóa Y và các khóa con K1 tới K6 được định dạng thành các tập hợp 8-bit (byte). Gọi n là độ dài theo bit, gọi *m* là độ dài theo byte, ta có $m = n/8$

Ở bước S401, sử dụng Y và ba tập khóa con K4, K5, K6, trong đó các bước giải mã dữ liệu được thực hiện như sau:

(S411) chuyển dạng phi tuyến ngược D1S1 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số Y và K6, mỗi tập có *m* phần tử, Y11 được tính như sau:

$$Y11 = ISbox(Y + K6) = \{ISbox(Y_i + K6_i)\} \text{ với } 1 \le i \le m \qquad (4.1)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y11 có *m* phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải mã.

S412, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S2 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập Y11 để tạo ra một kết quả của tập Y12, Y12 gồm *m* phần tử kết quả, Y12 được tính như sau:

$$Y12 = F(Y11) = \{F_i(Y11_i)\} \text{ với } 1 \le i \le m \qquad (4.2)$$

biểu diễn toán học của biểu thức (4.3) như sau:

$$\begin{cases} Y12_1 = E_{1,1} * Y11_1 + E_{1,2} * Y11_2 + \cdots + E_{1,m} * Y11_m \\ Y12_2 = E_{2,1} * Y11_1 + E_{2,2} * Y11_2 + \cdots + E_{2,m} * Y11_m \\ \cdots \\ Y12_m = E_{m,1} * Y11_1 + E_{m,2} * Y11_2 + \cdots + E_{m,m} * Y11_m \end{cases} \qquad (4.3)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận *E* là một ma trận vuông *m* hàng và *m* cột $E_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận *D* theo quy ước sau:

- Hàng thứ Z của ma trận *E* bằng hàng thứ *m* của ma trận *D*:
  $E_{Z,i} = D_{m,i}$ với $1 \le i \le m$
- Các phần tử còn lại của ma trận *E* là các phần tử còn lại của ma trận *D* ở vị trí đối xứng qua đường chéo chính của ma trận *D*:
  $E_{k,i} = D_{i,k}$ với $1 \le k \le m, 1 \le i < m, k \ne Z$

Với quy ước nội suy trên, mà trận *E* vẫn giữ nguyên được thuộc tính xáo trộn ngẫu nhiên đối xứng của ma trận *D*.

S413, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S3 thực hiện m phương trình phi tuyến, các phần tử đầu vào của khối D1S3 gồm m phần tử của tập Y12 và m phần tử của tập khóa con K5, Y13 được tính như sau:

$$Y13 = F(Y12, K5) = \{F_i(Y12_i + K5_i)\} \text{ với } 1 \le i \le m \quad (4.4)$$

biểu diễn toán học của biểu thức (4.4) như sau:

$$\begin{cases} Y13_1 = G_{1,1} * (Y12_1 + K5_1) + G_{1,2} * (Y12_2 + K5_2) + \cdots + G_{1,m} * (Y12_m + K5_m) \\ Y13_2 = G_{2,1} * (Y12_1 + K5_1) + G_{2,2} * (Y12_2 + K5_2) + \cdots + G_{2,m} * (Y12_m + K5_m) \\ \cdots \\ Y13_m = G_{m,1} * (Y12_1 + K5_1) + G_{m,2} * (Y12_2 + K5_2) + \cdots + G_{m,m} * (Y12_m + K5_m) \end{cases} \quad (4.5)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận G là một ma trận vuông m hàng và m cột $G_{m,m}$ với các phần tử nhị phân, ma trận G được nội suy từ ma trận C theo các quy ước sau:

- Hàng thứ T của ma trận G có $m-1$ phần tử là lặp lại hàng thứ m của ma trận C:
  $G_{T,i} = C_{m,i}$ với $1 \le i \le m-1$
- Cột thứ m của ma trận G có $m-1$ phần tử là lặp lại cột thứ T của ma trận C:
  $G_{k,m} = C_{k,T}$ với $1 \le k \le m, k \ne T$; $G_{T,m} = 0$
- Các phần tử còn lại của ma trận G là nghịch đảo các phần tử còn lại của ma trận C ở vị trí đối xứng qua đường chéo chính của ma trận C:
  $G_{k,i} = \bar{C}_{i,k}$ với $1 \le k \le m, 1 \le i \le m-1, k \ne T$

Do Y12 là tập hợp m các phần tử phi tuyến, nên Y13 cũng là một tập gồm m phần tử phi tuyến,

(S414) chuyển dạng phi tuyến ngược D1S4 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y13 và K4, mỗi tập có m phần tử, Y14 được tính như sau:

$$Y14 = ISbox(Y13) + K4 = \{ISbox(Y13_i) + K4_i\} \text{ với } 1 \le i \le m \quad (4.6)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y14 có *m* phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải mã;

Bởi vì Y13 là tập hợp *m* các phần tử phi tuyến, nên $ISbox(Y13) + K4$ cũng là một tập gồm *m* phần tử phi tuyến.

S415, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S5 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử phi tuyến của tập Y14 để tạo ra một kết quả của tập Y15, Y15 gồm *m* phần tử kết quả phi tuyến, Y15 được tính như sau:

$$Y15 = F(Y14) = \{F_i(Y14_i)\} \text{ với } 1 \le i \le m \quad (4.7)$$

Biểu diễn toán học của biểu thức (4.7) như sau:

$$\begin{cases} Y15_1 = H_{1,1} * Y14_1 + H_{1,2} * Y14_2 + \cdots + H_{1,m} * Y14_m \\ Y15_2 = H_{2,1} * Y14_1 + H_{2,2} * Y14_2 + \cdots + H_{2,m} * Y14_m \\ \cdots \\ Y15_m = H_{m,1} * Y14_1 + H_{m,2} * Y14_2 + \cdots + H_{m,m} * Y14_m \end{cases} \quad (4.8)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận *H* là một ma trận vuông *m* hàng và *m* cột $H_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận *B* theo quy ước sau:

- Hàng thứ *Z* của ma trận *H* bằng hàng thứ *1* của ma trận *B*:
  $H_{Z,i} = B_{1,i}$ với $1 \le i \le m$
- Các phần tử còn lại của ma trận *H* là các phần tử còn lại của ma trận *B* ở vị trí đối xứng qua đường chéo chính của ma trận *B*:
  $H_{k,i} = B_{i,k}$ với $1 \le k \le m, 1 < i \le m, k \ne Z$

Với quy ước nội suy trên, mà trận *H* vẫn giữ nguyên được thuộc tính xáo trộn ngẫu nhiên đối xứng của ma trận *B*.

S416, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S6 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối D1S6 gồm *m* phần tử của tập Y15 và *m* phần tử của tập khóa con K5, Y16 được tính như sau:

$$Y16 = F(Y15, K5) = \{F_i(Y15_i + K5_i)\} \text{ với } 1 \le i \le m \quad (4.9)$$

Biểu diễn toán học của biểu thức (39) như sau:

$$\begin{cases} Y16_1 = L_{1,1} * (Y15_1 + K5_1) + L_{1,2} * (Y15_2 + K5_2) + \cdots + L_{1,m} * (Y15_m + K5_m) \\ Y16_2 = L_{2,1} * (Y15_1 + K5_1) + L_{2,2} * (Y15_2 + K5_2) + \cdots + L_{2,m} * (Y15_m + K5_m) \\ \cdots \\ Y16_m = L_{m,1} * (Y15_1 + K5_1) + L_{m,2} * (Y15_2 + K5_2) + \cdots + L_{m,m} * (Y15_m + K5_m) \end{cases} \quad (4.10)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số L là một ma trận vuông m hàng và m cột $L_{m,m}$ với các phần tử nhị phân, Ma trận L được nội suy từ ma trận A theo các quy ước sau:

- Hàng thứ T của ma trận L có $m-1$ phần tử là lặp lại hàng thứ 1 của ma trận A:
  $L_{T,i} = A_{1,i}$ với $1 < i \leq m$
- Cột thứ 1 của ma trận L có $m-1$ phần tử là lặp lại cột thứ T của ma trận A:
  $L_{k,1} = A_{k,T}$ với $1 \leq k \leq m, k \neq T$; $L_{T,1} = 0$
- Các phần tử còn lại của ma trận L là nghịch đảo các phần tử còn lại của ma trận A ở vị trí đối xứng qua đường chéo chính của ma trận A:
  $L_{k,i} = \bar{A}_{i,k}$ với $1 \leq k \leq m, 1 < i \leq m, k \neq T$

Do Y15 là tập hợp m các phần tử phi tuyến, nên $Y15 + K5$ cũng là một tập gồm m phần tử phi tuyến. Do vậy, Y16 cũng là một tập gồm m phần tử phi tuyến.

S417 chuyển dạng phi tuyến ngược D1S7 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y16 và K4, mỗi tập có m phần tử, Y17 được tính như sau:

$$Y17 = ISbox(Y16) + K4 = \{ISbox(Y16_i) + K4_i\} \text{ với } 1 \leq i \leq m \quad (4.11)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y17 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật.

Bởi vì Y16 là tập hợp m các phần tử phi tuyến, nên $ISbox(Y16) + K4$ cũng là một tập gồm m phần tử phi tuyến.

Ở bước S402, sử dụng Y và ba tập khóa con K1, K2 và K3, kết quả của S401 là một tập m phần tử phi tuyến Y17, kết quả này là đầu vào của S402, trong đó các bước giải mã dữ liệu được thực hiện như sau:

S421 chuyển dạng phi tuyến ngược D2S1 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y17 và K3, mỗi tập có m phần tử, Y21 được tính như sau:

$$Y21 = ISbox(Y17 + K3) = \{ISbox(Y17_i + K3_i)\} \text{ với } 1 \le i \le m \quad (4.12)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổ hợp tuyến tính của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y21 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo mã hoá dữ liệu.

S422, mã hoá sử dụng hệ phương trình phi tuyến ngược D2S2 thực hiện m phương trình, mỗi phương trình xử lý m phần tử của tập Y21 để tạo ra một kết quả của tập Y22, Y22 gồm m phần tử kết quả, Y22 được tính như sau:

$$Y22 = F(Y21) = \{F_i(Y21_i)\} \text{ với } 1 \le i \le m \quad (4.13)$$

Biểu diễn toán học của biểu thức (4.13) như sau:

$$\begin{cases} Y22_1 = E_{1,1} * Y21_1 + E_{1,2} * Y21_2 + \cdots + E_{1,m} * Y21_m \\ Y22_2 = E_{2,1} * Y21_1 + E_{2,2} * Y21_2 + \cdots + E_{2,m} * Y21_m \\ \cdots \\ Y22_m = E_{m,1} * Y21_1 + E_{m,2} * Y21_2 + \cdots + E_{m,m} * Y21_m \end{cases} \quad (4.14)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số E là một ma trận vuông m hàng và m cột $E_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận D như mô tả trong S412, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S2.

S423, mã hoá sử dụng hệ phương trình phi tuyến ngược D2S3 thực hiện m phương trình phi tuyến, phần tử đầu vào của khối D2S3 gồm m phần tử của tập Y22 và m phần tử của tập khóa con K2, Y23 được tính như sau:

$$Y23 = F(Y22, K2) = \{F_i(Y22_i + K2_i)\} \text{ với } 1 \le i \le m \quad (4.15)$$

Biểu diễn toán học của biểu thức (4.15) như sau:

$$\begin{cases} Y23_1 = G_{1,1} * (Y22_1 + K2_1) + G_{1,2} * (Y22_2 + K2_2) + \cdots + G_{1,m} * (Y22_m + K2_m) \\ Y23_2 = G_{2,1} * (Y22_1 + K2_1) + G_{2,2} * (Y22_2 + K2_2) + \cdots + G_{2,m} * (Y22_m + K2_m) \\ \cdots \\ Y23_m = G_{m,1} * (Y22_1 + K2_1) + G_{m,2} * (Y22_2 + K2_2) + \cdots + G_{m,m} * (Y22_m + K2_m) \end{cases} \quad (4.16)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số G là một ma trận vuông m hàng và m cột $G_{m,m}$ với các phần tử nhị phân. được nội suy từ ma trận C như mô tả trong S413, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S3.

S424) chuyển dạng phi tuyến ngược D2S4 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y23 và K1, mỗi tập có m phần tử, Y24 được tính như sau:

$$Y24 = ISbox(Y23) + K1 = \{ISbox(Y23_i) + K1_i\} \text{ với } 1 \leq i \leq m \quad (4.17)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y24 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật.

Bởi vì Y23 là tập hợp m các phần tử phi tuyến, nên $ISbox(Y23) + K1$ cũng là một tập gồm m phần tử phi tuyến.

S425, mã hoá sử dụng hệ phương trình phi tuyến ngược D2S5 thực hiện m phương trình, mỗi phương trình xử lý m phần tử phi tuyến của tập Y24 để tạo ra một kết quả của tập Y25, Y25 gồm m phần tử kết quả phi tuyến, Y25 được tinh như sau:

$$Y25 = F(Y24) = \{F_i(Y24_i)\} \text{ với } 1 \leq i \leq m \quad (4.18)$$

Biểu diễn toán học của biểu thức (4.18) như sau:

$$\begin{cases} Y25_1 = H_{1,1} * Y24_1 + H_{1,2} * Y24_2 + \cdots + H_{1,m} * Y24_m \\ Y25_2 = H_{2,1} * Y24_1 + H_{2,2} * Y24_2 + \cdots + H_{2,m} * Y24_m \\ \cdots \\ Y25_m = H_{m,1} * Y24_1 + H_{m,2} * Y24_2 + \cdots + H_{m,m} * Y24_m \end{cases} \quad (4.19)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số H là một ma trận vuông m hàng và m cột $H_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận B như mô tả ở S415, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S5.

S426, mã hoá sử dụng hệ phương trình phi tuyến ngược D2S6 thực hiện *m* phương trình phi tuyến, phần tử đầu vào của khối D2S6 gồm *m* phần tử của tập Y25 và *m* phần tử của tập khóa con K2, Y26 được tính như sau

$$Y26 = F(Y25, K2) = \{F_i(Y25_i + K2_i)\} \text{ với } 1 \le i \le m \qquad (4.20)$$

Biểu diễn toán học của biểu thức (4.20) như sau:

$$\begin{cases} Y26_1 = L_{1,1} * (Y25_1 + K2_1) + L_{1,2} * (Y25_2 + K2_2) + \cdots + L_{1,m} * (Y25_m + K2_m) \\ Y26_2 = L_{2,1} * (Y25_1 + K2_1) + L_{2,2} * (Y25_2 + K2_2) + \cdots + L_{2,m} * (Y25_m + K2_m) \\ \cdots \\ Y26_m = L_{m,1} * (Y25_1 + K2_1) + L_{m,2} * (Y25_2 + K2_2) + \cdots + L_{m,m} * (Y25_m + K2_m) \end{cases} \qquad (4.21)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số *L* là một ma trận vuông *m* hàng và *m* cột $L_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận *A* như mô tả ở S416, mã hoá sử dụng hệ phương trình phi tuyến ngược D1S6.

S427) chuyển dạng phi tuyến ngược D2S7 thực hiện *m* hàm chuyển dạng phi tuyến của 2 tập biến số Y26 và K1, mỗi tập có *m* phần tử, khối dữ liệu X được tính như sau:

$$X = ISbox(Y26) + K1 = \{ISbox(Y26_i) + K1_i\} \text{ với } 1 \le i \le m \qquad (4.22)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X có *m* phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật.

Bởi vì Y26 là tập hợp *m* các phần tử phi tuyến, nên $ISbox(Y26) + K1$ cũng là một tập gồm *m* phần tử phi tuyến. Do đó khối dữ liệu X cũng là một tập gồm *m* phần tử phi tuyến.

Ví dụ thực hiện sáng chế

Xử lý văn bản gốc có độ dài 10000 byte bằng phương pháp mã hoá theo sáng chế. Theo đó, chọn khóa K0 = 64 byte (64 x 8 = 512 bit), và theo quy trình của sáng chế để có được 6 khoá con một chiều K1-K6; chia văn bản gốc thành các khối X có độ dài bằng K0, 10000/64 = 156 khối x 64-byte + 1 khối x 16-byte. Lưu ý rằng, khối có độ dài 16-byte này sẽ được đệm thêm 48-byte (gồm các bit 0

hoặc 1 do người dùng quy ước) để khối này có đủ độ dài 64-byte. Như vậy, tổng cộng sẽ được 157 khối có độ dài 64-byte. Sau đó, mỗi khối 64-byte này sẽ được mã hóa và thu được tổng cộng sẽ có 157 khối mã hóa Y có độ dài 64-byte.

Ở chiều giải mã, 157 khối mã hóa Y được giải mã thành 157 khối dữ liệu thô X, khối từ 1 tới 156 là dữ liệu gốc ban đầu, khối thứ 157 sẽ được bỏ đi 48-byte đệm để lấy ra 16-byte dữ liệu gốc.

Hiệu quả đạt được của sáng chế

Phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa theo sáng chế khắc phục được nhược điểm của các phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi khác đã biết bằng cách tạo ra một kiến trúc mật mã là các hệ phương trình phi tuyến, xử lý các khối dữ liệu có độ dài bằng độ dài khóa K0, từ 128-bit tới 8192-bit với mỗi bước thay đổi là 16-bit, không sử dụng các bảng tìm kiếm thay thế 32-bit, ngoài các S-box 8-bit, thay đổi độ dài khóa dễ dàng mà không ảnh hưởng tới chất lượng giải thuật, phát triển một phương pháp mới tạo các khóa con (subkey) một chiều có mức độ độc lập tuyến tính lẫn nhau cao, thực hiện một cơ chế mã hóa/giải mã mới là tối thiểu vòng lặp kiến trúc mạng và tối đa số lượng khóa con có thể sử dụng trên mỗi vòng lặp. Cụ thể hơn, cơ chế mã hóa và giải mã sử dụng 2 vòng lặp kiến trúc mạng và 3 khóa con riêng biệt cho mỗi vòng lặp. Phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa theo sáng chế dễ dàng triển khai trên mọi nền tảng bộ vi xử lý mà không ảnh hưởng tới các hiệu quả bảo mật, tạo ra một phương pháp mã hóa/giải mã mới có độ dài khóa và độ dài khối dữ liệu biến đổi ngẫu nhiên.

Trên đây, sáng chế đã được mô tả chi tiết theo các phương án ưu tiên thực hiện, và có thể kèm theo các phương án thay thế hoặc tương đương hoặc các ví dụ cụ thể, sử dụng các mô tả và thuật ngữ phù hợp để người có hiểu biết trung bình về lĩnh vực kỹ thuật này có thể hiểu và thực hiện được giải pháp theo sáng chế. Vì vậy, người có hiểu biết trung bình về lĩnh vực kỹ thuật tương ứng có thể dễ dàng tạo ra các thay đổi, cải biến, hoặc thay thế tương đương dựa vào các nội dung và phương án được mô tả. Do đó, các thay đổi, cải biến, hoặc thay thế tương đương này được coi là không nằm ngoài phạm vi của sáng chế, và phạm vi bảo hộ của sáng chế hiển nhiên là không bị giới hạn bởi các nội dung và phương án được mô tả mà được xác định trong yêu cầu bảo hộ dưới đây.

YÊU CẦU BẢO HỘ

1. Phương pháp mã hóa và giải mã khối dữ liệu có kích thước biến đổi theo độ dài khóa bao gồm các bước:

(S1) khởi tạo khóa ban đầu K0 gồm *m* phần tử có độ dài từ 16-byte (128-bit) đến 256-byte (8192-bit), trong đó *m* là số chẵn byte,

(S2) tạo 6 tập khóa con một chiều K1-K6 dựa trên khoá K0 và hàm S-box để sử dụng cho bước mã hóa (S3) và bước giải mã dữ liệu (S4) bao gồm:

(S211) tạo ba khóa phụ Ks11, Ks12 và Ks13 có độ dài bằng nhau từ khoá K0;

khởi tạo: $\mathrm{idx} = \mathrm{Sbox}(\mathrm{K0_0}\text{^}\mathrm{K0_1})$ (1.1)

lặp lại *m* lần các bước sau:

$$\begin{cases} temp = \mathrm{Sbox}(\mathrm{idx}) \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor ; j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \mathrm{Ks11_i} = \mathrm{temp} \\ idx = \left\lfloor \frac{idx+1+(\mathrm{K0_j}\text{^}\mathrm{K0_{j1}})}{256} \right\rfloor \\ \mathrm{Ks12_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\mathrm{Ks12_i}\text{^}\mathrm{K0_{j1}})}{256} \right\rfloor \\ \mathrm{Ks13_i} = \mathrm{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \mathrm{i} \le \mathrm{m}-1 \quad (1.2)$$

(S211') tạo ra tập khóa con K1 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, bao gồm bốn tập dữ liệu đầu vào K0, Ks11, Ks12, Ks13, mỗi tập có *m* phần tử;

(S212) tạo ba khóa phụ có cùng độ dài Ks21, Ks22, Ks23 bằng cách sử dụng khóa K0 và khóa con K1, trong đó:

khởi tạo: $\mathrm{idx} = \mathrm{Sbox}(\mathrm{K0_0}\text{^}\mathrm{K1_1})$ (2.1)

lặp lại *m* lần các bước sau:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor;\ j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks21}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j\text{\^{}K1}_{j1})}{256} \right\rfloor \\ \text{Ks22}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks22}_i\text{\^{}K1}_{j1})}{256} \right\rfloor \\ \text{Ks23}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \quad \text{với } 0 \le i \le m-1 \qquad (2.2)$$

(S212') tạo ra tập khóa con K2 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận với 4 tập dữ liệu đầu vào K1, Ks21, Ks22, Ks23, mỗi tập có *m* phần tử;

(S213) tạo 3 khóa phụ Ks31, Ks32, Ks33 có độ dài bằng nhau bằng cách sử dụng khóa K0 và khóa con K2, trong đó:

khởi tạo: $\text{idx} = \text{Sbox}(\text{K0}_0\text{\^{}K2}_1)$ (3.1)

lặp lại *m* lần các bước sau:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor;\ j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks31}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j\text{\^{}K2}_{j1})}{256} \right\rfloor \\ \text{Ks32}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks32}_i\text{\^{}K2}_{j1})}{256} \right\rfloor \\ \text{Ks33}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \quad \text{với } 0 \le i \le m-1 \qquad (3.2)$$

(S213') tạo ra tập khóa con K3 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn, với 4 tập dữ liệu đầu vào K2, Ks31, Ks32, Ks33, mỗi tập có *m* phần tử;

(S214) tạo ba khóa phụ Ks41, Ks42, Ks43 có độ dài bằng nhau sử dụng khoá K0 và khóa con K3, trong đó;

khởi tạo: $\text{idx} = \text{Sbox}(\text{K0}_0\text{\^{}K3}_1)$ (4.1)

lặp lại *m* lần (với *m* là độ dài khóa tính theo byte) các bước sau:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor;\ j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks41}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j\text{\^{}K3}_{j1})}{256} \right\rfloor \\ \text{Ks42}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks42}_i\text{\^{}K3}_{j1})}{256} \right\rfloor \\ \text{Ks43}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \text{i} \le \text{m} - 1 \quad (4.2)$$

(S214') tạo ra tập khóa con K4 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, với 4 tập dữ liệu đầu vào K3, Ks41, Ks42, Ks43, mỗi tập có *m* phần tử;

(S215) tạo ba khóa phụ Ks51, Ks52, Ks53 có độ dài bằng K0 bằng cách sử dụng K0 và khóa con K4, trong đó:

khởi tạo: $\text{idx} = \text{Sbox}(\text{K0}_0\text{\^{}K4}_1)$ (5.1)

lặp lại *m* lần (với *m* là độ dài khóa tính theo byte) các bước sau:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor;\ j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks51}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j\text{\^{}K4}_{j1})}{256} \right\rfloor \\ \text{Ks52}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks52}_i\text{\^{}K4}_{j1})}{256} \right\rfloor \\ \text{Ks53}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \text{i} \le \text{m} - 1 \quad (5.2)$$

(S215') tạo ra tập khóa con K5 có *m* phần tử bằng cách sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận, với bốn tập dữ liệu đầu vào K4, Ks51, Ks52, Ks53, mỗi tập có *m* phần tử;

khởi tạo tập khóa con K6, bao gồm:

(S216) tạo ba khóa phụ Ks61, Ks62, Ks63 có độ dài bằng K0 bằng cách sử dụng K0 và khóa con K5, trong đó:

khởi tạo: $\text{idx} = \text{Sbox}(\text{K0}_0\text{\^{}K5}_1)$ (6.1)

lặp lại *m* lần (với *m* là độ dài khóa tính theo byte) các bước sau:

$$\begin{cases} temp = \text{Sbox(idx)} \\ j = \left\lfloor \frac{i+(m\gg 1)}{m} \right\rfloor;\ j1 = \left\lfloor \frac{j+1}{m} \right\rfloor \\ \text{Ks61}_i = \text{temp} \\ idx = \left\lfloor \frac{idx+1+(\text{K0}_j \text{\^{}} \text{K5}_{j1})}{256} \right\rfloor \\ \text{Ks62}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1+(\text{Ks62}_i \text{\^{}} \text{K5}_{j1})}{256} \right\rfloor \\ \text{Ks63}_i = \text{Sbox}(idx) \\ idx = \left\lfloor \frac{idx+1}{256} \right\rfloor \end{cases} \text{ với } 0 \le \text{i} \le \text{m} - 1 \qquad (6.2)$$

(S216') tạo ra tập khóa con K6 có *m* phần tử sử dụng kiến trúc mạng phương trình phi tuyến đa ẩn thuận với bốn tập dữ liệu đầu vào K5, Ks61, Ks62, Ks63, mỗi tập có *m* phần tử;

(S3) mã hóa khối dữ liệu văn bản thô đầu vào, chia văn bản thô đầu vào thành các khối dữ liệu X có độ dài bằng độ dài khóa K0, trong trường hợp, văn bản thô đầu vào không chia hết cho *m*, các byte đệm 0 hoặc 1 do người dùng quy ước được thêm vào khối dữ liệu X chứa các byte dữ liệu không chia hết, các bước mã hoá bao gồm:

(S301) mã hoá khối dữ liệu X bằng cách sử dụng lần lượt ba tập khóa con K1, K2, K3 để thu được tập kết quả phi tuyến đầu ra X17 có *m* phần tử,

(S302) mã hoá tập kết quả phi tuyến đầu ra X17 có *m* phần tử bằng cách sử dụng lần lượt ba tập khóa con K4, K5, K6 để thu được khối dữ liệu đã được mã hóa Y, có độ dài bằng khối dữ liệu X, tập hợp các khối dữ liệu Y để thu được văn bản mã hóa của văn bản thô đầu vào; trong đó,

(S301) bao gồm bảy bước thực hiện từ (S311) đến (S317), trong đó khối dữ liệu X và ba tập khóa con K1, K2 và K3 được định dạng thành *n* tập hợp 8-bit, cụ thể là *m*=*n*/8, cụ thể là:

(S311) chuyển dạng phi tuyến thuận E1S1 thực hiện *m* hàm chuyển dạng phi tuyến của 2 tập biến số X và K1, mỗi tập có *m* phần tử, trong đó X11 được tính như sau:

$$\text{X11} = \text{Sbox}(\text{X} + \text{K1}) = \{\text{Sbox}(\text{X}_i + \text{K1}_i)\} \text{ với } 1 \le \text{i} \le \text{m} \qquad (3.1)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X11 có *m* phần tử,

(S312) mã hoá sử dụng hệ phương trình phi tuyến thuận thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X11 để tạo ra một kết quả của tập X12, X12 gồm *m* phần tử kết quả, trong đó X12 được tính như sau:

$$X12 = F(X11) = \{F_i(X11_i)\} \text{ với } 1 \le i \le m \quad (3.2)$$

biểu diễn toán học của biểu thức (3.2) như sau:

$$\begin{cases} X12_1 = A_{1,1} * X11_1 + A_{1,2} * X11_2 + \cdots + A_{1,m} * X11_m \\ X12_2 = A_{2,1} * X11_1 + A_{2,2} * X11_2 + \cdots + A_{2,m} * X11_m \\ \cdots \\ X12_m = A_{m,1} * X11_1 + A_{m,2} * X11_2 + \cdots + A_{m,m} * X11_m \end{cases} \quad (3.3)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulo hai trong trường nhị phân GF (2),

ma trận hệ số A là một ma trận vuông *m* hàng và *m* cột $A_{m,m}$ với các phần tử nhị phân, có các đặc điểm sau: chỉ phương trình đầu tiên phụ thuộc vào *m* phần tử đầu vào, các phương trình còn lại phụ thuộc vào m-1 phần tử đầu vào, ma trận A luôn tồn tại một hàng (hàng đầu tiên) và một cột bất kỳ có các hệ số luôn khác 0, ký hiệu cột có các hệ số luôn khác 0 là cột thứ T;

(S313) biến đổi hệ phương trình phi tuyến thuận E1S3 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E1S3 gồm *m* phần tử của tập X12 và *m* phần tử của tập khóa con K2, trong đó X13 được tính như sau:

$$X13 = F(X12, K2) = \{F_i(X12_i + K2_i)\} \text{ với } 1 \le i \le m \quad (3.4)$$

biểu diễn toán học của biểu thức (3.4) như sau:

$$\begin{cases} X13_1 = B_{1,1} * (X12_1 + K2_1) + B_{1,2} * (X12_2 + K2_2) + \cdots + B_{1,m} * (X12_m + K2_m) \\ X13_2 = B_{2,1} * (X12_1 + K2_1) + B_{2,2} * (X12_2 + K2_2) + \cdots + B_{2,m} * (X12_m + K2_m) \\ \cdots \\ X13_m = B_{m,1} * (X12_1 + K2_1) + B_{m,2} * (X12_2 + K2_2) + \cdots + B_{m,m} * (X12_m + K2_m) \end{cases}$$

(3.5)

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulchai trong trường nhị phân GF (2),

ma trận hệ số B là một ma trận vuông *m* hàng và *m* cột $B_{m,m}$ với các phần tử nhị phân, ma trận B sẽ luôn có một và chỉ một cột có m-1 phần tử là 0, kí hiệu thứ tự cột này là Z,

(S314) chuyển dạng phi tuyến thuận E1S4: thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X13 và K1, mỗi tập có *m* phần tử, theo phương trình sau:

$$X14 = Sbox(X13 + K1) = \{Sbox(X13_i + K1_i)\} \text{ với } 1 \leq i \leq m \qquad (3.6)$$

trong đó,

phép cộng là phép toán XOR-bit, do X13 là tập hợp *m* các phần tử phi tuyến, nên X13 + K1 cũng là một tập gồm *m* phần tử phi tuyến,

Sbox là hàm thay thế để biến tổng phi tuyến của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X14 có *m* phần tử,

(S315) mã hoá sử dụng hệ phương trình phi tuyến thuận E1S5 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X14 để tạo ra một kết quả của tập X15, X15 gồm *m* phần tử kết quả, trong đó X15 được tính như sau:

$$X15 = F(X14) = \{F_i(X14_i)\} \text{ với } 1 \leq i \leq m \qquad (3.7)$$

biểu diễn toán học của biểu thức (3.7) như sau:

$$\begin{cases} X15_1 = C_{1,1} * X14_1 + C_{1,2} * X14_2 + \cdots + C_{1,m} * X14_m \\ X15_2 = C_{2,1} * X14_1 + C_{2,2} * X14_2 + \cdots + C_{2,m} * X14_m \\ \cdots \\ X15_m = C_{m,1} * X14_1 + C_{m,2} * X14_2 + \cdots + C_{m,m} * X14_m \end{cases} \qquad (3.8)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulchai trong trường nhị phân GF (2),

ma trận hệ số C là một ma trận vuông *m* hàng và *m* cột $C_{m,m}$ với các phần tử nhị phân,

các phần tử của ma trận C là đối xứng theo trục ngang với các phần tử của ma trận A hay ma trận C là phép hoán đổi đối xứng các hàng của ma trận A theo trục hoành để tạo sự xáo trộn ngẫu nhiên đối xứng, biểu diễn của ma trận C theo ma trận A như sau:

$$C_{k,i} = A_{m-k+1,i} \text{ với } 1 \le k \le m, 1 \le i \le m \quad (3.9)$$

tương tự như ma trận A, ma trận C cũng luôn tồn tại một hàng (hàng cuối cùng) và một cột bất kỳ có các hệ số luôn khác 0, ký hiệu cột có các hệ số luôn khác 0 là cột thứ T,

(S316) biến đổi hệ phương trình phi tuyến thuận E1S6: thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E1S6 gồm *m* phần tử của tập X15 và *m* phần tử của tập khóa con K2, trong đó X16 được tính như sau:

$$X16 = F(X15, K2) = \{F_i(X15_i + K2_i)\} \text{ với } 1 \le i \le m \quad (3.10)$$

biểu diễn toán học của biểu thức (3.10) như sau:

$$\begin{cases} X16_1 = D_{1,1} * (X15_1 + K2_1) + D_{1,2} * (X15_2 + K2_2) + \cdots + D_{1,m} * (X15_m + K2_m) \\ X16_2 = D_{2,1} * (X15_1 + K2_1) + D_{2,2} * (X15_2 + K2_2) + \cdots + D_{2,m} * (X15_m + K2_m) \\ \cdots \\ X16_m = D_{m,1} * (X15_1 + K2_1) + D_{m,2} * (X15_2 + K2_2) + \cdots + D_{m,m} * (X15_m + K2_m) \end{cases} \quad (3.11)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số D là một ma trận vuông *m* hàng và *m* cột $D_{m,m}$ với các phần tử nhị phân, các phần tử của D là đối xứng theo trục ngang với các phần tử của ma trận B hay ma trận D là phép hoán đổi đối xứng các hàng của ma trận B theo trục hoành để tạo sự xáo trộn ngẫu nhiên đối xứng, do đó ma trận D cũng sẽ luôn có một cột có m-1 phần tử là 0, kí hiệu thứ tự cột này là Z, biểu diễn toán học của D theo B như sau:

$$D_{k,i} = B_{m-k+1,i} \text{ với } 1 \le k \le m, 1 \le i \le m \quad (3.12)$$

(S317) chuyển dạng phi tuyến thuận E1S7: thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X16 và K3, mỗi tập có *m* phần tử, X17 được tính như sau

$$X17 = Sbox(X16) + K3 = \{Sbox(X16_i) + K3_i\} \text{ với } 1 \le i \le m \quad (3.13)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X17 có *m* phần tử;

bước (S302) bao gồm bảy bước thực hiện từ (S321) đến (S327) sử dụng tập kết quả phi tuyến đầu ra X17 thu được từ (S317) và lần lượt ba tập khóa con K4, K5 và K6, cụ thể là:

(S321) chuyển dạng phi tuyến thuận E2S1: thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X17 và K4, mỗi tập có *m* phần tử, X21 được tính như sau:

$$X21 = Sbox(X17 + K4) = \{Sbox(X17_i + K4_i)\} \text{ với } 1 \le i \le m \quad (3.14)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X21 có *m* phần tử,

(S322) biến đổi hệ phương trình phi tuyến thuận E2S2 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X21 để tạo ra một kết quả của tập X22, X22 cũng cũng gồm *m* phần tử kết quả, X22 được tính như sau:

$$X22 = F(X21) = \{F_i(X21_i)\} \text{ với } 1 \le i \le m \quad (3.15)$$

biểu diễn toán học của biểu thức (3.15) như sau:

$$\begin{cases} X22_1 = A_{1,1} * X21_1 + A_{1,2} * X21_2 + \cdots + A_{1,m} * X21_m \\ X22_2 = A_{2,1} * X21_1 + A_{2,2} * X21_2 + \cdots + A_{2,m} * X21_m \\ \cdots \\ X22_m = A_{m,1} * X21_1 + A_{m,2} * X21_2 + \cdots + A_{m,m} * X21_m \end{cases} \quad (3.16)$$

trong đó:

phép cộng là phép toán XOR-bit, và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2)

ma trận hệ số A là ma trận có chức năng như đã mô tả ở S312, khối biến đổi hệ phương trình phi tuyến thuận E1S2;

(S323) biến đổi hệ phương trình phi tuyến thuận E2S3 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E2S3 gồm *m* phần tử của tập X22 và *m* phần tử của tập khóa con K5, X23 được tính như sau:

$$X23 = F(X22, K5) = \{F_i(X22_i + K5_i)\} \text{ với } 1 \le i \le m \quad (3.17)$$

biểu diễn toán học của biểu thức (3.17) như sau:

$$\begin{cases} X23_1 = B_{1,1} * (X22_1 + K5_1) + B_{1,2} * (X22_2 + K5_2) + \cdots + B_{1,m} * (X22_m + K5_m) \\ X23_2 = B_{2,1} * (X22_1 + K5_1) + B_{2,2} * (X22_2 + K5_2) + \cdots + B_{2,m} * (X22_m + K5_m) \\ \cdots \\ X23_m = B_{m,1} * (X22_1 + K5_1) + B_{m,2} * (X22_2 + K5_2) + \cdots + B_{m,m} * (X22_m + K5_m) \end{cases}$$

(3.18)

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận B là ma trận có chức năng như đã mô tả ở S313, khối biến đổi hệ phương trình phi tuyến thuận E1S3;

(S324,) khối chuyển dạng phi tuyến E2S4 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X23 và K4, X24 có *m* phần tử, X24 được tính như sau:

$$X24 = Sbox(X23 + K4) = \{Sbox(X23_i + K4_i)\} \text{ với } 1 \le i \le m \quad (3.19)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X24 có *m* phần tử.

(S325) biến đổi hệ phương trình phi tuyến thuận E2S5 thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập X24 để tạo ra một kết quả của tập X25, X25 gồm *m* phần tử kết quả, X25 được tính như sau:

$$X25 = F(X24) = \{F_i(X24_i)\} \text{ với } 1 \le i \le m \quad (3.20)$$

biểu diễn toán học của biểu thức (3.20) như sau:

$$\begin{cases} X25_1 = C_{1,1} * X24_1 + C_{1,2} * X24_2 + \cdots + C_{1,m} * X24_m \\ X25_2 = C_{2,1} * X24_1 + C_{2,2} * X24_2 + \cdots + C_{2,m} * X24_m \\ \cdots \\ X25_m = C_{m,1} * X24_1 + C_{m,2} * X24_2 + \cdots + C_{m,m} * X24_m \end{cases} \quad (3.21)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận C là ma trận có chức năng như đã mô tả ở S315, khối biến đổi hệ phương trình phi tuyến thuận E1S5,

(S326) mã hoá sử dụng hệ phương trình phi tuyến thuận E2S6 thực hiện *m* phương trình phi tuyến, các phần tử đầu vào của khối E2S6 gồm *m* phần tử của tập X25 và *m* phần tử của tập khóa con K5, X26 được tính như sau:

$$X26 = F(X25, K5) = \{F_i(X25_i + K5_i)\} \text{ với } 1 \leq i \leq m \quad (3.22)$$

biểu diễn toán học của biểu thức (3.22) như sau:

$$\begin{cases} X26_1 = D_{1,1} * (X25_1 + K5_1) + D_{1,2} * (X25_2 + K5_2) + \cdots + D_{1,m} * (X25_m + K5_m) \\ X26_2 = D_{2,1} * (X25_1 + K5_1) + D_{2,2} * (X25_2 + K5_2) + \cdots + D_{2,m} * (X25_m + K5_m) \\ \cdots \\ X26_m = D_{m,1} * (X25_1 + K5_1) + D_{m,2} * (X25_2 + K5_2) + \cdots + D_{m,m} * (X25_m + K5_m) \end{cases} \quad (3.23)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận D là ma trận có chức năng như đã mô tả ở S316, khối biến đổi hệ phương trình phi tuyến thuận E1S6,

(S327) chuyển dạng phi tuyến E2S7 thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số X26 và K6, mỗi tập có *m* phần tử, khối dữ liệu mã hóa Y được tính như sau:

$$Y = Sbox(X26 + K6) = \{Sbox(X26_i + K6_i)\} \text{ với } 1 \leq i \leq m \quad (3.24)$$

trong đó:

phép cộng là phép toán XOR-bit,

Sbox là hàm thay thế để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y có *m* phần tử.

(S4) giải mã văn bản mã hóa thu được từ bước (S3), chia văn bản mã hóa thành các khối dữ liệu Y có độ dài bằng độ dài khóa K0 gồm *m* phần tử, giải mã khối dữ liệu Y bao gồm các bước:

(S401) giải mã khối dữ liệu Y bằng cách sử dụng lần lượt ba tập khóa con K6, K5, K4 để thu được tập kết quả phi tuyến đầu ra Y17 có *m* phần tử,

(S402) giải mã tập kết quả phi tuyến đầu ra Y17 bằng cách sử dụng lần lượt ba tập khóa con K3, K2, K1 để thu được khối dữ liệu thô X có độ dài bằng độ dài của khối dữ liệu mã hoá Y, tập hợp các khối dữ liệu X để được văn bản thô ban đầu;

trong đó, đầu vào của (S402) là kết quả đầu ra của (S401), khối dữ liệu mã hóa Y và các khóa con K1 tới K6 được định dạng thành *n* tập hợp 8-bit (1 byte), $m = n/8$, trong đó:

bước (S401) sử dụng khối dữ liệu mã hoá Y và lần lượt ba tập khóa con K6, K5, và K4, cụ thể là:

(S411) chuyển dạng phi tuyến ngược D1S1: thực hiện *m* hàm chuyển dạng phi tuyến của hai tập biến số Y và K6, mỗi tập có *m* phần tử, Y11 được tính như sau:

$$Y11 = ISbox(Y + K6) = \{ISbox(Y_i + K6_i)\} \text{ với } 1 \leq i \leq m \qquad (4.1)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổ hợp tuyến tính của hai tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y11 có *m* phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải mã;

(S412) biến đổi hệ phương trình phi tuyến ngược D1S2: thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử của tập Y11 để tạo ra một *m* kết quả *m* của tập Y12, Y12 gồm *m* phần tử kết quả, Y12 được tính như sau:

$$Y12 = F(Y11) = \{F_i(Y11_i)\} \text{ với } 1 \leq i \leq m \qquad (4.2)$$

biểu diễn toán học của biểu thức (4.2) như sau:

$$\begin{cases} Y12_1 = E_{1,1} * Y11_1 + E_{1,2} * Y11_2 + \cdots + E_{1,m} * Y11_m \\ Y12_2 = E_{2,1} * Y11_1 + E_{2,2} * Y11_2 + \cdots + E_{2,m} * Y11_m \\ \cdots \\ Y12_m = E_{m,1} * Y11_1 + E_{m,2} * Y11_2 + \cdots + E_{m,m} * Y11_m \end{cases} \qquad (4.3)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulkhai trong trường nhị phân GF (2),

ma trận *E* là một ma trận vuông *m* hàng và *m* cột $E_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận *D* theo quy ước sau:

- hàng thứ *Z* của ma trận *E* bằng hàng thứ *m* của ma trận *D*:
  $E_{Z,i} = D_{m,i}$ với $1 \leq i \leq m$
- các phần tử còn lại của ma trận *E* là các phần tử còn lại của ma trận *D* ở vị trí đối xứng qua đường chéo chính của ma trận *D*:

$E_{k,i} = D_{i,k}$ với $1 \le k \le m, 1 \le i < m, k \ne Z$

(S413) biến đổi hệ phương trình phi tuyến ngược D1S3 thực hiện m phương trình phi tuyến, các phần tử đầu vào của khối D1S3 gồm m phần tử của tập Y12 và m phần tử của tập khóa con K5, Y13 được tính như sau:

$$Y13 = F(Y12, K5) = \{F_i(Y12_i + K5_i)\} \text{ với } 1 \le i \le m \quad (4.4)$$

biểu diễn toán học của biểu thức (4.4) như sau:

$$\begin{cases} Y13_1 = G_{1,1} * (Y12_1 + K5_1) + G_{1,2} * (Y12_2 + K5_2) + \cdots + G_{1,m} * (Y12_m + K5_m) \\ Y13_2 = G_{2,1} * (Y12_1 + K5_1) + G_{2,2} * (Y12_2 + K5_2) + \cdots + G_{2,m} * (Y12_m + K5_m) \\ \cdots \\ Y13_m = G_{m,1} * (Y12_1 + K5_1) + G_{m,2} * (Y12_2 + K5_2) + \cdots + G_{m,m} * (Y12_m + K5_m) \end{cases} \quad (4.5)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulkhai trong trường nhị phân GF (2),

ma trận G là một ma trận vuông m hàng và m cột $G_{m,m}$ với các phần tử nhị phân, ma trận G được nội suy từ ma trận C theo các quy ước sau:

- hàng thứ T của ma trận G có $m - 1$ phần tử là lặp lại hàng thứ m của ma trận C:
  $G_{T,i} = C_{m,i}$ với $1 \le i \le m - 1$
- cột thứ m của ma trận G có $m - 1$ phần tử là lặp lại cột thứ T của ma trận C:
  $G_{k,m} = C_{k,T}$ với $1 \le k \le m, k \ne T$; $G_{T,m} = 0$
- các phần tử còn lại của ma trận G là nghịch đảo các phần tử còn lại của ma trận C ở vị trí đối xứng qua đường chéo chính của ma trận C:
  $G_{k,i} = \bar{C}_{i,k}$ với $1 \le k \le m, 1 \le i \le m - 1, k \ne T$

(S414) chuyển dạng phi tuyến ngược D1S4 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y13 và K4, mỗi tập có m phần tử, Y14 được tính như sau:

$$Y14 = ISbox(Y13) + K4 = \{ISbox(Y13_i) + K4_i\} \text{ với } 1 \le i \le m \quad (4.6)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y14 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải mã;

(S415) biến đổi hệ phương trình phi tuyến ngược D1S5 thực hiện m phương trình, mỗi phương trình xử lý m phần tử phi tuyến của tập Y14 để tạo ra một kết quả của tập Y15, Y15 gồm m phần tử kết quả phi tuyến, Y15 được tính như sau:

$$Y15 = F(Y14) = \{F_i(Y14_i)\} \text{ với } 1 \le i \le m \quad (4.7)$$

biểu diễn toán học của biểu thức (4.7) như sau:

$$\begin{cases} Y15_1 = H_{1,1} * Y14_1 + H_{1,2} * Y14_2 + \cdots + H_{1,m} * Y14_m \\ Y15_2 = H_{2,1} * Y14_1 + H_{2,2} * Y14_2 + \cdots + H_{2,m} * Y14_m \\ \cdots \\ Y15_m = H_{m,1} * Y14_1 + H_{m,2} * Y14_2 + \cdots + H_{m,m} * Y14_m \end{cases} \quad (4.8)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulo hai trong trường nhị phân GF (2),

ma trận H là một ma trận vuông m hàng và m cột $H_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận B theo quy ước sau:

- hàng thứ Z của ma trận H bằng hàng thứ *1* của ma trận B:
  $H_{Z,i} = B_{1,i}$ với $1 \le i \le m$
- các phần tử còn lại của ma trận H là các phần tử còn lại của ma trận B ở vị trí đối xứng qua đường chéo chính của ma trận B:
  $H_{k,i} = B_{i,k}$ với $1 \le k \le m, 1 < i \le m, k \ne Z$

(S416) biến đổi hệ phương trình phi tuyến ngược D1S6 thực hiện m phương trình phi tuyến, các phần tử đầu vào của khối D1S6 gồm m phần tử của tập Y15 và m phần tử của tập khóa con K5, Y16 được tính như sau:

$$Y16 = F(Y15, K5) = \{F_i(Y15_i + K5_i)\} \text{ với } 1 \le i \le m \quad (4.9)$$

biểu diễn toán học của biểu thức (7.9) như sau:

$$\begin{cases} Y16_1 = L_{1,1} * (Y15_1 + K5_1) + L_{1,2} * (Y15_2 + K5_2) + \cdots + L_{1,m} * (Y15_m + K5_m) \\ Y16_2 = L_{2,1} * (Y15_1 + K5_1) + L_{2,2} * (Y15_2 + K5_2) + \cdots + L_{2,m} * (Y15_m + K5_m) \\ \cdots \\ Y16_m = L_{m,1} * (Y15_1 + K5_1) + L_{m,2} * (Y15_2 + K5_2) + \cdots + L_{m,m} * (Y15_m + K5_m) \end{cases} \quad (4.10)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modulo hai trong trường nhị phân GF (2),

ma trận hệ số L là một ma trận vuông m hàng và m cột $L_{m,m}$ với các phần tử nhị phân, ma trận L được nội suy từ ma trận A theo các quy ước sau:

- hàng thứ T của ma trận L có $m-1$ phần tử là lặp lại hàng thứ 1 của ma trận A:
  $L_{T,i} = A_{1,i}$ với $1 < i \leq m$
- cột thứ 1 của ma trận L có $m-1$ phần tử là lặp lại cột thứ T của ma trận A:
  $L_{k,1} = A_{k,T}$ với $1 \leq k \leq m, k \neq T$; $L_{T,1} = 0$
- các phần tử còn lại của ma trận L là nghịch đảo các phần tử còn lại của ma trận A ở vị trí đối xứng qua đường chéo chính của ma trận A:
  $L_{k,i} = \bar{A}_{i,k}$ với $1 \leq k \leq m, 1 < i \leq m, k \neq T$

(S417) chuyển dạng phi tuyến ngược D1S7: thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y16 và K4, mỗi tập có m phần tử, Y17 được tính như sau:

$$Y17 = ISbox(Y16) + K4 = \{ISbox(Y16_i) + K4_i\} \text{ với } 1 \leq i \leq m \qquad (4.11)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y17 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật,

do Y16 là tập hợp m các phần tử phi tuyến, nên $ISbox(Y16) + K4$ cũng là một tập gồm m phần tử phi tuyến,

bước (S402) sử dụng một tập phần tử phi tuyến Y17 kết quả của bước (S401) và lần lượt ba khóa con K3, K2, và K1, cụ thể là:

(S421) chuyển dạng phi tuyến ngược D2S1 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y17 và K3, mỗi tập có m phần tử, Y21 được tính như sau:

$$Y21 = ISbox(Y17 + K3) = \{ISbox(Y17_i + K3_i)\} \text{ với } 1 \leq i \leq m \qquad (4.12)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổ hợp tuyến tính của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y21 có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo mã hoá dữ liệu,

(S422) khối biến đổi hệ phương trình phi tuyến ngược D2S2 thực hiện m phương trình, mỗi phương trình xử lý m phần tử của tập Y21 để tạo ra một kết quả của tập Y22, Y22 gồm m phần tử kết quả, Y22 được tính như sau:

$$Y22 = F(Y21) = \{F_i(Y21_i)\} \text{ với } 1 \le i \le m \quad (4.13)$$

biểu diễn toán học của biểu thức (4.13) như sau:

$$\begin{cases} Y22_1 = E_{1,1} * Y21_1 + E_{1,2} * Y21_2 + \cdots + E_{1,m} * Y21_m \\ Y22_2 = E_{2,1} * Y21_1 + E_{2,2} * Y21_2 + \cdots + E_{2,m} * Y21_m \\ \cdots \\ Y22_m = E_{m,1} * Y21_1 + E_{m,2} * Y21_2 + \cdots + E_{m,m} * Y21_m \end{cases} \quad (4.14)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số E là một ma trận vuông m hàng và m cột $E_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận D như mô tả trong S412, khối biến đổi hệ phương trình phi tuyến ngược D1S2,

(S423) biến đổi hệ phương trình phi tuyến ngược D2S3 thực hiện m phương trình phi tuyến, phần tử đầu vào của khối D2S3 gồm m phần tử của tập Y22 và m phần tử của tập khóa con K2, Y23 được tính như sau:

$$Y23 = F(Y22, K2) = \{F_i(Y22_i + K2_i)\} \text{ với } 1 \le i \le m \quad (4.15)$$

biểu diễn toán học của biểu thức (4.15) như sau:

$$\begin{cases} Y23_1 = G_{1,1} * (Y22_1 + K2_1) + G_{1,2} * (Y22_2 + K2_2) + \cdots + G_{1,m} * (Y22_m + K2_m) \\ Y23_2 = G_{2,1} * (Y22_1 + K2_1) + G_{2,2} * (Y22_2 + K2_2) + \cdots + G_{2,m} * (Y22_m + K2_m) \\ \cdots \\ Y23_m = G_{m,1} * (Y22_1 + K2_1) + G_{m,2} * (Y22_2 + K2_2) + \cdots + G_{m,m} * (Y22_m + K2_m) \end{cases} \quad (4.16)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số G là một ma trận vuông m hàng và m cột $G_{m,m}$ với các phần tử nhị phân. được nội suy từ ma trận C như mô tả trong S413, khối biến đổi hệ phương trình phi tuyến ngược D1S3,

(S424) chuyển dạng phi tuyến ngược D2S4 thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y23 và K1, mỗi tập có m phần tử, Y24 được tính như sau: m

$$Y24 = ISbox(Y23) + K1 = \{ISbox(Y23_i) + K1_i\} \text{ với } 1 \le i \le m \quad (4.17)$$

trong đó:

phép cộng là phép toán XOR-bit,

ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra Y24 có *m* phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật,

(S425) biến đổi hệ phương trình phi tuyến ngược D2S5: thực hiện *m* phương trình, mỗi phương trình xử lý *m* phần tử phi tuyến của tập Y24 để tạo ra một kết quả của tập Y25, Y25 gồm *m* phần tử kết quả phi tuyến, Y25 được tinh như sau:

$$Y25 = F(Y24) = \{F_i(Y24_i)\} \text{ với } 1 \le i \le m \quad (4.18)$$

biểu diễn toán học của biểu thức (4.18) như sau:

$$\begin{cases} Y25_1 = H_{1,1} * Y24_1 + H_{1,2} * Y24_2 + \cdots + H_{1,m} * Y24_m \\ Y25_2 = H_{2,1} * Y24_1 + H_{2,2} * Y24_2 + \cdots + H_{2,m} * Y24_m \\ \cdots \\ Y25_m = H_{m,1} * Y24_1 + H_{m,2} * Y24_2 + \cdots + H_{m,m} * Y24_m \end{cases} \quad (4.19)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán nhân modul hai trong trường nhị phân GF (2),

ma trận hệ số *H* là một ma trận vuông *m* hàng và *m* cột $H_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận *B* như mô tả ở S415, khối biến đổi hệ phương trình phi tuyến ngược D1S5,

(S426) biến đổi hệ phương trình phi tuyến ngược D2S6: thực hiện *m* phương trình phi tuyến, phần tử đầu vào của khối D2S6 gồm *m* phần tử của tập Y25 và *m* phần tử của tập khóa con K2, Y26 được tính như sau:

$$Y26 = F(Y25, K2) = \{F_i(Y25_i + K2_i)\} \text{ với } 1 \le i \le m \quad (5.20)$$

biểu diễn toán học của biểu thức (5.20) như sau:

$$\begin{cases} Y26_1 = L_{1,1} * (Y25_1 + K2_1) + L_{1,2} * (Y25_2 + K2_2) + \cdots + L_{1,m} * (Y25_m + K2_m) \\ Y26_2 = L_{2,1} * (Y25_1 + K2_1) + L_{2,2} * (Y25_2 + K2_2) + \cdots + L_{2,m} * (Y25_m + K2_m) \\ \cdots \\ Y26_m = L_{m,1} * (Y25_1 + K2_1) + L_{m,2} * (Y25_2 + K2_2) + \cdots + L_{m,m} * (Y25_m + K2_m) \end{cases} \quad (4.21)$$

trong đó:

phép cộng là phép toán XOR-bit và phép nhân là phép toán modul hai trong trường nhị phân GF (2),

ma trận hệ số L là một ma trận vuông m hàng và m cột $L_{m,m}$ với các phần tử nhị phân, được nội suy từ ma trận A như mô tả ở S416, khối biến đổi hệ phương trình phi tuyến ngược D1S6,

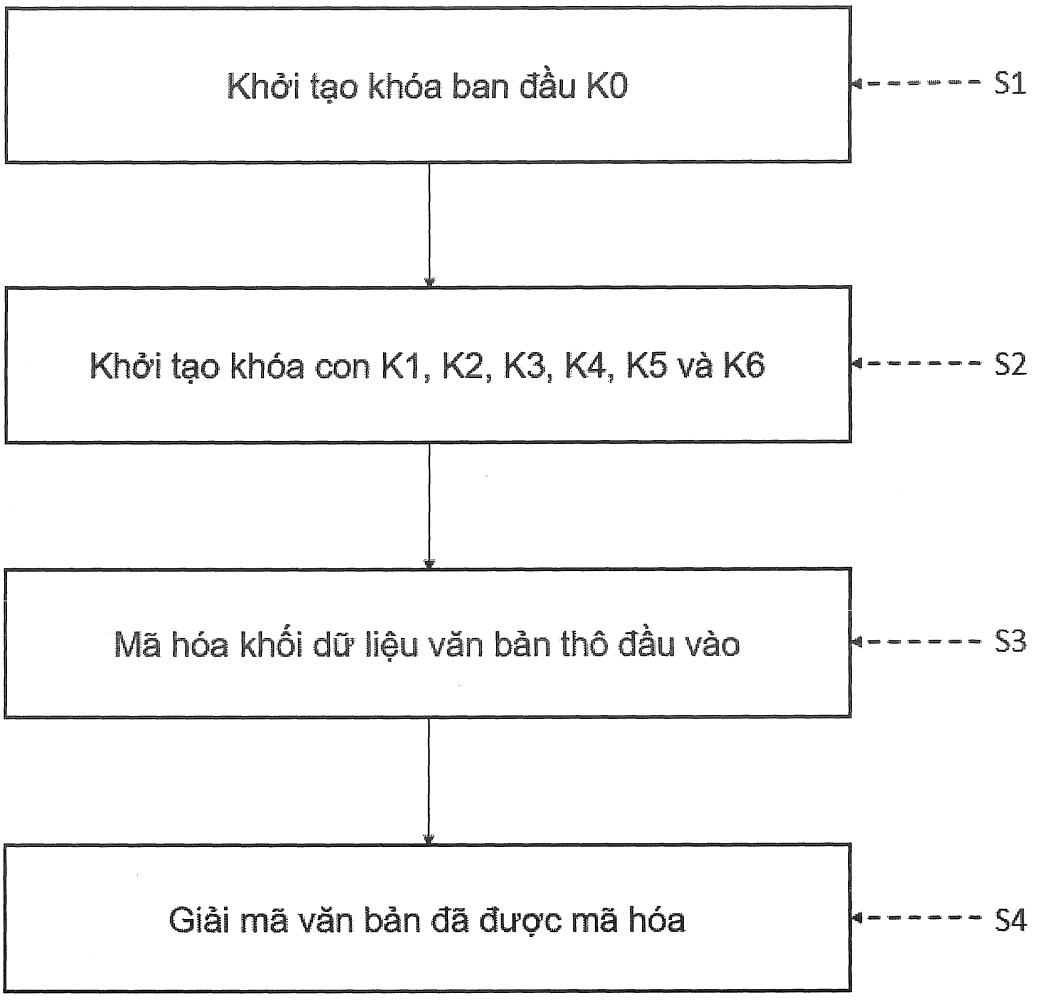
(S427) chuyển dạng phi tuyến ngược D2S7: thực hiện m hàm chuyển dạng phi tuyến của 2 tập biến số Y26 và K1, mỗi tập có m phần tử, khối dữ liệu X được tính như sau:

$$X = ISbox(Y26) + K1 = \{ISbox(Y26_i) + K1_i\} \text{ với } 1 \leq i \leq m \qquad (4.22)$$

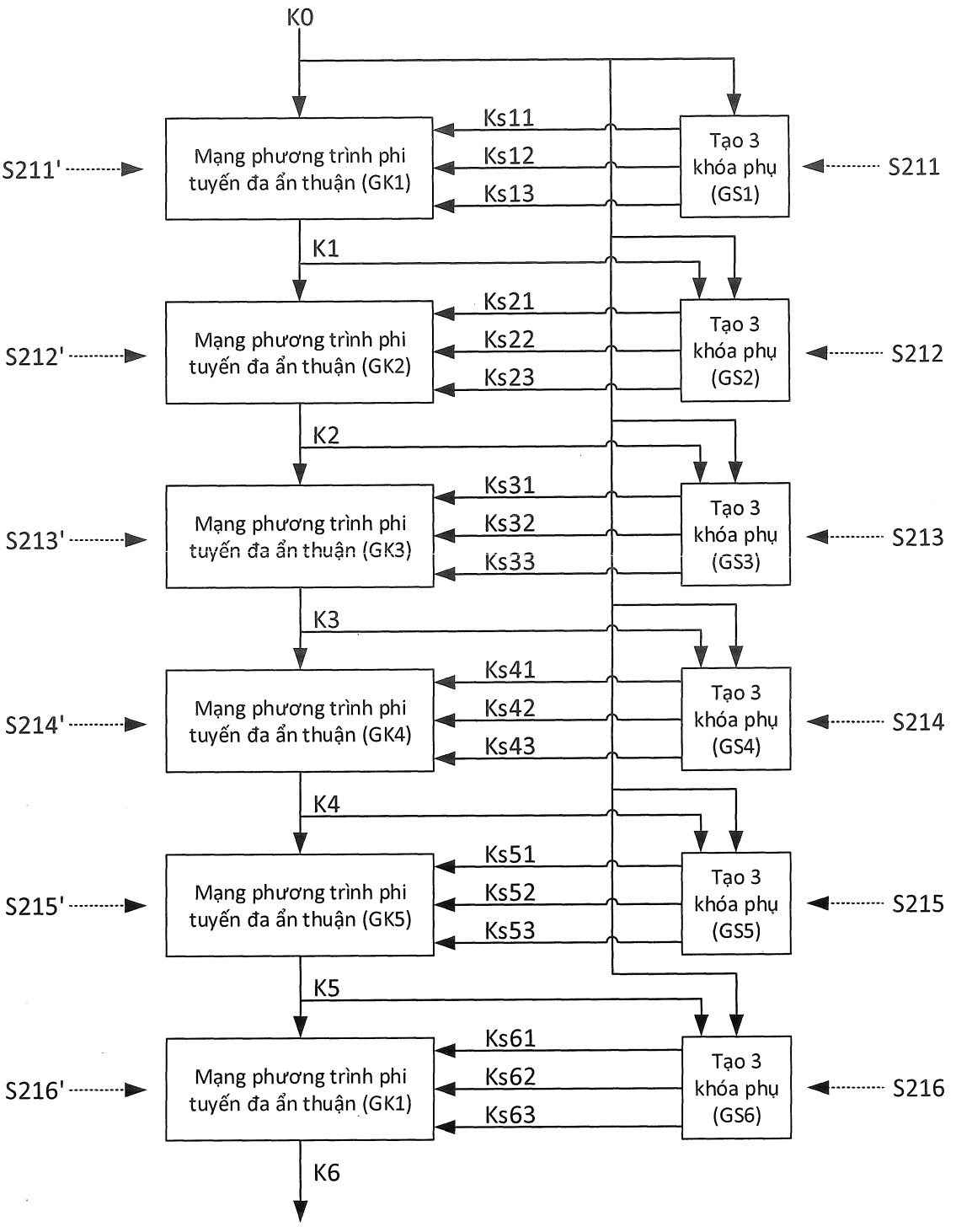
trong đó:

phép cộng là phép toán XOR-bit,

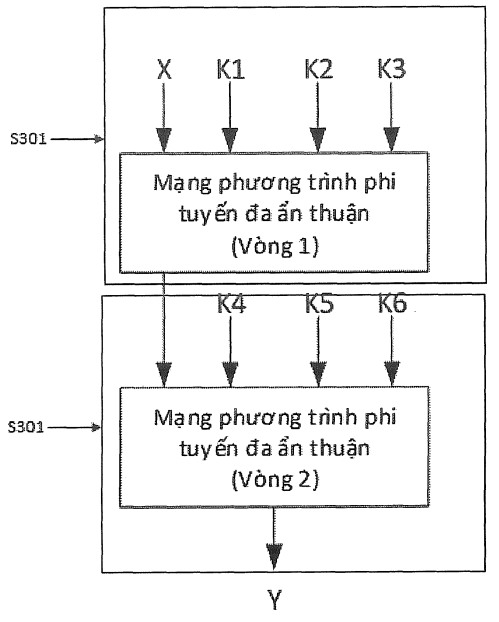
ISbox là hàm thay thế ngược để biến tổng phi tuyến của 2 tập phần tử đầu vào thành một tập kết quả phi tuyến đầu ra X có m phần tử, hàm *ISbox* được nội suy ngược từ hàm *Sbox* khi khởi tạo giải thuật.


Khởi tạo khóa ban đầu K0
S1
Khởi tạo khóa con K1, K2, K3, K4, K5 và K6
S2
Mã hóa khối dữ liệu văn bản thô đầu vào
S3
Giải mã văn bản đã được mã hóa
S4

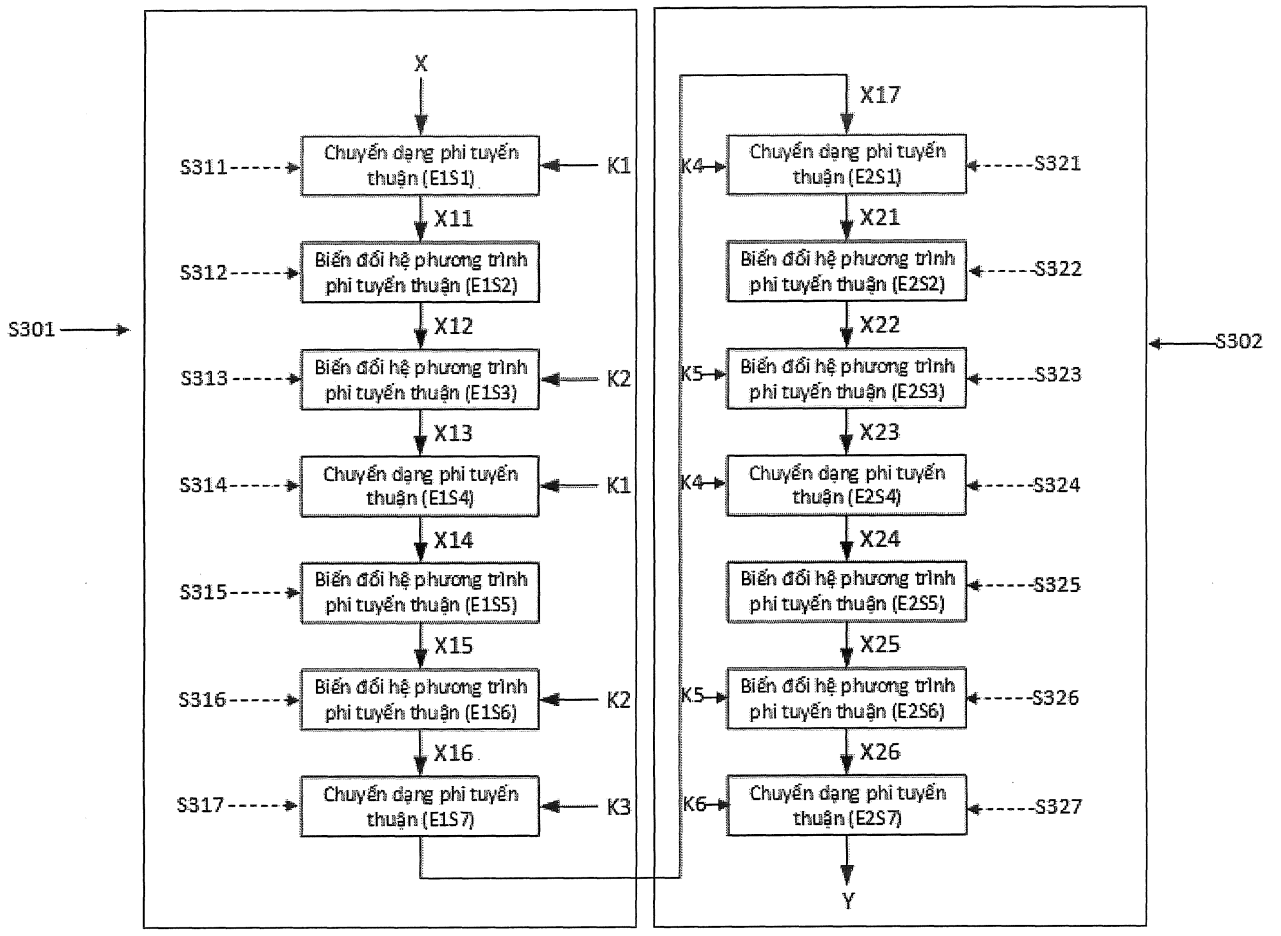
Hình 1


K0
S211'
Mạng phương trình phi tuyến đa ẩn thuận (GK1)
Ks11
Ks12
Ks13
Tạo 3 khóa phụ (GS1)
S211
K1
S212'
Mạng phương trình phi tuyến đa ẩn thuận (GK2)
Ks21
Ks22
Ks23
Tạo 3 khóa phụ (GS2)
S212
K2
S213'
Mạng phương trình phi tuyến đa ẩn thuận (GK3)
Ks31
Ks32
Ks33
Tạo 3 khóa phụ (GS3)
S213
K3
S214'
Mạng phương trình phi tuyến đa ẩn thuận (GK4)
Ks41
Ks42
Ks43
Tạo 3 khóa phụ (GS4)
S214
K4
S215'
Mạng phương trình phi tuyến đa ẩn thuận (GK5)
Ks51
Ks52
Ks53
Tạo 3 khóa phụ (GS5)
S215
K5
S216'
Mạng phương trình phi tuyến đa ẩn thuận (GK1)
Ks61
Ks62
Ks63
Tạo 3 khóa phụ (GS6)
S216
K6

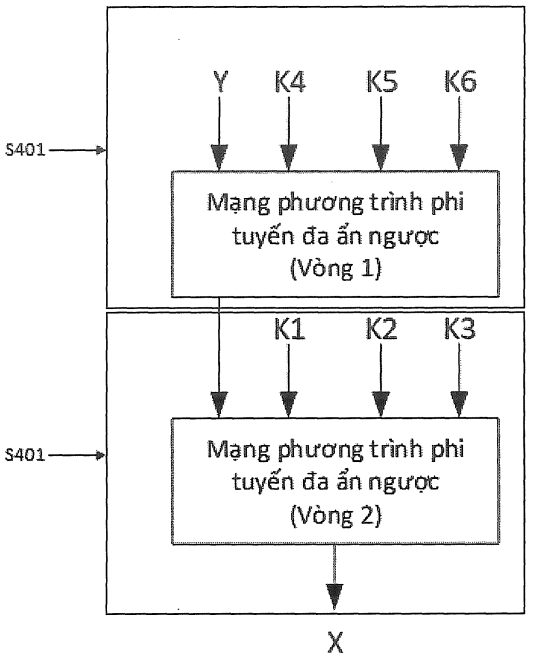
Hình 2



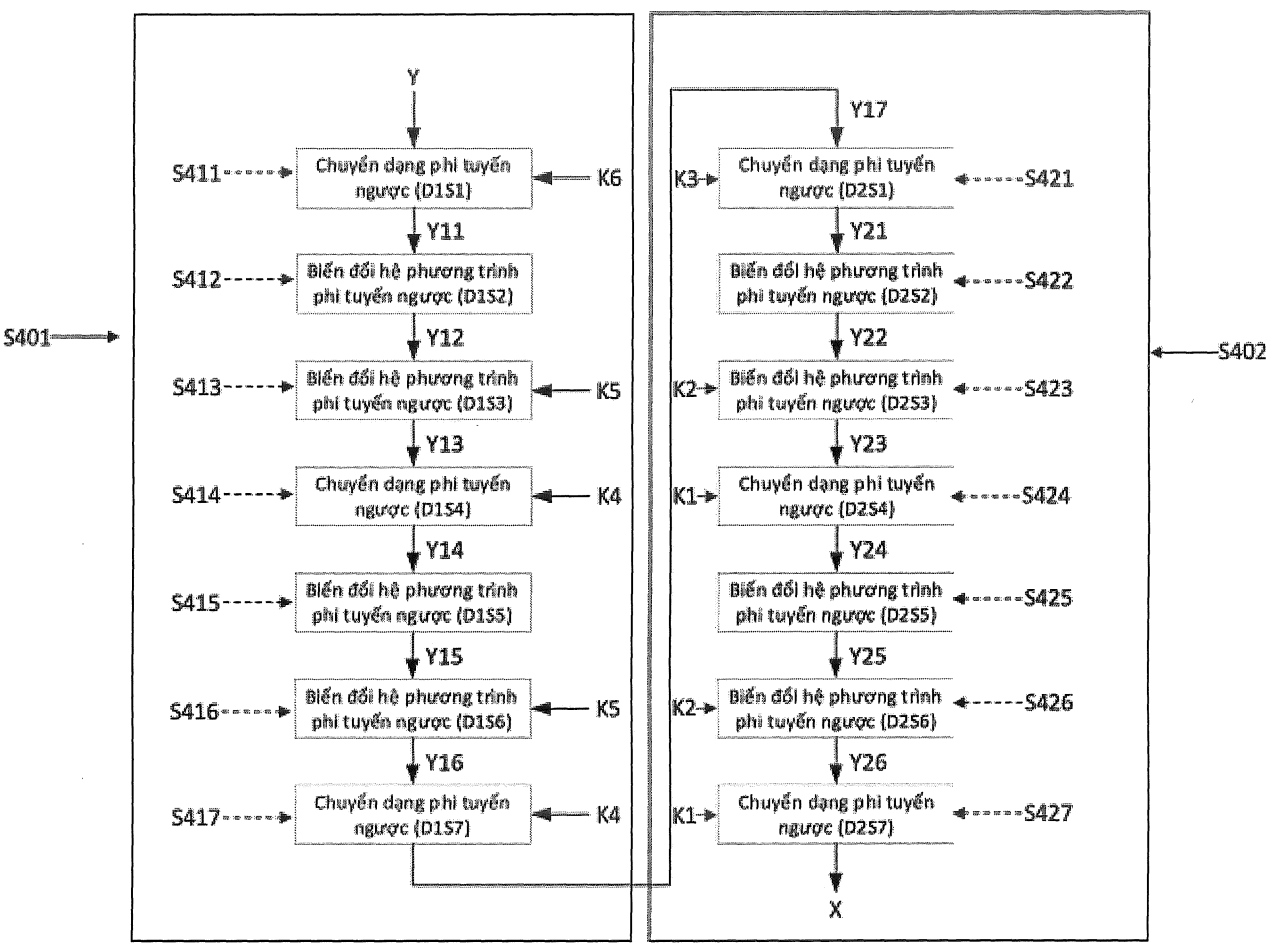
Hình 3a



Hình 3b



Hình 4a



Hình 4b