BAN CƠ YẾU CHÍNH PHỦ

THUYẾT MINH
DỰ THẢO TIÊU CHUẨN VIỆT NAM

CÔNG NGHỆ THÔNG TIN – CÁC KỸ THUẬT AN TOÀN – CÁC THUẬT TOÁN MẬT MÃ –MÃ KHỐI MKV

HÀ NỘI, 2024

Mục Lục

1. Tên gọi và ký hiệu của TCVN

Tiếng Việt: TCVN XXXX:2024 Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Mã khối MKV

Tiếng Anh: TCVN XXXX:2024 Information technology – Security techniques – Encryption algorithms – Block cipher MKV

Ký hiệu tiêu chuẩn: TCVN XXXX:2024

Ghi chú: Khi có quyết định công bố chính thức của Bộ Khoa học – Công nghệ thì tiêu chuẩn này sẽ được gán một số hiệu theo quy tắc ban hành chuẩn hiện hành.

2. Đặt vấn đề

2.1. Đối tượng tiêu chuẩn hoá

Bảo mật và an toàn thông tin, trong đó kỹ thuật mật mã đóng vai trò then chốt, là yếu tố tiên quyết để triển khai các hoạt động giao dịch điện tử. Song song với việc xây dựng hạ tầng kỹ thuật và cơ sở pháp lý, hoạt động xây dựng và ban hành tiêu chuẩn và quy chuẩn kỹ thuật có vai trò cực kỳ quan trọng. Công tác này không chỉ định hướng cho người dùng và đảm bảo cho nhà sản xuất trong việc nâng cao chất lượng, sức cạnh tranh của sản phẩm, hàng hóa và dịch vụ trên thị trường trong nước và quốc tế mà còn góp phần rút ngắn khoảng cách phát triển giữa nước ta và các nước trong lĩnh vực này. Mã khối là một trong các nguyên thủy mật mã quan trọng đảm bảo tính bí mật của thông tin và tham gia nhiều vào các lược đồ/giao thức mật mã. Tại Việt nam đã ban hành một số thuật toán mã khối trong TCVN 11367-3:2016 dựa trên việc chấp nhận nguyên viện chuẩn ISO/IEC 18033-3. Tiêu chuẩn TCVN XXXX:2024 bổ sung thêm một mã khối mới, được gọi tên là MKV, do Ban Cơ yếu Chính phủ xây dựng với mục tiêu đưa ra một chuẩn mã khối riêng của Việt Nam trong lĩnh vực dân sự.

2.2. Tình hình tiêu chuẩn hoá thuật toán mã khối

2.2.1. Vì sao phải tiêu chuẩn hoá mật mã

Cũng giống như tiêu chuẩn hoá trong các lĩnh vực khác, tiêu chuẩn mật mã là yếu tố quan trọng góp phần đảm bảo cho sự hoạt động an toàn của hạ tầng kỹ thuật, của hệ thống thông tin, của dữ liệu của tổ chức, cá nhân. Đối với các nhà thiết kế và sản xuất, tiêu chuẩn mật mã sẽ hỗ trợ để họ có thể cung cấp cho thị trường những sản phẩm mật mã chất lượng cao, phù hợp với các đối tượng sử dụng. Đối với người sử dụng mật mã (tổ chức, cá nhân) hai ưu thế lớn nhất trong sử dụng mật mã là đảm bảo tính an toàn và tính liên thông. Thứ nhất, người sử dụng có thể tin tưởng là các sản phẩm dựa trên tiêu chuẩn hoá là các sản phẩm an toàn vì chúng đã được một số lượng lớn các chuyên gia kiểm định, đã được thử thách trên thực tế và đã được nhiều tổ chức chấp nhận.Thứ hai nếu các sản phẩm dựa trên các tiêu chuẩn thì dù chúng được sản xuất bởi các tổ chức khác nhau hoặc được thiết kế để vận hành trên các nền tảng khác nhau vẫn giao tác được với nhau. Trên thế giới, không phải quốc gia nào cũng có chuẩn mã hoá của

riêng mình, chỉ có một số ít các nước đã nghiên cứu ban hành chuẩn mã hoá cho lĩnh vực dân sự như Mỹ, Liên bang Nga, Trung Quốc, Hàn Quốc, Nhật Bản, Belarus, ... Một nhà mật mã học nổi tiếng người Nga đã nói "*Một quốc gia được coi là vĩ đại nếu quốc gia đó sở hữu vũ khí hạt nhân, làm chủ không gian và có chuẩn mật mã của riêng mình*". Do vậy, việc sở hữu riêng chuẩn mật mã là một trong những yếu tố khẳng định tính tự chủ của Việt Nam trong công cuộc bảo vệ chủ quyền Quốc gia.

2.2.2. Tình hình tiêu chuẩn hoá trong nước ngoài nước và trong nước

Ngoài nước. Thuật toán mã hoá có thể coi là "cốt lõi của cốt lõi", là trái tim của các sản phẩm bảo mật, an toàn thông tin, muốn làm chủ được công nghệ phải làm chủ được thuật toán mật mã. Các chuẩn mã khối trên thế giới thường được công bố rộng rãi và được nhiều quốc gia, tổ chức mật mã uy tín khuyến cáo sử dụng.

Phạm vi quốc tế:

- Đối với tổ chức ISO/IEC, ta có
  - Chuẩn *ISO/IEC 18033-3*: Gồm các mã khối có kích thước 64-bit là DEA (triple DES)**,** MISTY1**,** CAST-128**,** HIGHT; các mã khối có kich thước 128-bit là các mã khối AES, Camellia, SEED, SM4 và Kuznyechik.
  - Chuẩn mã khối hạng nhẹ ISO/IEC 29192-2: PRESENT, CLEFIA.
- Đối với IETF ta có khuyến cáo như RFC 2040: RC5, RFC 2144: CAST, RFC 2944: MISTY1, RFC 3713: Camellia, RFC 4009: SEED, RFC 5794: ARIA, RFC 7801: Kuznyechik, ….

Phạm vi các quốc gia: Một số quốc gia tiêu biểu có chuẩn mã khối riêng gồm:

- Belarus: có chuẩn mã khối Bel-T.
- Trung quốc: có chuẩn mã khối GB/T 32905-2016: SMS4.
- Ucraina: có chuẩn DSTU 7624: Kalyna
- Liên Bang Nga: có chuẩn mã khối GOST R 34.11-2015: Kuznyechik, Magma.
- Hàn Quốc: có chuẩn KS X 1213:2004: ARIA, KS X 3246: LEA, TTAS.KO-12.0004: SEED, TTAS.KO-12.0040: HIGHT.
- Mỹ: có chuẩn FIPS 185: Skipjack, FIPS 197: AES.
- Nhật bản: Mặc dù Nhật Bản không có các thuật toán mật mã được tiêu chuẩn hóa. Tuy nhiên, CRYPTREC duy trì một danh sách "thuật toán được khuyến cáo" chứa các mã khối sau: AES, Camellia, trong đó Camellia là thuật toán được các nhà khoa học của Nhật phát triển.

Bên cạnh đó, có quốc gia sử dụng các khuyến cáo: Một số quốc gia không có chuẩn mã khối riêng hoặc định mức cho các thuật toán mật mã và thay vào đó chọn công bố danh sách các thuật toán "được khuyến cáo" như: Canada: Trung tâm an ninh mạng Canada đưa ra một số khuyến cáo cho bộ mật mã TLS đầy đủ, trong đó mã khối được khuyến cáo là AES; Malaysia: Dự án MySEAL liệt kê một số nguyên tắc mật mã được coi là an toàn cùng với lý do đằng sau việc đưa vào (không) các thuật toán khác

nhau; Nhiều quốc gia cũng đã khuyến cáo sử dụng các tiêu chuẩn trong ISO/IEC 18033-2 như Pháp, Đức, Anh, Isarel, …

Trong nước. Một trong những mục tiêu của Chiến lược "Make in Vietnam" do Thủ tướng Chính phủ ban hành là phát triển kinh tế số chiếm 20% GDP, với việc xác định các bước tiến đột phá mang tính hệ thống, nhấn mạnh vào chuyển đổi chủ quyền công nghệ, có sự chuyển dịch mạnh mẽ từ lắp ráp, gia công sang sáng tạo, thiết kế một cách chủ động, tạo ra các sản phẩm công nghệ "Make in Vietnam".

Trước bối cảnh lịch sử của sự chuyển mình trong xu thế phát triển của nền kinh tế số, bên cạnh việc thực hiện các nhiệm vụ chính trị quan trọng được Đảng và Nhà nước giao, với vai trò là Cơ quan mật mã quốc gia, Ban Cơ yếu Chính phủ đã chỉ đạo tổ chức nghiên cứu, xây dựng thuật toán mã khối dân sự để thiết kế, chế tạo các sản phẩm bảo mật, an toàn thông tin phục vụ phát triển kinh tế số, xã hội số.

Ở một khía cạnh khác, khi mà bài toán bảo mật thông tin có sức nóng hơn bao giờ hết trước sự phát triển không ngừng của công nghệ lượng tử và các vấn đề đảm bảo an toàn thông tin trước những tấn công thám mã dựa trên tính toán lượng tử. Ban Cơ yếu Chính phủ đặt mục tiêu quan trọng là phải có một thuật toán mã hoá "Make in Vietnam" không chỉ an toàn lượng tử mà còn đảm bảo hiệu năng cần thiết để đáp ứng nhu sử dụng trong thời đại số. Ngoài ra, thuật toán không chỉ đảm bảo an toàn và hiệu quả sử dụng, phải có đặc trưng riêng, có cấu trúc riêng so với các chuẩn khác trên thế giới.

Hiện nay, trong lĩnh vực dân sự, Ban Cơ yếu Chính phủ đã xây dựng và đề xuất Bộ Khoa học và Công nghệ công bố được 55 tiêu chuẩn quốc gia trong lĩnh vực mật mã bao gồm:

TT	Ký hiệu	Tên tiêu chuẩn	Cơ quan đề xuất
1	TCVN 7635:2007	Công nghệ thông tin – Kỹ thuật mật mã – Chữ ký số	Ban Cơ yếu Chính phủ
2	TCVN 7816:2007	Công nghệ thông tin – Kỹ thuật mật mã thuật toán mã dữ liệu AES (Phiên bản mới nhất TCVN 11367-3:2016)	Ban Cơ yếu Chính phủ
3	TCVN 7817-1:2007	Công nghệ thông tin – Kỹ thuật mật mã quản lý khóa – Phần 1: Khung tổng quát (Phiên bản mới nhất ISO/IEC 11770–1:2010)	Ban Cơ yếu Chính phủ
4	TCVN 7817-2:2007	Công nghệ thông tin – Kỹ thuật mật mã quản lý khóa – Phần 2: Cơ chế sử dụng kỹ thuật đối xứng (Phiên bản mới	Ban Cơ yếu Chính phủ

		nhất ISO/IEC 11770-2:2018)	
5	TCVN 7817-3:2007	Công nghệ thông tin – Kỹ thuật mật mã quản lý khóa – Phần 3: Các cơ chế sử dụng kỹ thuật không đối xứng (Phiên bản mới nhất ISO/IEC 11770-3:2021)	Ban Cơ yếu Chính phủ
6	TCVN 7817-4:2007	Công nghệ thông tin – Kỹ thuật mật mã quản lý khóa – Phần 4: Cơ chế dựa trên bí mật yếu (Phiên bản mới nhất ISO/IEC 11770-4:2017)	Ban Cơ yếu Chính phủ
7	TCVN 7818-1:2007	Công nghệ thông tin – Kỹ thuật mật mã dịch vụ tem thời gian – Phần 1: Khung tổng quát	Ban Cơ yếu Chính phủ
8	TCVN 7818-2:2007	Công nghệ thông tin – Kỹ thuật mật mã dịch vụ tem thời gian – Phần 2: Cơ chế token độc lập	Ban Cơ yếu Chính phủ
9	TCVN 7818-3:2007	Công nghệ thông tin – Kỹ thuật mật mã dịch vụ tem thời gian – Phần 3: Cơ chế tạo thẻ liên kết	Ban Cơ yếu Chính phủ
10	TCVN 11295:2016	Công nghệ thông tin – Các kỹ thuật an toàn – Yêu cầu an toàn cho mô-đun mật mã	Ban Cơ yếu Chính phủ
11	TCVN 11367-1:2016	Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 1: Tổng quan (Phiên bản mới nhất ISO/IEC 18033-1:2021)	Ban Cơ yếu Chính phủ
12	TCVN 11367-2:2016	Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 2: Mật mã phi đối xứng	Ban Cơ yếu Chính phủ
13	TCVN 11367-3:2016	Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 3: Mã khối	Ban Cơ yếu Chính phủ
14	TCVN 11367-4:2016	Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 4: Mã dòng	Ban Cơ yếu Chính phủ
15	TCVN 11816-1:2017	Công nghệ thông tin – Các kỹ thuật	Ban Cơ yếu

		an toàn – Hàm băm – Phần 1: Tổng quan	Chính phủ
16	TCVN 11816-2:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Hàm băm – Phần 2: Hàm băm sử dụng mã khối n-bit.	Ban Cơ yếu Chính phủ
17	TCVN 11816-3:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Hàm băm – Phần 3: Hàm băm chuyên dụng	Ban Cơ yếu Chính phủ
18	TCVN 11816-4:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Hàm băm – Phần 4: Hàm băm sử dụng số học đồng dư	Ban Cơ yếu Chính phủ
19	TCVN 11817-1:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Xác thực thực thể – Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
20	TCVN 11817-2:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Xác thực thực thể – Phần 2: Cơ chế sử dụng thuật toán mã hóa đối xứng	Ban Cơ yếu Chính phủ
21	TCVN 11817-3:2017	Công nghệ thông tin – Các kỹ thuật an toàn – Xác thực thực thể – Phần 1: Cơ chế sử dụng kỹ thuật chữ ký số	Ban Cơ yếu Chính phủ
22	TCVN 12214-1:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Chữ ký số kèm phụ lục - Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
23	TCVN 12214-2:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Chữ ký số kèm phụ lục - Phần 2: Các cơ chế dựa trên phân tích số nguyên	Ban Cơ yếu Chính phủ
24	TCVN 12214-3:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Chữ ký số kèm phụ lục - Phần 3: Các cơ chế dựa trên logarit rời rạc	Ban Cơ yếu Chính phủ
25	TCVN 11367-5:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Thuật toán mật mã - Phần 5: Mật mã dựa trên định danh	Ban Cơ yếu Chính phủ
26	TCVN 12211:2018	Công nghệ thông tin - Các kỹ thuật	Ban Cơ yếu

		an toàn - Yêu cầu kiểm thử cho mô đun mật mã	Chính phủ
27	TCVN 12212:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Phương pháp kiểm thử giảm thiểu các lớp tấn công không xâm lấn chống lại các mô đun mật mã	Ban Cơ yếu Chính phủ
28	TCVN 12213:2018	Công nghệ thông tin - Các kỹ thuật an toàn - Chế độ hoạt động cho mã khối n-bit	Ban Cơ yếu Chính phủ
29	TCVN 12852-1:2020	Công nghệ thông tin – Kỹ thuật an toàn – Kỹ thuật mật mã dựa trên đường cong elliptic – Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
30	TCVN 12852-5:2020	Công nghệ thông tin – Kỹ thuật an toàn – Kỹ thuật mật mã dựa trên đường cong elliptic – Phần 5: Các kỹ thuật tạo đường cong elliptic	Ban Cơ yếu Chính phủ
31	TCVN 12853:2020	Công nghệ thông tin – Kỹ thuật an toàn – Bộ tạo bit ngẫu nhiên	Ban Cơ yếu Chính phủ
32	TCVN 12855-2:2020	Công nghệ thông tin – Kỹ thuật an toàn – Lược đồ chữ ký số có khôi phục thông điệp – Phần 2: Các cơ chế dựa trên phân tích số nguyên	Ban Cơ yếu Chính phủ
33	TCVN 12855-3:2020	Công nghệ thông tin – Kỹ thuật an toàn – Lược đồ chữ ký số có khôi phục thông điệp – Phần 3: Các cơ chế dựa trên bài toán Logarit rời rạc	Ban Cơ yếu Chính phủ
34	TCVN 12854-1:2020	Công nghệ thông tin – Kỹ thuật an toàn – Mật mã hạng nhẹ -Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
35	TCVN 12854-2:2020	Công nghệ thông tin – Kỹ thuật an toàn – Mật mã hạng nhẹ - Phần 2: Mã khối	Ban Cơ yếu Chính phủ
36	TCVN 12854-3:2020	Công nghệ thông tin – Kỹ thuật an toàn – Mật mã hạng nhẹ - Phần 3:	Ban Cơ yếu Chính phủ

		Mã dòng	
37	TCVN 12854-4:2020	Công nghệ thông tin – Kỹ thuật an toàn – Mật mã hạng nhẹ - Phần 4: Cơ chế sử dụng kỹ thuật phi đối xứng	Ban Cơ yếu Chính phủ
38	TCVN 11817-4:2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể - Phần 4: Cơ chế sử dụng hàm kiểm tra mật mã	Ban Cơ yếu Chính phủ
39	TCVN 11817-5:2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể - Phần 5: Cơ chế sử dụng kỹ thuật tri thức không	Ban Cơ yếu Chính phủ
40	TCVN 11817-6:2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể - Phần 6: Cơ chế sử dụng truyền dữ liệu thủ công	Ban Cơ yếu Chính phủ
41	TCVN 13175:2020	Công nghệ thông tin – Các kỹ thuật an toàn – Mã hóa ký	Ban Cơ yếu Chính phủ
42	TCVN 12854-5: 2020	Công nghệ thông tin – Kỹ thuật an toàn – Mật mã hạng nhẹ – Phần 5: Các hàm băm	Ban Cơ yếu Chính phủ
43	TCVN 13176:2020	Công nghệ thông tin – Kỹ thuật an toàn – Bộ tạo số nguyên tố	Ban Cơ yếu Chính phủ
44	TCVN 13177:2020	Công nghệ thông tin – Kỹ thuật an toàn – Các thuật toán mật mã và kiểm thử phù hợp các cơ chế an toàn	Ban Cơ yếu Chính phủ
45	TCVN 7817-5:2020	Công nghệ thông tin – Kỹ thuật an toàn – Quản lý khóa - Phần 5: Nhóm quản lý khóa	Ban Cơ yếu Chính phủ
46	TCVN 13178-1: 2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể ẩn danh - Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
47	TCVN 13178-2: 2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể ẩn danh - Phần 2: Các cơ chế dựa trên chữ ký	Ban Cơ yếu Chính phủ

		sử dụng một nhóm khóa công khai	
48	TCVN 13178-4: 2020	Công nghệ thông tin – Kỹ thuật an toàn – Xác thực thực thể ẩn danh - Phần 4: Các cơ chế dựa trên bí mật yếu	Ban Cơ yếu Chính phủ
49	TCVN 11367-6:2022	Công nghệ thông tin – Các kỹ thuật an toàn – Thuật toán mật mã – Phần 6: Mã hóa đồng cấu	Ban Cơ yếu Chính phủ
50	TCVN 13460-1:2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số mù – Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
51	TCVN 13460-2:2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số mù – Phần 2: Các cơ chế dựa trên logarit rời rạc	Ban Cơ yếu Chính phủ
52	TCVN 13461-1: 2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số ẩn danh – Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
53	TCVN 13461-2: 2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chữ ký số ẩn danh – Phần 2: Các cơ chế sử dụng một khóa công khai nhóm	Ban Cơ yếu Chính phủ
54	TCVN 13462-1:2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chia sẻ bí mật – Phần 1: Tổng quan	Ban Cơ yếu Chính phủ
55	TCVN 13462-2:2022	Công nghệ thông tin – Các kỹ thuật an toàn – Chia sẻ bí mật – Phần 2: Các cơ chế cơ bản	Ban Cơ yếu Chính phủ

Các chuẩn được chấp thuận nguyên vẹn theo các chuẩn của ISO/IEC bởi vì Việt Nam là thành viên của tổ chức ISO/IEC. Tuy nhiên, để tạo sự thống nhất, thông suốt trong vấn đề bảo mật thông tin trong các hệ thống khác nhau, và khẳng định tính tự chủ về mật mã, việc xây dựng chuẩn riêng khẳng định tư tưởng và xu thế "Make in Vietnam" càng thật sự cần thiết.

3. Sở cứ xây dựng các yêu cầu kỹ thuật

3.1. Tổng hợp, phân tích các tiêu chuẩn quốc tế, tài liệu kỹ thuật, các kết quả nghiên cứu liên quan tới mã khối

3.1.1. Khảo sát thiết kế và độ an toàn của các mã khối tiêu biểu

Sau đây là một số khảo sát về nguyên lý thiết kế và kết quả độ an toàn của các mã khối tiêu biểu được chuẩn hóa và khuyến cáo trên thế giới:

Bảng 1 - Tổng hợp một số thiết kế cho một số chuẩn mã khối tiêu biểu

Mã khối (Tiêu chuẩn, Khuyến cáo	**Đặc điểm thiết kế**	**Độ an toàn và một số lưu ý**
AES ISO/IEC 18033-3 FIPS 197	**Các phiên bản kích thước khối/độ dài khóa/số vòng:** 128/128/10 128/192/12 128/256/14 **Cấu trúc:** mạng SPN. **Hàm vòng:** Sử dụng 3 phép biến đổi cơ bản là SubByte, MixColumns, ShiftRows để tạo tính xáo trộn và khuếch tán. *Xáo trộn:* Biến đổi phi tuyến SubByte dựa trên S-hộp 8-bit, được affine từ ánh xạ nghịch đảo x^{-1} trên trường $GF(2^8)$. *Khuếch tán:* Biến đổi tuyến tính MixColumn dựa trên các ma trận MDS có kích thước 4×4 trên trường $GF(2^8)$; Biến đổi ShiftRows chuyển vị các byte, có tính khuếch tán đều theo. **Lược đồ khoá:** Sử dụng các biến đổi cơ bản trong hàm vòng, tuy nhiên chỉ một lần lặp lại nên các mối quan hệ giữa các vòng vẫn còn, dễ dẫn đến tấn công khóa quan hệ.	Thuật toán chiến thắng trong cuộc thi tuyển chọn AES, được thiết kế bởi Joan Daemen và Vicent Rijmen (tên ban đầu là Rijndael [65]). Độ an toàn chứng minh được trước thám mã vi sai và tuyến tính, suy ra từ chiến lược vệt lan rộng. Một thám mã có tên "Biclique" [66] là một tấn công khôi phục khóa đầu tiên lên AES-128, AES-192, AES-256 đầy đủ số vòng với độ phức tạp tính toán tốt hơn vét cạn, cụ thể tương ứng $2^{126,2}$, $2^{189,4}$ và $2^{254,4}$. Trước đó, không có tấn công nào có thể phá vỡ hơn 7 vòng của bất kỳ phiên bản nào của AES, bài báo [68] là tấn công đầu tiên được xem xét phá vỡ bảy vòng của AES-128 và 8 vòng của AES-192, còn [70] tấn công lên 8 vòng của AES-256 với độ phức tạp thời gian thấp hơn so với tấn công vét cạn. Tấn công với độ phức tạp thấp nhất là tấn công trong mô hình khóa quan hệ [78], [84]. Các tấn công này được giả sử kẻ tấn công có thể tác động được vào khóa chính nhằm khai thác các quan hệ có được với khóa chính, ví dụ có thể lật bit khóa bằng cách tiêm lỗi. Tuy nhiên, độ phức tạp của tấn công kiểu này sẽ tăng lên rất nhanh khi số lượng bit được xem xét trong quan hệ có được với khóa chính là không tầm thường tăng.
Camellia ISO/IEC 18033-3 RFC 3713	**Các phiên bản kích thước khối/độ dài khóa/số vòng:** 128/128/10 128/192/12 128/256/14 **Cấu trúc:** Feistel **Hàm vòng:** Tầng phi tuyến sử dụng S-hộp 8-bit khác nhau, mỗi cái được sử dụng lặp	Tấn công tốt nhất lên Camellia với số vòng được rút gọn được công bố cho tới này là các tấn công hình vuông và hình chữ nhật (tức là thuộc nhóm các tấn công tích phân và tấn công boomerang). Tấn công hình vuông lên chín vòng của Camellia được giới thiệu bởi Yongjin Yeom, Sangwoo Park và IlJun Kim yêu cầu 2^{61} bản rõ

	lại hai lần. Tầng tuyến tính sử dụng ma trận nhị phân có số nhánh tối ưu. Một tầng biến đổi logic "FL-hàm" và nghịch đảo của nó (FL^{-1}-hàm) được áp dụng cho các nhánh cấu trúc Feistel. **Lược đồ khoá**: Sử dụng luôn hàm mã. Sau khi tổ hợp một vài khóa được từ một phần khóa bí mật, dữ liệu này được mã hóa bởi hai vòng mã hóa với các hằng số được xác định trước được coi là khóa: hằng số thứ i là biểu diễn nhị phân của căn bậc hai của số nguyên tố thứ i. Từ đầu ra của phép mã hóa rút gọn này, tất cả khóa con nhận được bằng cách sử dụng dịch vòng bit và trích bit.	được lựa chọn và một lượng lớn 2^{202} phép mã hóa. Tấn công hình chữ nhật được đề xuất bởi Taizo Shirai [98] đã phá vỡ 10 vòng với 2^{127} bản rõ được lựa chọn và yêu cầu 2^{241} truy cập bộ nhớ. Mã pháp này có thể đạt được hiệu năng rất tốt và dường như có lợi thế lớn khi cài đặt cứng hóa tối ưu nếu cần. Với cấu trúc Feistel, phép mã hóa và giải mã sử dụng chung một mạch và chỉ cần thêm một số chi phí hỗ trợ để có thể cài đặt cả hai. Lược đồ khóa có thể một phần chồng chéo với giải mã, song nó không thể cài đặt song song hóa hoàn toàn như quá trình giải mã.
SEED ISO/IEC 18033-3	**Kích cỡ khối/Độ dài khóa/Số vòng:** 128/128/16 **Cấu trúc:** Feistel lồng nhau (nested Feistel) **Hàm vòng:** Chia đầu vào của nó thành hai nửa được XOR với một khóa con, sau đó đi vào một mã pháp nhỏ 3 tầng sử dụng một hàm G và phép cộng modulo 2^{32}, trong hàm G có các biến đổi sau: Biến đổi phi tuyến sử dụng hai S-hộp 8-bit là các hoán vị x^{247} và x^{251} trong $GF(2^8)$, có thể biểu diễn dưới dạng affine hoá của ánh xạ nghịch đảo x^{-1}. Biến đổi tuyến tính có ma trận biểu diễn dạng dịch vòng trên trường $GF(2^8)$. **Lược đồ khoá:** Sử dụng hàm G và phép dịch vòng.	Tấn công tốt nhất lên SEED được công bố là tấn công vi sai lên 8-vòng: Trong năm 2011, Sung mô tả một vi sai 7-vòng với xác suất 2^{-122} đối với SEED (xem [9]), bằng cách tổng hợp xác suất của rất nhiều đặc trưng vi sai 7-vòng với cùng sai khác đầu vào-đầu ra, và cuối cùng đưa ra tấn công vi sai lên 8-vòng của SEED, trong bài bào này Sung cũng đã mô tả một vi sai 8-vòng với xác suất 2^{-124} lên SEED.
Kuznyechik GOST R 34.12	**Kích cỡ khối/Độ dài khóa/Số vòng:** 128/256/9 **Cấu trúc:** SPN (dạng AES) **Hàm vòng:** gồm hai biến đổi cơ bản: Biến đổi tuyến tính sử dụng một ma trận MDS có kích thước 16×16 trên trường $GF(2^8)$ có dạng ma trận đồng hành để hướng cài đặt hiệu quả phần cứng. Biến đổi phi tuyến sử dụng S-hộp có kích thước 8-bit có độ phi tuyến	Nguyên lý thiết kế của Kuznyechik được ít các tài liệu đề cập tới và không được đề cập trong chuẩn chính thức. Một điểm đáng chú ý trong thiết kế là ma trận MDS được sử dụng trong tầng tuyến tính dựa trên một ma trận đồng hành để có thể cài đặt hiệu quả phần cứng. Các S-hộp 8-bit trong tầng phi tuyến được sinh ngẫu nhiên song một số tấn công dịch ngược (reverse-engineering) đã chỉ ra rằng

	không cao bằng S-hộp của mã khối AES, được tác giả khẳng định sử dụng sinh ngẫu nhiên. **Lược đồ khoá:** Sử dụng cấu trúc Feistel với hàm vòng chính là hàm vòng của bước mã hoá có đầu vào khoá là các hằng số để tránh tấn công trượt và có số vòng lặp đủ lớn để chống tấn công vi sai khóa quan hệ.	nó có thể được phân rã thành một mạng cấu trúc ẩn đối với các S-hộp có số chiều nhỏ hơn (xem [10]).Tuy nhiên, những kết quả này hoàn toàn không ảnh hưởng đến độ an toàn của Kuznyechik, mà những kết quả này có ý nghĩa rất lớn trong việc tăng tốc độ trong thực thi phần cứng của thuật toán này. Tấn công gặp giữa lên 5 vòng là tấn công tốt nhất được biết lên mã khối Kuznyechi: tấn công này cho phép khôi phục khoá với độ phức tạp thời gian 2^{140}, độ phức tạp bộ nhớ 2^{153}, và độ phức tạp dữ liệu 2^{113}. Sau đó, nhóm này [11] đã công bố hai tấn công gây lỗi lên Kuznyechik từ đó chỉ ra sự quan trọng trong việc bảo vệ cài đặt của mã pháp. Cụ thể các tác giả đã trình bày 2 tấn công phân tích lỗi lên Kuznyechik trong hai cài đặt khác nhau.
Kalyna DSTU 7624	**Các phiên bản kích thước khối/độ dài khóa/số vòng:** 128/128/10 128/256/14 256/256/14 256/512/18 512/512/18 **Cấu trúc:** Sử dụng cấu trúc SPN, với thiết kế dạng thuật toán AES. **Hàm vòng:** gồm ba biến đổi tương tự như thuật toán AES: Biến đổi phi tuyến SubByte sử dụng các S-hộp 8-bit. Biến đổi tuyến tính MixColumn có kích thước 8×8 trên trường $GF(2^8)$. Một phép chuyển vị byte, đảm bảo tính khuếch tán đều. Mã khối sử dụng phép cộng khoá làm trắng trước và sau là phép cộng modulo 2^{64}. **Lược đồ khoá:** Khá phức tạp sử dụng các phép biến đổi của hàm vòng được lặp lại hai lần với các chỉ số chẵn và chỉ số lẻ khác nhau.	Mã khối Kalyna được đề xuất bởi Oliynykov cùng cộng sự và đã được lựa chọn là chuẩn mã khối của Ucraina vào năm 2015, có định hướng thiết kế trên các từ 64-bit. Có độ an toàn được suy ra hoàn toàn từ chiến lược vệt lan rộng. Trong [12], AlTawy và cộng sự đã giới thiệu chi tiết một tấn công khôi phục khóa đầu tiên chống lại chuẩn Kalyna-128/256 và Kalyna-256/512. Các tác giả đã áp dụng tấn công gặp ở giữa MITM để phá vỡ 7-vòng của Kalya và chỉ ra đây là tấn công tốt nhất tại thời điểm đó. Sau đó, một tấn công trong mô hình một khóa được mở rộng bởi nhóm tác giả Akshima đưa ra lên 9 vòng của Kalyna (xem [13]). Năm 2018, nhóm tác giả Li Lin đã giới thiệu một số kết quả cải tiến của tấn công gặp ở giữa lên Kalyna ([14]).
Bel IT	**Các phiên bản Kích thước khối/ độ dài khóa/số vòng** 128/128/8	Có cấu trúc khá phức tạp được lấy cảm hứng từ mạng Feistel và Lai-Massey. Trong đó, cấu trúc gồm bốn

	128/192/8 128/256/8 **Cấu trúc:** Lai ghép từ Feistel và Lai-Massey. **Hàm vòng:** Tính phi tuyến được đem lại do sử dụng các phép toán đại số (phép cộng modulo 2^{32} và phép XOR từng bit) và ba hàm khác nhau G_i (với $i = 5,13,21$). Các hàm G_i chứa một áp dụng song song của một S-hộp 8-bit và phép dịch vòng dựa trên chỉ số i. Hàm F của cả thành phần Feistel và Lai-Massey bao gồm phép trộn khóa (phép cộng modulo 32) và áp dụng một lần hàm G_i cho các từ khác nhau. Nó một cách khác chúa là một F-hàm tựa GOST (GOST 28147-89) với kích cỡ 32 bit. **Lược đồ khoá:** Đơn giản được lấy trực tiếp từ khoá chính, không qua biến đổi nào.	nhánh, được chia ra hai cặp, mỗi cặp này được trộn với nhau bởi mạng Feistel ba vòng. Tuy nhiên, hai mạng Feistel song song này được trộn với nhau ở giữa (sau hai vòng của mạng chứa a và b, và vòng đầu của mạng chứa c và d), bằng cách áp dụng một vòng Lai-Massey cho nhánh b và c. Các thám mã lên mã khối này. Có thể kể tới thám mã tích phân trong [56] và thám mã dựa trên gây lỗi trong [15]. Độ an toàn đối với thám mã vi sai có thể được tính toán dựa trên MILP của mã khối này trong [16].
SMS4 GB/T 32905	**Kích cỡ khối/độ dài khóa/số vòng** 128/128/32 **Cấu trúc:** Feistel không cân bằng bốn nhánh. **Hàm vòng:** S-hộp 8 bit Biến đổi tuyến tính dựa trên dịch vòng, có số nhánh 5. **Lược đồ khoá:** Dựa trên một biến đổi cập nhật tương đương với hàm vòng như thay thế biến đổi tuyến tính bằng một tuyến tính khác tương đương.	Zhang và cộng sự đã đưa ra tấn công vi sai cải tiến lên 22 vòng trong [17]; Liu và cộng sự đã giới thiệu tấn công tuyến tính bội lên SMS4 rút gọn 22 vòng [18]. Nhóm tác giả Su đã cải tiến hơn kết quả trong [17] bằng cách đề xuất tấn công vi sai lên phiên bản 23 vòng [19]; tấn công này là tốt nhất so với các kết quả đánh giá trước đó dựa trên số lượng vòng và độ phức tạp. Cho và Nyber cũng đã giải đáp câu hỏi được đặt ra trong [20] và đề xuất một tấn công tuyến tính đa chiều lên SMS4 23 vòng [21]. Hơn nữa, nhóm tác giả Zhang đã đưa ra cận dưới của số lượng các S-hộp tích cực tuyến tính đối với mã khối tựa SMS4 trong [22]. Nhóm tác giả Liu tiếp tục cải tiến tấn công tuyến tính cho SMS 23 vòng trong bài báo [23]. Gần đây, nhóm tác giả Chen cũng đã xem xét tấn công vi sai không thể trong bài báo [24]; cũng như tấn công tuyến tính đa chiều trong [21].
ARIA KS X 1213-1	**Kích cỡ khối/độ dài khóa/số vòng** 128/128/12 128/192/14 128/256/16	Tấn công tốt nhất lên ARIA là tấn công gặp ở giữa lên 8-vòng với độ phức tạp dữ liệu là 2^{56} [25].

	Cấu trúc: SPN **Hàm vòng:** Tầng phi tuyến sử dụng hai S-hộp 8bit dựa trên ánh xạ nghịch đảo. Tầng tuyến tính sử dụng ma trận nhị phân 16×16 trên trường $GF(2^8)$ có số nhánh tối ưu. **Lược đồ khoá:** sử dụng một mã pháp 3-vòng Feistel 256-bit với các hằng số vòng.	
RC6	**Kích thước khối/Độ dài khóa/Vòng:** 128/128/20 128/192/20 128/256/20 **Cấu trúc:** Feistel (4 nhánh) **Hàm vòng:** Sử dụng các phép toán cộng và nhân số nguyên để tăng độ khuếch tán cho mỗi vòng. Hơn nữa, sử dung phép dịch vòng phụ thuộc dữ liệu. **Lược đồ khóa:** Đơn giản.	Tấn công thống kê tốt nhất lên RC6 bởi Henri Gilbert và cộng sự [26] đã phá vỡ $RC6-32/14/16$ và tấn công tương quan tốt nhất bởi Knudsen và Meier [27] có thể phân biệt và khôi phục khóa lên RC6-32/15/32. Từ một phần khóa được gọi là các khóa yếu, RC6 bị tấn thương bởi tấn công tuyến tính bội (multiple linear attack) lên 18 vòng.
LEA ISO/IEC 29192-2	**Các phiên bản kích cỡ khối/độ dài khóa/số vòng** 128/128/24; 128/192/28; 128/256/32 **Cấu trúc:** Mạng Feistel 4-nhánh với hàm vòng có dạng ARX **Hàm vòng:** Sử dụng các phép dịch vòng theo các tham số đã được lựa chọn định hướng thiết kế, phép cộng modulo, phép XOR. **Lược đồ khóa:** Đơn giản, sử dụng phép cộng modulo để tạo tính phi tuyến giữa các khóa vòng.	Hiện nay, chưa có tấn công thám mã hiệu quả lên LEA được công bố. Tất cả các đánh giá độ an toànđược đưa ra trong bài báo thiết kế [28]. Trong đó, thuật toán có hành lang an toàn là 15/24 vòng đối với phiên bản 128-bit, 16/28 vòng đối với phiên bản 192-bit, 18/32 vòng đối với phiến bản 256-bit.
CLEFIA ISO/IEC 29192-2	**Các phiên bản kích cỡ khối/độ dài khóa/số vòng** 128/128/18; 128/192/22; 128/256/26 **Cấu trúc:** Feistel **Hàm vòng:** Sử dụng hai S-hộp 8-bit, hai ma trận MDS có kích thước 4×4 trên trường $GF(2^8)$ để thiết kế đạt được số lượng S-hộp hoạt động nhiều hơn theo cơ chế chuyển đổi khuếch tán. **Lược đồ khóa:** Đơn giản, sử dụng phép cộng modulo để tạo tính phi tuyến giữa các khóa vòng.	Tấn công tốt nhất lên Clefia là tấn công vi sai không thể yêu cầu $2^{126,83}$ bản rõ lựa chọn đã phá vỡ 13 vòng với độ phức tạp $2^{126,83}$ phép mã hóa đối với độ dài khóa 128-bit [29], các tấn công tương tự có thể áp dụng cho 14 và 15 vòng đối với phiên bản có kích thước khóa 192-bit và 256-bit.
MISTY-1 ISO/IEC 18033-3	**Kích thước khối/Độ dài khóa/Vòng:** 64/128/8	Được thiết kế bởi Matsui trong [30]. Tấn công tốt nhất lên số vòng rút gọn của MISTY là tấn công tích phân 5-

	Cấu trúc: Feistel **Hàm vòng:** F-hàm của hàm vòng được gọi là các "FO-hàm", có dạng mạng Matsui có kích thước 32-bit cân bằng, hai nhánh với ba vòng. Hàm vòng trong FO-hàm cũng là mạng Matsui với kích thước 16-bit (9+7). Hàm FO sử dụng hai S-hộp, một S-hộp 9-bit và một S-hộp 7 bit. Một hàm nữa được sử dụng là hàm có khóa FL. Hàm này gần tương tự như với hàm FL của mã khối Camellia, điểm khác là bỏ phép dịch vòng. Việc chọn mạng Matsui cho phép một vài phép toán có thể thực hiện song song hóa. **Lược đồ khóa:** Đơn giản.	vòng bởi Knudsen và Wagner [31]. Tấn công này yêu cầu 2^{34} bản rõ được lựa chọn và độ phức tạp thời gian là 2^{48}. Trong bài báo gần đây, Yasutaka Igarashi và cộng sự đã phá vỡ 8 vòng của thuật toán MISTY-2 không có hàm FL với thời gian $2^{57.4}$ sử dụng 2^{35} khối bản rõ được chọn.
HIGHT TTAS.KO-12.0040	**Kích cỡ khối/độ dài khóa/số vòng** 64/128/18 **Cấu trúc:** Feistel dựa trên cấu trúc ARX **Hàm vòng:** Sử dụng phép cộng XOR, phép cộng modulo 2^8 và hai hàm tuyến tính F_0 và F_1, cung cấp độ khuếch tán hướng bit. **Lược đồ khóa:** Đơn giản, sử dụng thanh ghi 128-bit.	Tấn công vi sai không thể lên 27-vòng của HIGHT trong bài báo [32].
SIMON ISO/29167-21	**Các phiên bản kích cỡ khối/độ dài khóa/số vòng:** 32/64/32; 48/72/36; 48/96/36; 64/96/42; 64/128/44; 96/96/52; 96/144/54; 128/128/68; 128/192/69 128/256/72 **Cấu trúc:** có Feistel cổ điển **Hàm vòng:** sử dụng các phép toán đơn giản: phép XOR từng bit, phép toán AND từng bit và phép dịch vòng trái. Các nhà thiết kế SIMON phản đối bước làm trắng bản mã và bản rõ vì cho rằng các phép toán như vậy ảnh hưởng bất lợi đến kích thước mạch. **Lược đồ khoá:** dựa trên một thanh ghi dịch phản hồi FRS đơn giản dựa trên các phép XOR và dịch vòng.	Thuật toán hướng phần mềm. Kết quả thám mã tốt nhất lên SIMON là thám mã vi sai lên 46 vòng cho SIMON 128/128 với $2^{125,6}$ dữ liệu, $2^{40,6}$ byte bộ nhớ, và độ phức tạp thời gian là $2^{125,7}$ với tỉ lệ thành công là 0,632 [33].
SPECK ISO/29167-21	**Các phiên bản kích cỡ khối/độ dài khóa/số vòng:** 32/64/22; 48/72/22; 48/96/23; 64/96/26; 64/128/27; 96/96/28; 96/144/29; 128/128/32; 128/192/33	Thuật toán hướng phần cứng. Kết quả thám mã tốt nhất lên SPECK là thám mã vi sai lên số vòng rút gọn, đã phá vỡ khoảng 70-75% số vòng của hầu hết các phiên bản xem [34].

	128/256/34 **Cấu trúc:** ARX **Hàm vòng:** sử dụng các phép toán trên các từ n-bit sau: $\oplus$, phép toán XOR từng bit; +, phép toán cộng modulo 2^n và phép dịch vòng trái và phải. Các nhà thiết kế thừa nhận rằng hàm vòng SPECK có các điểm tương đồng với hàm trộn của mã khối Threefish đồng thời chỉ ra một số điểm khác biệt đáng kể (chẳng hạn, số lượng dịch vòng được cố định, không có hoán vị từ). **Lược đồ khoá:** sử dụng lại hàm vòng cho việc sinh các khóa vòng.	
PRESENT ISO/IEC 29167-11 ISO/IEC 29192-2	**Các phiên bản kích thước khối/độ dài khóa/vòng:** 128/80/31; 128/128/31 **Cấu trúc:** SPN **Hàm vòng sử dụng:** S-hộp 4 bit thuộc vào 16 lớp S-hộp tối ưu về cài đặt phần cứng. Tầng tuyến tính hoán vị bit **Lược đồ khoá:** Sử dụng thanh ghi dịch để cập nhật trạng thái khoá bằng cách dịch vòng và phần các bit ít ý nghĩa được cho qua S-hộp và được cộng XOR với một vài bit của giá trị bộ đếm.	Tấn công vi sai 26 vòng trên 31 vòng được đề xuất năm 2014 [35]. Các tấn công một số vòng dựa trên thám mã biclique [35].

3.1.2. Mã khối trong giai đoạn chuyển dịch hậu lượng tử

3.1.2.1. Giới thiệu về tính toán lượng tử

Một máy tính lượng tử không còn là một ý tưởng, giả thuyết. Theo nhiều chuyên gia, đây là công nghệ quan trọng nhất của thế giới và đang có một cuộc chạy đua giữa các quốc gia để giành ưu thế trong công nghệ lượng tử với một máy tính lượng tử có đủ số lượng qubit và khả năng chịu lỗi. Mỹ, Trung Quốc, Pháp, Anh, Đức và Nga là những người dẫn đầu trong cuộc đua, trong khi các quốc gia khác đang nỗ lực hết sức để tham gia. Cuộc chạy đua giành quyền kiểm soát công nghệ "Tính toán lượng tử" không chỉ giới hạn trong phạm vi quốc gia mà còn được thúc đẩy đáng kể bởi những gã khổng lồ công nghệ hàng đầu như Microsoft, IBM, Google, D-Wave, Toshiba, … Tính toán lượng tử trở nên phổ biến sau khi xuất bản bài báo "*Vật lý mô phỏng bằng máy tính*" của nhà vật lý lý thuyết người Mỹ Feynman [36]. Trong bài báo, Feynman đề xuất việc sử dụng các trạng thái lượng tử để tính toán.

Tính toán lượng tử là một ứng dụng của cơ chế lượng tử sử dụng hiện tượng lượng tử để thực hiện tính toán. Máy tính lượng tử là một thiết bị thực hiện tính toán lượng tử. Nó điều khiển các trạng thái của qubit theo cách có kiểm soát để thực hiện

các thuật toán. Trong các máy tính cổ điển, thông tin được mã hóa theo từng bit, trong đó mỗi bit có thể là 0 hoặc 1. Trong tính toán lượng tử, thông tin được mã hóa trong qubit. Các qubit có thể đồng thời ở cả 0 và 1. Cơ chế lượng tử là một hiện tượng kỳ lạ. Máy tính lượng tử được xây dựng bằng cách sử dụng các tính năng sau của trạng thái lượng tử:

- **Sự chồng chất:** Các hệ thống lượng tử có thể tồn tại ở hai trạng thái cùng một lúc. Một qubit có thể ở 0 và 1 cùng một lúc. Khi phép đo được thực hiện, qubit thu gọn về 0 hoặc 1.
- **Sự vướng víu:** Đó là một hiện tượng cơ học lượng tử trong đó trạng thái của các hạt bị vướng víu có thể được mô tả có liên quan đến nhau. Phép đo được thực hiện trên một hạt bị vướng víu sẽ ngay lập tức ảnh hưởng đến hạt bị vướng víu khác bất kể khoảng cách giữa các hạt bị vướng víu.
- **Giao thoa:** Ý tưởng cơ bản trong tính toán lượng tử là kiểm soát xác suất qubit thu gọn vào một trạng thái đo cụ thể. Giao thoa lượng tử, sản phẩm phụ của sự chồng chất, cho phép kiểm soát phép đo qubit hướng tới trạng thái hoặc tập hợp trạng thái mong muốn.

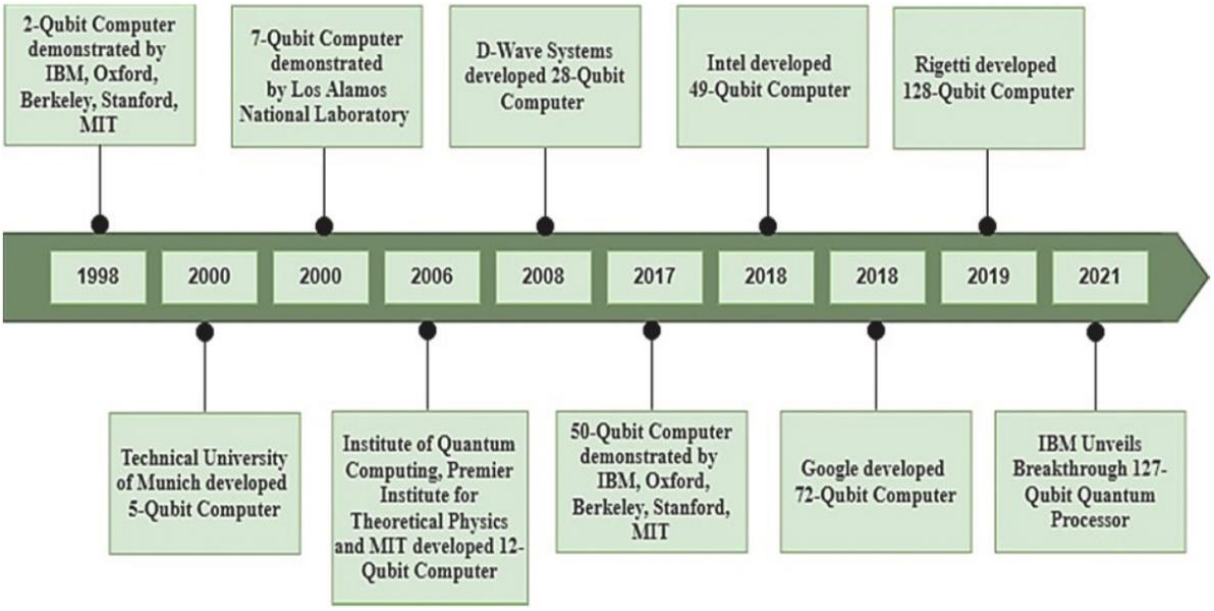
3.1.2.2. Thang thời gian phát triển tính toán lượng tử

Lĩnh vực tính toán lượng tử đang phát triển nhanh chóng. Trên toàn cầu có sự tăng trưởng theo cấp số nhân trong lĩnh vực này. Xây dựng một máy tính lượng tử với qubit cao hơn và kiểm soát lỗi chính xác là mục tiêu duy nhất của các nhà nghiên cứu. Có những thành tựu đáng kể trong 22 năm qua trong lĩnh vực này xem Hình 1. Các trung tâm tính toán MIT, Oxford, Berkely và IBM có thể phát triển máy tính lượng tử 2 qubit vào đầu năm 1998. Google đã phát triển máy tính lượng tử 72 qubit vào năm 2018. Rigetti vào năm 2019, tuyên bố sẽ phát triển máy tính lượng tử 128 qubit trong vòng một năm.

Sự phát triển của Máy tính lượng tử có thể được chia thành ba thế hệ.

- **Thế hệ đầu tiên:** Máy tính lượng tử thế hệ đầu tiên được phát triển ở giai đoạn đầu cho mục đích sử dụng phi thương mại. Các mô hình này được xây dựng để chứng minh khái niệm với độ phức tạp từ thấp đến trung bình.
- **Thế hệ thứ hai:** Nhiều tổ chức đạt được bước đột phá trong nghiên cứu ban đầu và sở hữu cơ sở hạ tầng phần cứng cần thiết có thể phát triển máy tính lượng tử với số lượng qubit và độ phức tạp cao hơn. Máy tính lượng tử thế hệ thứ hai chỉ được thiết kế và phát triển cho các ứng dụng thương mại và nghiên cứu cao cấp tập trung vào khả năng mở rộng và tốc độ được cải thiện. Những máy tính lượng tử này có thể được cho thuê để đáp ứng nhu cầu tính toán cao hơn giống như tính toán đám mây để phục vụ trên cơ sở nhu cầu.
- **Thế hệ thứ ba:** Thế hệ thứ ba sẽ là ưu thế lượng tử thực sự vì sự tăng trưởng và phát triển theo cấp số nhân sẽ làm giảm chi phí phần cứng. Máy tính lượng tử sẽ có giá cả phải chăng và dễ tiếp cận với đại chúng. Máy tính lượng tử thế hệ thứ ba sẽ

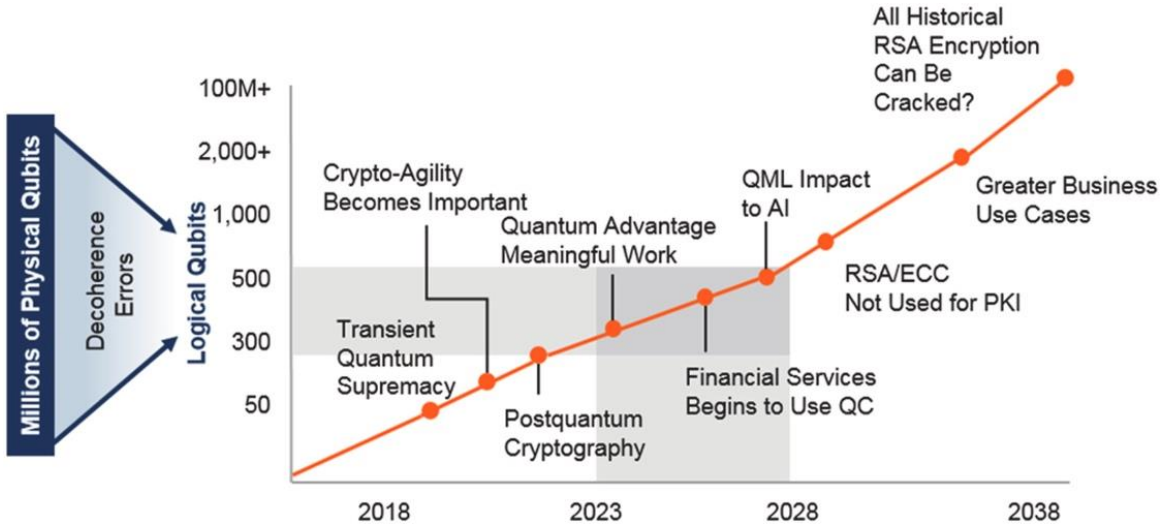
mang lại giải pháp khả thi cho nhiều ứng dụng phi thương mại và nó sẽ vượt trội so với máy tính và ứng dụng cổ điển.



Hình 1 - Tăng trưởng tính toán lượng tử theo qubit từ năm 1998 đến năm 2021.

Có nhiều thách thức công nghệ đối với sự phát triển của Máy tính lượng tử với số lượng qubit cao hơn. Ban đầu, nghiên cứu đột phá về tính toán lượng tử diễn ra khá chậm và mất nhiều thời gian để hiện thực hóa mô hình hoạt động. Tuy nhiên, trong vài năm gần đây, có một sự phát triển chưa từng thấy. Các nhà khoa học trên khắp thế giới đang giải quyết nhiều vấn đề thách thức khác nhau để phát triển một máy tính lượng tử giá cả phải chăng với độ chính xác cao hơn [37-40]. Gartner vào năm 2018 đã công bố dòng thời gian ước tính cho dung lượng qubit vật lý và ứng dụng máy tính lượng tử trong đời thực (xem Hình 2). Nếu dòng thời gian của Gartner được tin tưởng, thì xã hội sẽ sớm chứng kiến uy quyền lượng tử. Mặc dù hình dung về tương lai cho thấy một thành tựu rất tốt cho lĩnh vực tính toán, nhưng đó là dấu hiệu của một cơn bão đối với an ninh mạng.

Sự phát triển của công nghệ lượng tử là một tin tốt cho nhân loại vì nó sẽ nâng tầm cuộc sống của chúng ta. Tuy nhiên, nó cũng gây ra mối đe dọa nghiêm trọng đối với an ninh mạng, đòi hỏi chúng ta phải thay đổi cách mã hóa dữ liệu của mình. Theo điểm nổi bật của MIT Technology Review, máy tính lượng tử 20 triệu qubit có thể phá vỡ con số 2048 bit chỉ trong 8 giờ [41]. Mặc dù máy tính lượng tử 20 triệu qubit hiện không tồn tại, nhưng chúng ta cần chuẩn bị sẵn sàng và đón đầu mối đe dọa. Chúng ta không thể đợi cho đến khi những máy tính lượng tử mạnh mẽ đó bắt đầu phá mã hóa của chúng ta, thì sẽ quá muộn. Đã đến lúc phân tích nghiêm túc mối đe dọa hậu lượng tử và chuẩn bị cho phù hợp [42-44]. Chúng ta cần xác định định hướng nghiên cứu và lập kế hoạch chiến lược để đối phó với các nguy cơ này.



Hình 2 - Ước tính dòng thời gian Qubit (Nguồn Gartner [45]).

3.1.2.3. Kế hoạch dịch chuyển mật mã hậu lượng tử

Theo mô tả của Tiến sĩ Michele Mosca [46], tính cấp bách đối với bất kỳ tổ chức nào trong việc chuyển sang mật mã kháng lượng tử hoặc an toàn lượng tử phụ thuộc vào các thông số sau:

- **Thời hạn sử dụng (X năm):** Số năm chúng ta cần khóa mật mã của mình để duy trì an toàn và dữ liệu của bạn được bảo vệ.
- **Thời gian di chuyển (Y năm):** Số năm cần thiết để phát triển, triển khai và di chuyển sang giải pháp an toàn lượng tử.
- **Dòng thời gian nguy cơ (Z Years):** Số năm trước khi các máy tính lượng tử quy mô lớn được chế tạo có thể phá vỡ các thuật toán mật mã hiện tại.

Nếu Dòng thời gian nguy cơ ngắn hơn tổng Thời gian sử dụng và Thời gian di chuyển tức là 'X + Y > Z' thì đó là vấn đề cần quan tâm nghiêm túc vì các tổ chức sẽ không thể bảo vệ "tài sản" của họ cho số năm cần thiết để chống lại tấn công lượng tử [40, 47, 48].

Đánh giá chính xác dòng thời gian nguy cơ (tức là giá trị cho 'Z') là một nhiệm vụ đầy thách thức vì có rất nhiều trở ngại trong việc xây dựng máy tính lượng tử với số lượng qubit và hiệu quả cần thiết. Tuy nhiên, xu hướng nghiên cứu và phát triển hiện nay cho thấy một ngày không xa chúng ta sẽ có những chiếc máy tính lượng tử với sức mạnh tính toán cần thiết [49, 50]. Tại một thời điểm nào đó, chúng ta cũng có thể mong đợi Định luật Moore hỗ trợ mở rộng quy mô phát triển của công nghệ tính toán lượng tử như đã từng xảy ra với các công nghệ tính toán cổ điển.

Mối đe dọa không chỉ là thách thức đối với viễn cảnh tương lai mà nó còn có ý nghĩa quan trọng đối với chính ngày hôm nay [51, 52]. Rất có thể nhiều tin tặc, do chính nhà nước bảo trợ hoặc chính các quốc gia có thể đang chặn và thu thập các tin nhắn được mã hóa với hình dung rằng họ sẽ có thể giải mã các tin nhắn này khi có tài nguyên máy tính lượng tử trong tương lai. Nếu các tin nhắn riêng tư của ngày hôm nay

bị tiết lộ ngay cả trong tương lai, nó sẽ có tác động bất lợi nghiêm trọng đối với các tập đoàn, tổ chức chính phủ, quân đội và quan hệ ngoại giao của các quốc gia. Vì vậy, mối đe dọa không phải là trong tương lai như Định lý Mosca đã chỉ ra mà đã bắt đầu từ ngày đầu tiên thuật toán của Shor được giới thiệu vào năm 1994. Do đó, bất kỳ giao tiếp nào được thực hiện kể từ năm 1994 bằng thuật toán mã hóa hiện có không an toàn lượng tử đều dễ bị tấn công và có thể bị lộ trong tương lai [53, 54].

3.1.2.4. Sự ảnh hưởng của tính toán lượng tử tác động đến thuật toán mật mã hiện có

Một máy tính lượng tử thực tế với số lượng qubit cần thiết (hàng triệu qubit) sẽ có thể phá vỡ tất cả các hệ thống mật mã khóa công khai hiện đại **.** Báo cáo năm 2016 của Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) về mật mã sau lượng tử được trình bày trong Bảng 2, nêu bật tác động của tính toán lượng tử đối với các thuật toán mật mã phổ biến.

Bảng 2 - Tác động của tính toán lượng tử đối với mật mã phổ biến.

Thuật toán mật mã	Kiểu	Mục đích	Tác động từ máy tính lượng tử quy mô lớn
AES	Khóa đối xứng	mã hóa	Kích thước khóa lớn hơn cần thiết
SHA-2, SHA-3	————	Hàm băm	Đầu ra lớn hơn cần thiết
RSA	Khóa công khai	Chữ ký, thiết lập khóa	Không còn an toàn
ECDSA, ECDH (Mật mã đường cong elliptic)	Khóa công khai	Chữ ký, trao đổi khóa	Không còn an toàn
DSA (Mật mã trường hữu hạn)	Khóa công khai	Chữ ký, trao đổi khóa	Không còn an toàn

Các thuật toán tiềm năng được đánh giá dựa trên nhiều yếu tố như khả năng kháng lại các tấn công phân tích mã hóa, hiệu quả của thuật toán, khả năng tương tác, tính khả thi khi triển khai, …

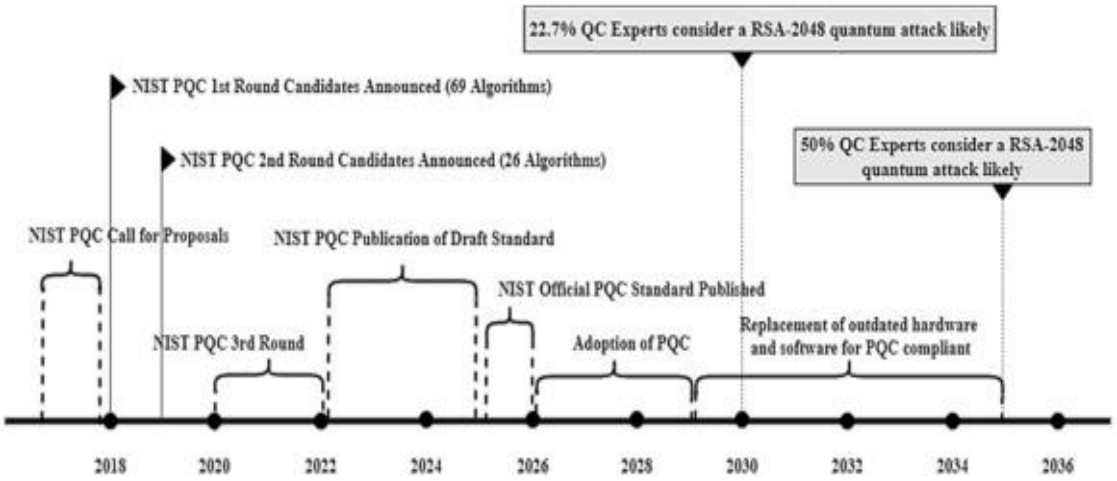
3.1.2.5. Các nỗ lực chuyển dịch hậu lượng tử trên thế giới

Tầm quan trọng của thuật toán Quantum-Safe đã được các tổ chức chính phủ và những gã khổng lồ CNTT trên toàn thế giới hiểu rõ. Mối đe dọa là có thật và ngày đó đang đến rất nhanh. Do đó, nhiều quốc gia và tổ chức tiêu chuẩn đã chủ động thiết kế, phát triển, thử nghiệm và chiến lược di chuyển cho các thuật toán an toàn lượng tử [55].

Nhóm làm việc, diễn đàn và các tổ chức tiêu chuẩn đang xuất bản khuôn khổ, tài liệu chính sách, sách trắng, chi tiết kỹ thuật, tài liệu tiêu chuẩn hóa và chiến lược chuyển đổi hiệu quả về chi phí sang tiền mã hóa hậu lượng tử. Một số tổ chức dẫn đầu quá trình tiêu chuẩn hóa là:

Viện tiêu chuẩn và công nghệ Quốc gia Hoa Kỳ. Vào cuối năm 2016, Viện Tiêu chuẩn và Công nghệ Quốc gia Hoa Kỳ (NIST – National Institute of Standards and Technology) đã bắt đầu nỗ lực thu thập, phân tích nhằm chuẩn hóa các thuật toán mật mã mới có khả năng kháng lại các tấn công bằng máy tính lượng tử quy mô lớn. Nỗ lực này được đặt tên là dự án tiêu chuẩn hóa mật mã hậu lượng tử (PQC – Post Quantum Cryptography) và tập trung vào các thuật toán mật mã phi đối xứng cho *mã hóa dữ liệu, chữ ký số* và *các cơ chế bọc khóa.*. Dòng thời gian hợp lý để phát triển và áp dụng các tiêu chuẩn NIST PQC được thể hiện trong Hình 3 [97]. Dự án PQC được dự định kết thúc vào năm 2024. Tuy nhiên, NIST đã có kế hoạch ban hành một chuẩn chữ ký số trong thời gian sớm hơn do lo ngại trước tình hình phát triển của máy tính lượng tử với những hiểu biết chắc chắn nhất về an toàn lượng tử tại thời điểm hiện tại. Đó là những nguyên thủy chữ ký số dựa trên hàm băm với độ an toàn không dựa vào giả thiết bài toán khó nào mà hoàn toàn dựa vào độ an toàn của hàm băm cơ sở. Hiện tại, NIST đang tìm kiếm các phản hồi về dự thảo tiêu chuẩn chữ ký số dựa trên hàm băm có trạng thái của mình. Cụ thể, NIST đã đưa ra *dự thảo tiêu chuẩn NIST SP 800-208* về khuyến cáo đối với lược đồ chữ ký dựa trên hàm băm "stateful" (HBS – Hash Based Signature) vào 21 tháng 6 năm 2018. Trong đó, thuật ngữ "stateful" nhằm chỉ người ký phải lưu giữ trạng thái qua các lần ký để tránh việc sử dụng lại các khóa bí mật. Khác với các lược đồ chữ ký số dựa trên các bài toán khó đã biết như bài toán phân tích số hoặc bài toán logarit rời rạc sẽ bị phá vỡ hoàn toàn bằng cách sử dụng một máy tính lượng tử với năng lực tính toán đủ lớn, lược đồ chữ ký số dựa trên hàm băm vẫn đảm bảo được an toàn do độ an toàn của hàm băm đối với các năng lực tính toán dựa trên máy tính lượng tử. Cụ thể, hàm băm vẫn đạt được mức an toàn bằng một nửa mức an toàn cổ điển khi sử dụng thuật toán Gorver (thuật toán được xem là tốt nhất để phá vỡ độ an toàn của hàm băm) trên máy tính lượng tử. Ngày 05/7/2022, NIST đã công bố nhóm các thuật toán đầu tiên chiến thắng trong cuộc thi tuyển chọn các thuật toán mật mã hậu lượng tử PQC kéo dài 6 năm. NIST đã lựa chọn nhóm thuật toán mật mã đầu tiên được thiết kế chống lại sự tấn công của các máy tính lượng tử trong tương lai. Bốn thuật toán đã được chọn sẽ trở thành một phần của tiêu chuẩn mật mã hậu lượng tử của NIST, dự kiến sẽ được hoàn thiện sau khoảng hai năm. Các thuật toán được thiết kế cho hai mục tiêu chính: mã hóa và xác thực. Tất cả bốn thuật toán được tạo ra bởi các chuyên gia cộng tác từ nhiều quốc gia và tổ chức. Trong đó, đối với mục tiêu mã hóa, NIST đã chọn thuật toán CRYSTALS-Kyber. Còn đối với mục tiêu xác thực, NIST đã chọn ba thuật toán CRYSTALS-Dilithium, FALCON và SPHINCS +. Trong đó, các sự kiện chính như sau:

Thời gian	Sự kiện
4/2015	Hội thảo về an ninh mạng trong thế giới hậu lượng tử tại Gaithersburg, Hoa Kỳ
2/2016	Tiêu chuẩn hóa PQC: Thông báo và phác thảo Lời kêu gọi của NIST cho bản thuyết trình của các đệ trình được đưa ra tại PQCrypto 2016
4/2016	NISTIR 8105, Báo cáo về Mật mã hậu lượng tử, dược phát hành
12/2016	Thông báo Đăng ký Liên bang – Thông báo Yêu cầu Đề cử cho các thuật toán mật mã hậu lượng tử khóa công khai
30/11/2017	Hạn nộp hồ sơ cho Quy trình tiêu chuẩn hóa PQC của NIST
12/2017	Các ứng viên đầu tiên được công bố. Giai đoạn bình luận công khai về các ứng viên vòng đầu tiên bắt đầu.
4/2018	Hội thảo tiêu chuẩn hóa NIST PQC đầu tiên tại Ft. Lauderdale, Florida
1/2019	Các ứng viên vòng hai đươc công bố. NISTIR 8240 – Báo cáo thực trạng về Vòng đầu tiên của Quy trình tiêu chuẩn hóa PQC của NIST được phát hành. Giai đoạn bình luận công khai về các ứng viên vòng hai bắt đầu.
4/2020	NIST mời nhận xét từ những người đệ trình và cộng đồng để hình thành quy trình ra quyết định cho việc lựa chọn các ứng viên vòng ba.
6/2020	NIST công bố các ứng viên lọt vào vòng chung kết thứ ba và các ứng viên thay thế. NIST IR 8309 – Báo cáo thực trạng về vòng thứ hai của Quy trình chuẩn hóa mật mã hậu lượng tử của NIST được phát hành. Giai đoạn bình luận của công chúng về các ứng viên vòng ba bắt đầu.
6/2021	Hội thảo tiêu chuẩn hóa PQC của NIST lần thứ ba được tổ chức trực tuyến.
7/2022	Công bố các thuật toán ứng viên sẽ được tiêu chuẩn hóa cùng với ứng cử viên thay thế tiến vào vòng thứ tư. NIST IR 8413 – Báo cáo thực trạng về Vòng thứ ba của Quy trình chuẩn hóa mật mã hậu lượng tử của NIST được phát hành.
7/2023-nay	Vòng 4



Hình 3 -Thời gian hợp lý để phát triển và áp dụng các tiêu chuẩn NIST PQC.

Viện tiêu chuẩn viễn thông Châu Âu . Vào tháng 3 năm 2015, Ủy ban kỹ thuật của Viện tiêu chuẩn viễn thông Châu Âu (ETSI - European Telecommunications Standard Institute) đã thành lập nhóm làm việc về mật mã an toàn lượng tử (WG QSC - Working Group for Quantum-Safe Cryptography). Đây có thể được coi là nhóm chuẩn hóa tập trung cho mật mã an toàn lượng tử mang tính thương mại đầu tiên. Trọng tâm hướng tới của nhóm là cài đặt thực thi và phát triển các nguyên thủy an toàn lượng tử, bao gồm các cân nhắc về hiệu năng, khả năng, giao thức và cân nhắc kiến trúc đối với các ứng dụng cụ thể. Từ đó cung cấp cho các nhóm và cơ quan tiêu chuẩn khác như Liên minh Viễn thông Quốc tế (ITU - International Telecommunications Union) và Tổ chức chuyên trách kỹ thuật internet (IETF - Internet Engineering Task Force). Công việc của ETSI đã được phát triển từ các nghiên cứu khảo sát phân tích hậu quả của việc sử dụng năm lớp nguyên thủy an toàn lượng tử để từ đó đưa ra các đặc tả về kỹ thuật (TS - Technical Specifications), cũng như công thức để xây dựng các hệ thống cụ thể mà khách hàng đang dự định triển khai. Hiện nay, các công việc đã hoàn thành của tổ chức này gồm: *Báo cáo nhóm ETSI QSC001* "*Phân tích các nguyên thủy an toàn lượng tử*"[1]; *Báo cáo nhóm ETSI QSC003* "*Các kịch bản nghiên cứu và sử dụng mật mã an toàn lượng tử*"[2]; *Báo cáo nhóm ETSI QSC004* "*Phân tích mối đe dọa của an toàn lượng tử*"[3]; *Báo cáo nhóm ETSI QSC006* "*Giới hạn của tính toán lượng tử lên khóa đối xứng*"[4]; *Báo cáo kỹ thuật ETSI ETSI TR 103 570* "*Phân tích cài đặt trao đổi khóa an toàn lượng tử*"[5]; *Báo cáo kỹ thuật ETSI TR 103 617* "*Các mạng riêng ảo an toàn lượng tử*"[6]; *Báo cáo kỹ thuật ETSI TR 103 618* "*Mã hóa dựa trên định danh an toàn lượng tử*"[7]. Hiện tại, ETSI cũng đang xây

[1] http://www.etsi.org/deliver/etsi_gr/QSC/001_099/003/01.01.01_60/gr_QSC003v010101p.pdf
[2] http://www.etsi.org/deliver/etsi_gr/QSC/001_099/003/01.01.01_60/gr_QSC003v010101p.pdf
[3] http://www.etsi.org/deliver/etsi_gr/QSC/001_099/004/01.01.01_60/gr_QSC004v010101p.pdf
[4] http://www.etsi.org/deliver/etsi_gr/QSC/001_099/006/01.01.01_60/gr_QSC006v010101p.pdf
[5] http://www.etsi.org/deliver/etsi_tr/103500_103599/103570/01.01.01_60/tr_103570v010101p.pdf
[6] https://www.etsi.org/deliver/etsi_tr/103600_103699/103617/01.01.01_60/tr_103617v010101p.pdf
[7] https://www.etsi.org/deliver/etsi_tr/103600_103699/103618/01.01.01_60/tr_103618v010101p.pdf

dựng một số báo cáo sau: QSSC-008 "*Đánh giá các chữ ký số an toàn lượng tử*"; QSSC-13 "*Các kỹ thuật dịch chuyển sang hệ thống an toàn lượng tử*"; QSC-14 "*Các đặc tả kỹ thuật đối với hệ thống con trao đổi khóa lai ghép*".

Tổ chức chuyên trách kỹ thuật Internet. Vào tháng 5 năm 2018, tổ chức này đã công bố phiên bản đa cây của hệ chữ ký Merkle mở rộng, được biết với tên gọi XMSS^{MT} là **chuẩn RFC 8391**. Đến tháng 4 năm 2019, phiên bản đa cây của hệ chữ ký Leigton – Micali, với tên gọi là HSS (Hierarchical Signature System), được công bố là **chuẩn RFC 8554**. Các chuẩn này tương đồng với dự thảo chuẩn NIST SP 800-208 về chữ ký số dựa trên hàm băm "stateful" đã được đề cập ở trên. Hiện nay, tổ chức chuyên trách kỹ thuật Internet đã thực hiện một số công việc tiếp theo cho các yêu cầu kháng lượng tử của mình bao gồm: *Dự thảo về khung tích hợp trao đổi khóa hậu lượng tử vào trong giao thức trao đổi khóa internet phiên bản 2* (IKEv2)[8]; *Dự thảo về trao đổi bổ trợ trong giao thức IKEv2*[9]; *Dự thảo về sử dụng khóa được chia sẻ trước trong CMS (Cryptographic Message Syntax) của trình thư điện tử an toàn S/MIME* [10]; *Dự thảo về sử dụng thuật toán chữ ký số dựa trên cây băm Merkle trong CMS*[11]; *Dự thảo về các khóa chia sẻ trước đối với IKEv2*[12]; *Dự thảo về sử dụng thuật toán chữ ký số dựa trên hàm băm với mã hóa và chữ ký đối tượng CBOR (COSE - CBOR Object Signing and Encryption)*[13].

Một số tổ chức khác: Ủy ban tiêu chuẩn chính thức X9 (ASC X9 - Accredited Standards Committee X9) của Viện tiêu chuẩn quốc gia Hoa Kỳ (ANSI - American National Standards Institute) là một nhóm tiêu chuẩn tài chính chuyên biệt, tập trung vào hệ thống thanh toán điện tử, kiểm tra và hoạt động tại văn phòng, báo cáo giao dịch ngân hàng doanh nghiệp, chứng khoán như cổ phiếu và trái phiếu, và bảo mật dữ liệu/thông tin. Ủy ban này đã thực hiện một số công việc sau: *Sách trắng thông tin của nhóm nghiên cứu rủi ro tính toán lượng tử X9:*[14]; *Báo cáo kỹ thuật TR.50 về kỹ thuật lượng tử trong CMS*[15].

Liên minh Viễn thông Quốc tế (ITU) là cơ quan tiêu chuẩn hóa công nghệ của Liên hợp quốc đặt tại Geneva, đại diện cho 293 quốc gia thành viên. Bộ phận viễn thông của ITU được viết tắt là ITU-T (International Telecommunications Union Telecommunication Sector). Nhóm nghiên cứu số 7 (SG17 - Study Group 17) là một nhóm con của ITU-T chuyên về bảo mật. Tổ chức Tiêu chuẩn Quốc tế (ISO - International Standards Organization) và Ủy ban Kỹ thuật Điện tử Quốc tế (IEC - International Electrotechnical Commission) hợp tác cùng nhau trong công tác bảo mật

[8] https://datatracker.ietf.org/doc/draft-tjhai-ipsecme-hybrid-qske-ikev2/
[9] https://datatracker.ietf.org/doc/draft-smyslov-ipsecme-ikev2-aux/
[10] https://datatracker.ietf.org/doc/draft-housley-cms-mix-with-psk/
[11] https://datatracker.ietf.org/doc/draft-housley-cms-mts-hash-sig/
[12] https://datatracker.ietf.org/doc/draft-ietf-ipsecme-qr-ikev2/
[13] https://datatracker.ietf.org/doc/draft-housley-suit-cose-hash-sig/
[14] https://x9.org/download-qc-ir/
[15] https://x9.org/download-qc-tr50/

tại SG17. Tổ chức này cũng đang nỗ lực nhằm đưa ra các mật mã an toàn kháng lượng tử. Tuy nhiên, các kết quả nhận được cũng rất hạn chế, mới chỉ nghiên cứu ban đầu liên quan tới tính toán lượng tử từ năm 2017 trong các báo cáo ITU-T SG 17 Contribution 529 "*Quá trình làm việc của nhóm nghiên cứu mật mã lượng tử*"[16] và ITU-T SG 17 Contribution 180 "*Đề xuất nghiên cứu mới cho truyền thông an toàn dựa trên mật mã hậu lượng tử*"[17].

Văn phòng Bảo mật Thông tin Liên bang Đức(BSI): Để chuẩn bị cho rủi ro trong tương lai, BSI cũng đã phát hành các hướng dẫn kỹ thuật đề xuất các thuật toán, độ dài khóa và đưa ra các đề xuất cụ thể cho các sơ đồ chữ ký dựa trên Merkle và XMSS. Trọng tâm là phát triển các thuật toán mã hóa khóa công khai an toàn lượng tử. Vào năm 2020, BSI cũng đã phát triển các đề xuất ban đầu cho việc chuyển đổi sang mật mã hậu lượng tử.

3.1.3. Định hướng thiết kế cho mã khối trong chuyển dịch hậu lượng tử

Như vậy, để chuẩn bị cho việc chuyển tiếp lượng tử cùng với sự phát triển của khoa học thám mã, mặc dù không bị ảnh hưởng nghiêm trọng như các thuật toán mã hóa khóa công khai cổ điển, mã khối nói riêng và các thuật toán mật mã đối xứng nói chung cần có định hướng thiết kế trong thời gian sắp tới. Bài toán *tăng độ dài khoá và kích cỡ khối* là đặc biệt cần thiết đối với thuật toán mã khối theo khuyến cáo của các tổ chức mật mã uy tín như NIST của Mỹ (xem [57]), BSI của Đức (xem [58]), Ecrypt-CSA của Châu Âu (xem [59]), ANSSI của Pháp (xem [60]), ENISA (xem [61]), NCSC của Anh (xem [62]),…

Trong [63], một trong số những yêu cầu quan trọng về độ an toàn của các thuật toán tham gia dự án được đưa ra như sau "*Khi cần độ an toàn lâu dài, các thuật toán này nên hướng tới bảo mật hậu lượng tử, hoặc ứng dụng phải cho phép chúng có thể dễ dàng thay thế bằng các thuật toán có độ an toàn hậu lượng tử*" ("When long-term security is needed, these algorithms should either aim for post-quantum security, or the application should allow them to be easily replaceable by algorithms with post-quantum security"). Điều này hoàn toàn có ý nghĩa do nỗ lực đưa ra tiêu chuẩn cho mật mã phi đối xứng hậu lượng tử của NIST gần đây sẽ chỉ có hiệu quả nếu mật mã đối xứng được sử dụng có khả năng kháng lượng tử. Tuy nhiên, để đạt được mức an toàn lượng tử hiệu quả là 128 bit, việc tăng kích thước khoá lên 256-bit là chưa đủ đảm bảo an toàn cho mã khối trong kịch bản mã pháp được lượng tử hoá, sẽ được thảo luận chi tiết trong phần sau. Hơn nữa, kích thước khối của mã khối còn ảnh hưởng trực tiếp tới độ an toàn của chế độ hoạt động sử dụng nó, trong khi độ an toàn này đang bị giảm sút do các thuật toán tìm va chạm lượng tử. Trong quá trình xem xét độ an toàn của mã khối AES của NIST trong [64], hầu hết các đánh giá đều đánh giá cao AES về độ an toàn đã có đối với các thám mã cổ điển, song đều đưa ra những yêu cầu

[16] https://www.itu.int/md/T17-SG17-C-0529,
[17] https://www.itu.int/md/T17-SG17-C-0180

về tăng độ dài khóa và kích thức khối để đảm bảo độ an toàn mong muốn trong bối cảnh hậu lượng tử. Đặc biệt, có yêu cầu cần một mã khối có kích thước khối là 512-bit; tuy nhiên các chuyên gia đều thống nhất kích cỡ khối 256-bit là lựa chọn phù hợp với thời điểm hiện nay. Do đó, một mã khối có kích cỡ khối 256-bit với độ dài khóa được gia tăng tương ứng là cần thiết cho thời kỳ chuyển tiếp hậu lượng tử.

3.2. Thông tin chung về mã khối MKV

Thuật toán mã khối MKV được thiết kế trong khuôn khổ nhiệm vụ khoa học cấp Ban "*Xây dựng thuật toán mã khối đảm bảo cân bằng giữa độ an toàn và hiệu năng xử lý dữ liệu phù hợp sử dụng để bảo mật thông tin trong lĩnh vực dân sự*" được Ban Cơ yếu Chính phủ phê duyệt năm 2020-2021 nhằm tạo ra một thuật toán định hướng sử dụng trong lĩnh vực dân sự. Nhiệm vụ được nghiệm thu tại hội đồng Khoa học - Công nghệ cấp Ban vào ngày: 25/05/2023, kết quả đạt loại ĐẠT. Sau đó, nhiệm vụ đã được ký thanh lý hợp đồng vào ngày 20/07/2023. Trong quá trình phát triển xây dựng, ban đầu mã khối được đặt tên là *ViEncrypt*, sau đó được đổi tên thành MKV để tạo sự thống nhất về kí hiệu cho các tiêu chuẩn mật mã được xây dựng sau này dùng bảo vệ thông tin trong lĩnh vực dân sự.

Mã khối MKV là mã pháp lặp, xử lý khối dữ liệu kích cỡ 128-bit (với khóa 128, 192 và 256 bit) hoặc kích cỡ dữ liệu 256-bit (với khóa 256, 384 và 512 bit). Mã khối này sử dụng cấu trúc dựa trên lược đồ mới, được đặt tên là FLC[18], đạt độ an toàn chứng minh được cả trong mô hình lý thuyết và thực tế, có rất ít mã khối đạt được độ an toàn chứng minh được này.

Cấu trúc FLC trong MKV là cấu trúc được các chuyên gia mật mã của Ban Cơ yếu Chính phủ xây dựng không giống với bất kỳ chuẩn mã khối nào trên thế giới. Có thể nói đây là cấu trúc mang đặc thù riêng theo xu thế "Make in Vietnam". *Cấu trúc FLC không vi phạm bản quyền và không vi phạm quyền sở hữu trí tuệ của bất kỳ tổ chức và cá nhân trong và ngoài nước nào*.

Thuật toán MKV đảm bảo độ an toàn trước các tấn công thám mã mạnh lên mã khối trong mô hình cổ điển và đảm bảo độ an toàn trước tấn công sử dụng thuật toán lượng tử Grover để vét cạn khóa. Tính khách quan trong đánh giá độ an toàn của MKV được thực hiện bởi các tổ chức và cá nhân có uy tín trong và ngoài nước.

Tổ chức:

- Trung tâm 8, Cơ quan An ninh Liên bang FSB, Liên bang Nga.
- Viện Nghiên cứu cao cấp về Toán.
- Phòng Thí nghiệm an toàn thông tin, Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội.

Cá nhân:

[18] Từ FLC viết tắt của từ Four Leaf Clover, nghĩa tiếng việt "cỏ bốn lá".

- PGS.TSKH. Nguyễn Văn Lợi, Trung tâm Ra đa, Tổng công ty Công nghiệp Công nghệ cao Viettel.
- TS. Lưu Hồng Dũng, Học viện KTQS.
- TS. Lê Xuân Đức, Viện 10/Bộ Tư lệnh 86.

Ngoài các cá nhân và tổ chức đã đăng kí trong nhiệm vụ ở trên, nhóm thiết kế cũng đã gửi một số chuyên gia ngoài nước như GS. Phan Dương Hiệu, Trường Bách khoa Paris, Pháp (có một số gợi ý về cách mở rộng mô hình lý thuyết khi xem xét lược đồ FLC); GS. Jia Guo, Đại học NTU, Singapore (có nhận xét về cấu trúc FLC); GS. Knudsen, Đại học DTU, Đan mạch (nhóm thiết kế đã gửi mô tả tổng thể thuật toán tuy nhiên không có phản hồi chính thức). Hơn nữa, độ an toàn của MKV và cấu trúc FLC của thuật toán được cộng đồng khoa học trong và ngoài nước công nhận và đánh giá cao trong các công bố trên hội nghị và tạp chí khoa học có uy tín:

- Cuong Nguyen, Anh Nguyen, Phong Trieu, Long Nguyen, and Lai Tran. *Analysis of a new practically secure SPN-based scheme in the Luby-Rackoff model*. in *The 9th International Conference on Future Data and Security Engineering*. 2022. Springer. (SCOPUS)
- Cuong Nguyen, Nam Tran, and Long Nguyen. *FLC: a new secure and pratical SPN-based scheme*. in *The 9th NAFOSTED Conference on Information and Computer Science (NICS)*. 2022. IEEE. (SCOPUS)
- Nam, Trần Sỹ, Nguyễn Văn Long, and Nguyễn Bùi Cương. *Đề xuất tầng tuyến tính và đánh giá khả năng cài đặt trong xây dựng mã khối 256 bit có cấu trúc FLC*. *Journal of Science and Technology on Information security* 2.16 (2022): 31-38.
- Báo cáo trong hội nghị "*Nghiên cứu ứng dụng mật mã và an toàn thông tin*" năm 2022 tại Học Viện KTMM – Chủ đề "*Về tầng tuyến tính an toàn, hiệu quả cho mã khối có cấu trúc FLC*".
- Tran Sy Nam, Nguyen Van Long, and Nguyen Bui Cuong. "An Optimized Bit-Slice Implementation of Secure 8-Bit Sbox Based on Butterfly Structure." *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*. IEEE, 2023.

Thuật toán MKV đã được quảng bá rộng rãi thông qua các công bố trên ấn phẩm của Ban Cơ yếu chính phủ:

- Hoàng Văn Thức, Nguyễn Bùi Cương. *Một số kết quả trong xây dựng và phát triển chuẩn mã khối sử dụng trong dân sự*, Tạp chí Cơ yếu Việt Nam, số 4-2022, trang 09-12.
- Nguyễn Quốc Toàn, Nguyễn Bùi Cương. *Ban Cơ yếu Chính phủ nghiên cứu, xây dựng chuẩn mã khối dân sự phục vụ chiến lược "Make in Vietnam"*, Tạp chí An toàn thông tin, Số 6 (076) 2023, trang 30-45,

Mã khối MKV cũng được trình bày tại Hội nghị toán học toàn quốc năm 2023, tổ chức tại thành phố Đà Nẵng.

3.3. Cơ sở thiết kế của MKV

Trong phần này, cơ sở thiết kế về cấu trúc, thành phần mật mã và lược đồ khóa được sử dụng cho MKV sẽ phân tích.

3.3.1. Lược đồ FLC và thể hiện FLC-SDS

Lược đồ FLC được nghiên cứu và đánh giá độ an toàn chứng minh được trong [1]. Lược đồ này đạt độ an toàn tới $O(2^{w/2})$ truy vấn trong mô hình Luby-Rackoff khi xem xét các nguyên thủy hàm vòng là bí mật có kích thước là w-bit, cụ thể 3-vòng của FLC đạt tính giả ngẫu nhiên và 5-vòng FLC đạt tính siêu giả ngẫu nhiên. Về bản chất, FLC sử dụng bốn hoán vị con có kích thước w-bit để tạo ra một hoán vị có kích thước $4w$-bit, dựa trên sự kết hợp của hàm "mật mã" f với tầng "phi mật mã" LC (Linear Compose). Hàm f được tính toán dựa trên bốn hàm khác nhau f_0, f_1, f_2, f_3 có kích cỡ w-bit và phép LC được định hướng xử lý theo từ w-bit để tăng hiệu quả cài đặt. Cụ thể, các biến đổi được xác định như sau:

$$f: V_{4w} \to V_{4w}$$

$$(x^0, x^1, x^2, x^3) \mapsto \big(f_0(x^0), f_1(x^1), f_2(x^2), f_3(x^3)\big)$$

và

$$LC: V_{4w} \to V_{4w}$$

$$(x^0, x^1, x^2, x^3) \mapsto (x^1 \oplus x^2 \oplus x^3, x^0 \oplus x^2 \oplus x^3, x^0 \oplus x^1 \oplus x^3, x^0 \oplus x^1 \oplus x^2)$$

Khi đó, hoán vị cơ sở cho lược đồ được xác định như sau:

$$F(f_0, f_1, f_2, f_3)\big((x^0, x^1, x^2, x^3)\big) = (y^0, y^1, y^2, y^3),$$

sao cho

$$\begin{cases} y^0 = f_1(x^1) \oplus f_2(x^2) \oplus f_3(x^3) \\ y^1 = f_0(x^0) \oplus f_2(x^2) \oplus f_3(x^3) \\ y^2 = f_0(x^0) \oplus f_1(x^1) \oplus f_3(x^3) \\ y^3 = f_0(x^0) \oplus f_1(x^1) \oplus f_2(x^2) \end{cases}$$

trong đó các hàm $f_i \in P_w$ (trong đó P_w kí hiệu tập các hoán vị trên không gian các xâu w-bit) và $x^i, y^i \in V_w$ với mọi $i \in \{0, 1, 2, 3\}$.

Tiếp theo, chúng ta định nghĩa *hoán vị r-vòng của lược đồ FLC* là hợp của r hoán vị cơ sở F. Hoán vị $4w$-bit này được tạo thành từ $4r$ hoán vị $f_0^1, f_1^1, f_2^1, f_3^1, \dots, f_0^r, f_1^r, f_2^r, f_3^r \in P_w$ như sau:

$$\mathcal{F}^{(r)}(f_0^1, f_1^1, f_2^1, f_3^1, \dots, f_0^r, f_1^r, f_2^r, f_3^r)$$

$$= F(f_0^r, f_1^r, f_2^r, f_3^r) \circ \dots \circ F(f_0^1, f_1^1, f_2^1, f_3^1).$$

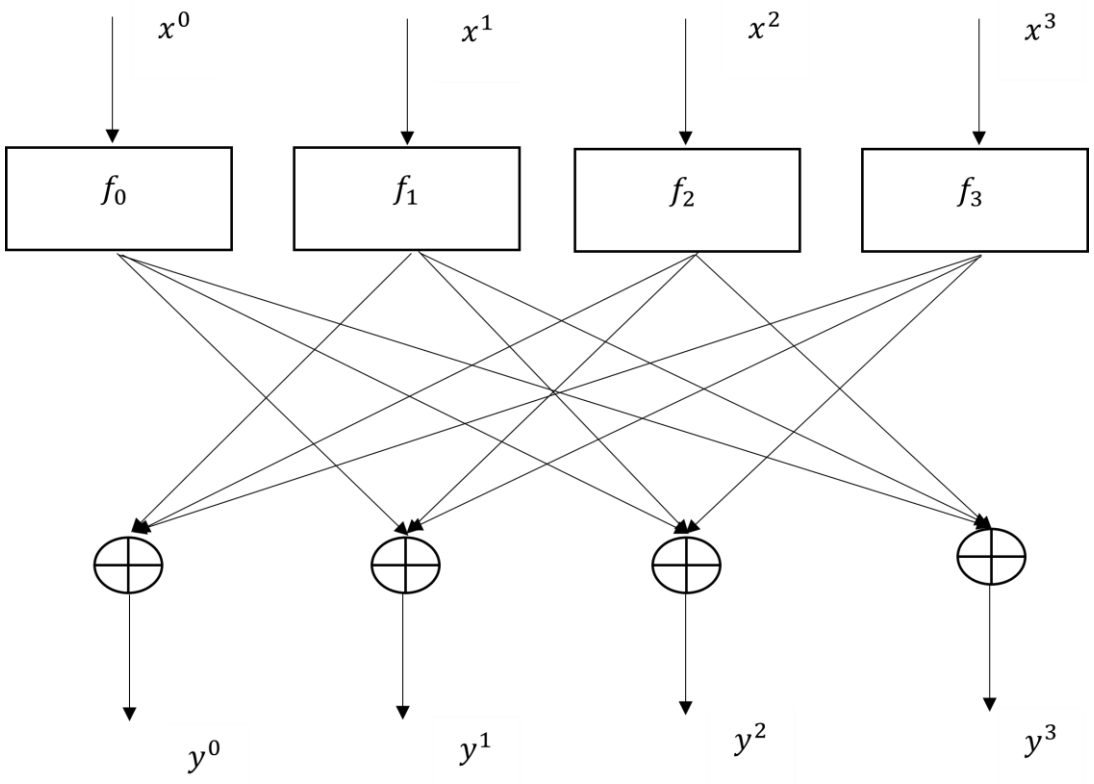
Khi đó, tính giả ngẫu nhiên và siêu giả ngẫu nhiên của FLC có được như sau:

Định lý 3. (xem [1]) *Cho 12 hoán vị ngẫu nhiên hoàn thiện* $f_0^1, f_1^1, f_2^1, f_3^1, f_0^2, f_1^2, f_2^2, f_3^2, f_0^3, f_1^3, f_2^3, f_3^3 \in P_w$, *đặt* $C = \mathcal{F}^{(3)}(f_0^1, f_1^1, f_2^1, f_3^1, f_0^2, f_1^2, f_2^2, f_3^2, f_0^3, f_1^3, f_2^3, f_3^3) \in P_{4w}$ *là hoán vị 3-vòng của lược đồ FLC và* $F^* \in P_{4w}$ *là hoán vị ngẫu nhiên hoàn thiện. Với mọi bộ phân biệt giả ngẫu nhiên* $\mathcal{A}$ *sử dụng tối đa* d *truy vấn mã hóa ta có*

$$\mathrm{Adv}_{\mathcal{A}}(C, F^*) \leq 5d(d-1)2^{-w+1}.$$

Định lý 4. (xem [1]) *Xét* $f_0^1, f_1^1, f_2^1, f_3^1, \ldots, f_0^5, f_1^5, f_2^5, f_3^5 \in P_w$ *là 20 hoán vị ngẫu nhiên hoàn thiện. Gọi* $D = F(f_0^1, f_1^1, f_2^1, f_3^1, \ldots, f_0^5, f_1^5, f_2^5, f_3^5)$ *là 5 vòng của lược đồ mới. Với mọi kẻ tấn công siêu giả ngẫu nhiễn* $\mathcal{A}$ *sử dụng tối đa* d *truy vấn mã hóa hoặc giải mã ta có:*

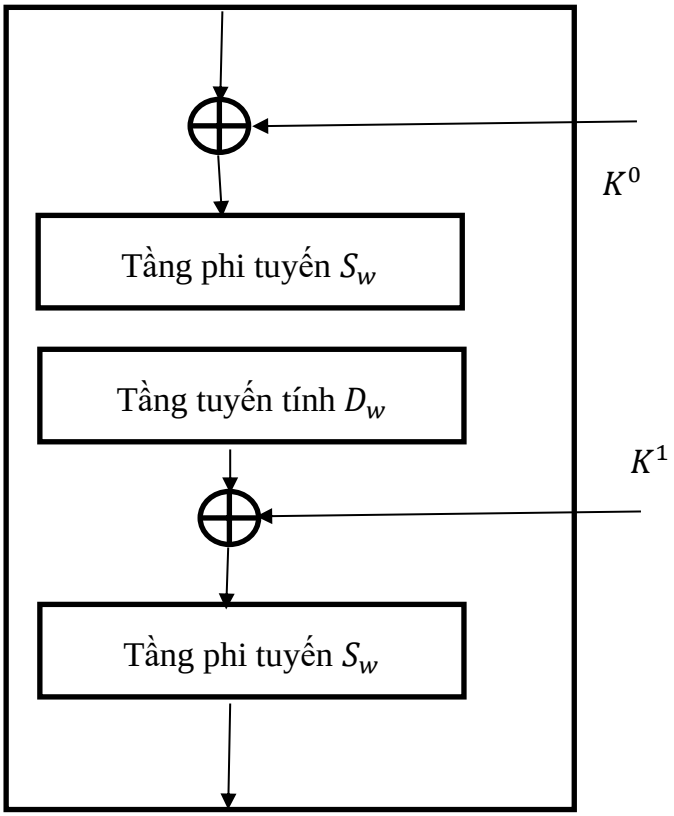
$$\mathrm{Adv}_{\mathcal{A}}(D, F^*) \leq d(d-1)2^{-w+4} + d(d-1)2^{-4w-1}.$$



Hình 4 - Mô tả 1 vòng của lược đồ lặp đề xuất.

Để đưa độ an toàn thực tế cho cấu trúc FLC, một thể hiện FLC-SDS với hàm f có dạng SDS được xem xét trong [2]. Với thể hiện này, chúng ta hoàn toàn xây dựng được một mã pháp lặp có cấu trúc SPN với kích cỡ $4w$-bit, trong đó hàm vòng F chính là hoán vị 1-vòng của FLC. Khi đó, chúng ta sẽ cụ thể hóa các thành phần mật mã cho các hoán vị con f_0, f_1, f_2, f_3. Cụ thể, hàm vòng này sử dụng bốn hoán vị khác nhau dựa trên một hàm cơ sở f_w giữa các xâu có độ dài w-bit. Trong đó hàm cơ

sở f_w bao gồm tầng cộng XOR khóa vòng, tầng phi tuyến S_w bao gồm t S-hộp n-bit để đảm bảo tính xáo trộn của thuật toán và tầng biến đổi tuyến tính D_w giữa các trạng thái w-bit, được minh hoạ trong Hình 5.



Hình 5 - Hàm cơ sở FLC-SDS

Khi đó, cận dưới của số các S-hộp hoạt động vi sai và tuyến tính đối với 2 vòng của mã pháp FLC-SDS được đưa ra dựa trên số nhánh có ma trận D_w (kí hiệu $\mathrm{Br_d}(D), \mathrm{Br_l}(D)$) tương ứng là số nhánh vi sai, tuyến tính) như sau:

Mệnh đề 1.(xem [2]) *Hai vòng của* $\mathrm{FLC-SDS}$ *có ít nhất* $4 \times \mathrm{Br_d}(D_w)$ *S-hộp hoạt động vi sai.*

Mệnh đề 2.(xem [2]) *Hai vòng của* $\mathrm{FLC-SDS}$ *có ít nhất* $4 \times \mathrm{Br}_l(D_w)$ *S-hộp hoạt động tuyến tính.*

3.3.2. Các thành phần mật mã cho MKV

3.3.2.1. Đặc điểm cấu trúc hàm vòng

Dựa trên thể hiện FLC-SDS, các thành phần mật mã cho hàm f_w trong thiết kế MKV-l được lựa chọn như sau:

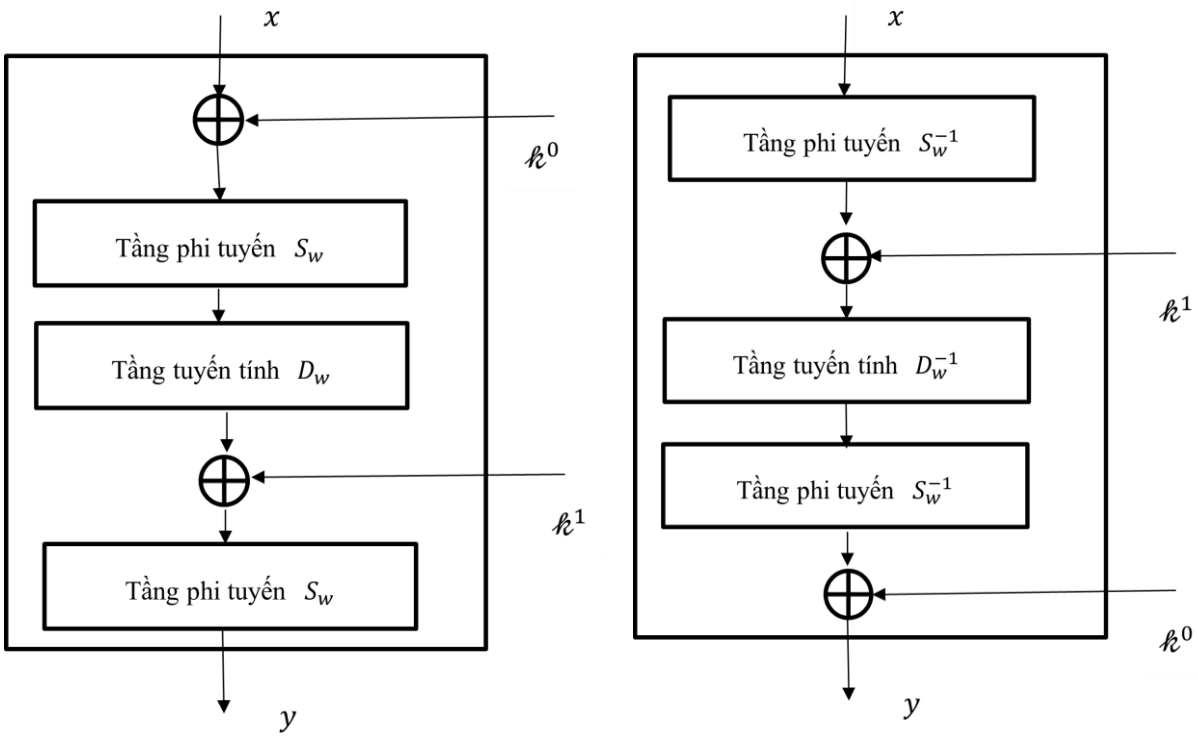
- S-hộp 8 bit cho tầng phi tuyến S_w cho cả hai phiên bản MKV-128 và MKV-256, để đảm bảo tính xáo trộn.
- Ma trận MDS kích cỡ $t \times t$ trên trường $\mathbb{F}_{2^8}$ cho tầng biến đổi tuyến tính D_w, để đảm bảo tính khuếch tán. Cụ thể, ma trận MDS có kích thước 4×4 cho MKV-128 và MDS có kích thước 8×8 cho MKV-256.

Chi tiết độ dài các tham số và các phép biến đổi được mô tả trong từng thuật toán được mô tả trong Bảng 3.

Bảng 3 - Tham số và các phép biến đổi được mô tả trong từng phiên bản

Phiên bản	l	k	R	w	t	Hàm vòng	Hàm cơ sở	Thành phần mật mã	
								Phi tuyến	Tuyến tính
MKV-128	128	128	6	32	4	F_{128}	f_{32}	S_{32}	D_{32}
		192	7						
		256	8						
MKV-256	256	256	6	64	8	F_{256}	f_{64}	S_{64}	D_{64}
		384	7						
		512	8						

Khi đó, f_w xử lý trên các trạng thái con. Cụ thể, S_w và S_w^{-1} tương ứng thực hiện chức năng của SubCells và InvSubCells trên từng từ (trạng thái con) của trạng thái đầu vào $x = (x^0, x^1, x^2, x^3)$. Trong khi, biến đổi tuyến tính D_w và D_w^{-1} thực hiện chức năng của MixWords và invMixWords. Hoán vị cơ sở f_w được xác định như sau $f_w: V_w \to V_w, x \mapsto S_w\left(D_w(S_w(x))\right)$ trong đó $x \in V_w$. Từ hoán vị cơ sở này, ta sẽ xác định f_0, f_1, f_2, f_3 bằng cách cộng XOR khóa vòng $\mathcal{K} = \hbar^0 \| \hbar^1 \in V_{2w}$. Với mỗi khóa vòng $\mathcal{K} = \hbar^0 \| k^1$, ta có hoán vị $f_w^{\mathcal{K}}$ được xác định như sau $f_w^{\mathcal{K}}: V_w \to V_w, x \mapsto S_w\left(D_w\left(S_w(x \oplus \hbar^0)\right) \oplus \hbar^1\right)$ và biến đổi nghịch đảo $\text{inv}f_w^{\mathcal{K}}: V_w \to V_w, x \mapsto S_w^{-1}\left(D_w^{-1}(S_w^{-1}(x) \oplus \hbar^1)\right) \oplus \hbar^0$



Hình 6 - Mô tả hoán vị $f_w^{\mathcal{K}}$ và $\text{inv}f_w^{\mathcal{K}}$.

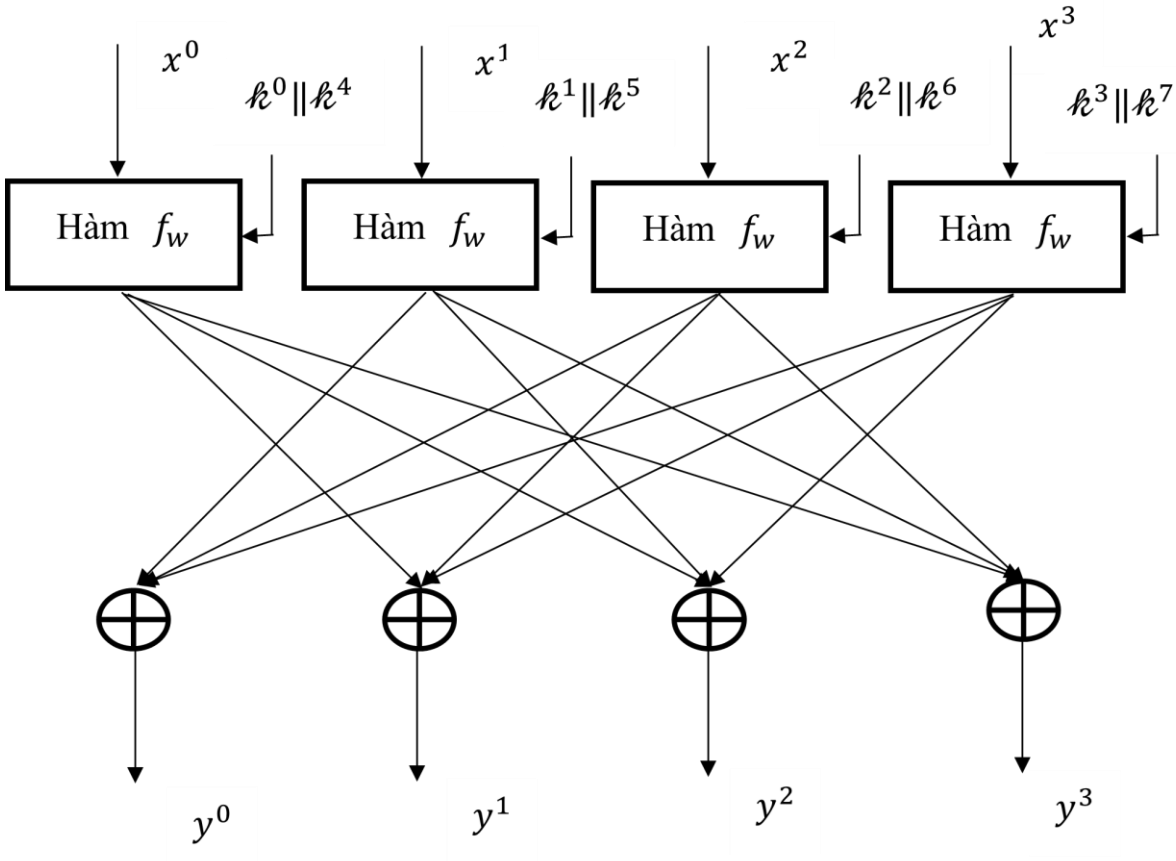
Khi đó, hàm vòng F_l của MKV-l (với hai phiên bản kích cỡ khối là 128 và 256-bit) sẽ có thể được trình bày dưới dạng sử dụng một cấu trúc chung (cấu trúc FLC) với bốn hàm f_0, f_1, f_2, f_3 phụ thuộc khóa thực hiện hoán vị các xâu w-bit, cụ thể đầu vào là trạng thái con $x^i \in V_w$ cùng khóa $\mathcal{K}$ có kích thước $2w$-bit tương ứng cho đầu ra là trạng thái con $y^i \in V_w$, biến đổi được sử dụng trong mỗi vòng của mã khối có kích thước l-bit, có đầu vào bao gồm một xâu trạng thái $X \in V_l$ cùng một khóa vòng $\mathcal{K} \in V_{2l}$ và đầu ra là xâu trạng thái $Y \in V_l$, như sau:

$$F_l: V_{2l} \times V_l \to V_l$$

$$(X, \mathcal{K}) = (\mathcal{k}^0 \| \cdots \| \mathcal{k}^7, x^0 \| x^1 \| x^2 \| x^3) \mapsto Y = y^0 \| y^1 \| y^2 \| y^3 \|$$

trong đó $x^i, y^i, \mathcal{k}^i \in V_w$. Cụ thể, các hàm f_0, f_1, f_2, f_3 được xây dựng từ hàm cơ sở f_w sử dụng các khóa khác nhau như sau:

$$\begin{cases} y^0 = f_w^{\mathcal{k}^1 \| \mathcal{k}^5}(x^1) \oplus f_w^{\mathcal{k}^2 \| \mathcal{k}^6}(x^2) \oplus f_w^{\mathcal{k}^3 \| \mathcal{k}^7}(x^3) \\ y^1 = f_w^{\mathcal{k}^0 \| \mathcal{k}^4}(x^0) \oplus f_w^{\mathcal{k}^2 \| \mathcal{k}^6}(x^2) \oplus f_w^{\mathcal{k}^3 \| \mathcal{k}^7}(x^3) \\ y^2 = f_w^{\mathcal{k}^0 \| \mathcal{k}^4}(x^0) \oplus f_w^{\mathcal{k}^1 \| \mathcal{k}^5}(x^1) \oplus f_w^{\mathcal{k}^3 \| \mathcal{k}^7}(x^3) \\ y^3 = f_w^{\mathcal{k}^0 \| \mathcal{k}^4}(x^0) \oplus f_w^{\mathcal{k}^1 \| \mathcal{k}^5}(x^1) \oplus f_w^{\mathcal{k}^2 \| \mathcal{k}^6}(x^2) \end{cases}$$



Hình 7 - Cấu trúc FLC của hàm vòng F_l.

Như vậy, việc thiết kế mã khối dạng FLC-SDS chỉ còn là xây dựng hai thành phần mật mã là S-hộp 8-bit cho biến đổi và ma trận MDS kích thước phù hợp trên trường $\mathbb{F}_2^8$. Phần tiếp theo sau đây sẽ trình bày chi tiết vấn đề này.

3.3.2.2. S-hộp 8-bit S

Phương pháp xây dựng các S-hộp có kích thước lớn dựa trên một cấu trúc từ các hoán vị kích thước nhỏ đã thực sự phát triển gần đây. Ly do là vì các S-hộp này có các tính chất cài đặt phần mềm tốt với các bảng tính sẵn, có cài đặt dạng bit-slice tốt hơn, đặc biệt phù hợp với mật mã hạng nhẹ với các bảng tra kích thước nhỏ và số lượng các cổng GE được giảm thiểu, hiệu quả cho phương pháp sử dụng mặt nạ chống tấn công kênh kề trong phần cứng. Dựa trên phương pháp tiếp cận này, S-hộp MKV được sinh dựa trên cấu trúc "A" của bài báo [67]. Cụ thể, S-hộp 8-bit $s: \mathbb{F}_{2^8} \to \mathbb{F}_{2^8}$ tương ứng đầu vào có dạng $x = x_l \| x_r \in V_8$ cho đầu ra $y = y_l \| y_r \in \mathbb{F}_{2^8}$ với $x_l, x_r, y_l, y_r \in \mathbb{F}_4$ được xác định bởi công thức sau:

$$y_r = \begin{cases} \pi_1(x_l) \cdot x_r \oplus 1, & x_r \neq 0, \\ \hat{\pi}_1(x_l) \oplus 1, & x_r = 0, \end{cases}$$

$$y_l = \begin{cases} \pi_2(x_r \cdot y_r), & y_r \neq 0, \\ \hat{\pi}_2(x_r), & y_r = 0, \end{cases}$$

trong đó

π_1 = {0x0, 0x1, 0xE, 0x9, 0xB, 0xD, 0x7, 0x6, 0x8, 0x3, 0xA, 0x4, 0xC, 0x5, 0x2, 0xF}

π_2 = {0x0, 0x1, 0xD, 0xB, 0xE, 0x9, 0x6, 0x7, 0xA, 0x4, 0xF, 0x2, 0x8, 0x3, 0x5,0xC}

$\hat{\pi}_1$ = {0x0, 0x1, 0x9, 0xE, 0xD, 0xB, 0x7, 0x6, 0xF, 0x2, 0xC, 0x5, 0xA, 0x4, 0x3, 0x8}

$\hat{\pi}_2$ = {0x0, 0x1, 0x9, 0xE, 0xD, 0xB, 0x7, 0x6, 0xF, 0x2, 0xC, 0x5, 0xA, 0x4, 0x3, 0x8}

Với phép nhân "·" trên trường $\mathbb{F}_4$ được xác định bởi đa thức sinh $f(x) = x^4 \oplus x \oplus 1$, các hoán vị $\pi_1, \pi_2, \hat{\pi}_1, \hat{\pi}_2$ đều có dạng lũy thừa như sau: $\pi_1 = x^{-4}$, $\pi_2 = x^{-2}$, $\hat{\pi}_1 = \hat{\pi}_2 = x^{-1}$. Khi đó, S-hộp có thể biểu diễn dưới dạng sau:

$$y_r = \begin{cases} x_l^{-4} \cdot x_r \oplus 1, & y_i \neq 0, \\ x_l^{-1} \oplus 1, & y_i = 0, \end{cases}$$

$$y_l = \begin{cases} (x_r \cdot y_r)^{-2}, & y_o \neq 0, \\ x_r^{-1}, & y_o = 0, \end{cases}$$

Với dạng này S-hộp s có những lợi thế cài đặt nhất định và đạt được các tính chất mật mã tốt, xem chi tiết kỹ thuật trong [3] và Phụ lục 1.

3.3.2.3. Ma trận MDS

Mã khối MKV có trạng thái được biểu diễn dưới dạng bảng vuông 4×4 hoặc ma trận 8×4 tương ứng trong đó mỗi phần tử nhận giá trị trên trường cơ sở được lựa chọn. Phép MixWords và phép InvMixWords được biểu diễn thực thi thông qua phép nhân ma trận kích thước 4×4 hoặc 8×8 với ma trận trạng thái có kích thước 4×4 đối với mã khối MKV-128 hoặc 8×4 đối với mã khối MKV-256. Để xây dựng một

biến đổi tuyến tính an toàn và hiệu quả, MKV sử dụng ma trận MDS có dạng lũy thừa của ma trận đồng hành cho từng phiên bản kích cỡ khối, với các xem xét về cài đặt chi tiết trong bài báo [2] và Phụ lục 1.

Đối với MKV-128, các ma trận đồng hành có dạng trong [69] như sau:

$$\begin{pmatrix} 0 & 0\text{x}01 & 0 & 0 \\ 0 & 0 & 0\text{x}01 & 0 \\ 0 & 0 & 0 & 0\text{x}01 \\ 0\text{x}01 & L & 0\text{x}01 & L \oplus 0\text{x}01 \end{pmatrix}$$

trong đó lũy thừa bậc 4 của nó là ma trận MDS. Với trường cơ sở $\mathbb{F}_{2^8}$ nêu trên, chọn biến đổi tuyến tính L là phép nhân với phần tử x (bằng 2 trong hệ thập phân). Khi đó, ma trận đồng hành A và A^{-1} được xác định như sau:

$$A_4 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0\text{x}01 & 0\text{x}02 & 0\text{x}01 & 0\text{x}03 \end{pmatrix},$$

$$A_4^{-1} = \begin{pmatrix} 0\text{x}02 & 0\text{x}01 & 0\text{x}03 & 0\text{x}01 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Lũy thừa bậc 4 của chúng xác định các ma trận được sử dụng trong MixWords và InvMixWords như sau $M_4 = A_4^4, M_4^{-1} = A_4^{-4}$.

Đối với các ma trận cho MKV-256, lựa chọn các hệ số cho ma trận đồng hành kích thước 8×8 trên trường $\mathbb{F}_{2^8}$, có dạng như sau:

$$A_8 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0\text{x}01 & 0\text{x}04 & 0\text{xDB} & 0\text{x}0\text{C} & 0\text{x}14 & 0\text{x}0\text{C} & 0\text{xDB} & 0\text{x}04 \end{pmatrix},$$

và nghịch đảo của nó là

$$A_8^{-1} = \begin{pmatrix} 0\text{x}04 & 0\text{xDB} & 0\text{x}0\text{C} & 0\text{x}14 & 0\text{x}0\text{C} & 0\text{xDB} & 0\text{x}04 & 0\text{x}01 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

Lũy thừa bậc 8 của các ma trận này cho chúng ta ma trận MDS kích thước 8×8 trên trường $\mathbb{F}_{2^8}$ với đa thức sinh nguyên thủy $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$, tức là, $M_8 = A_8^8$ và $M_8^{-1} = A_8^{-8}$.

3.3.3. Lược đồ khóa

MKV có thiết kế quá trình tạo các khóa con từ trạng thái khóa có kích thước gấp đôi trạng thái mã dựa trên hàm cập nhật trạng thái khóa $UpdateKS_l$. Hàm này sử dụng một lần lặp lược đồ V trong [6] với hàm vòng sử dụng hai lần lặp hàm vòng F_l của quá trình mã hóa để tận dụng cài đặt và tăng khả năng kháng lại thám mã vi sai khóa quan hệ. Ngoài ra, lược đồ này được lựa chọn để tránh mối quan hệ đơn giản giữa các khóa vòng con được lấy ra từ trạng thái khóa sau mỗi lần cập nhập trạng thái. Trạng thái khóa ban đầu K_0 được khởi tạo được khóa chính $2l$. Trong các phiên bản có độ dài khóa ngắn hơn, thì trạng thái này sẽ được lấy đầu dựa trên một phần phủ định của các khóa chính. Cuối cùng, các hằng số vòng khác nhau được thêm vào vị trí khóa nhằm củng cố độ an toàn của mã khối trước một số tấn công thám mã như tấn công khóa quan hệ, tấn công trượt, …

3.4. Phân tích độ an toàn của thuật toán mã khối MKV

Các kết quả đánh giá độ an toàn của MKV được tổng hợp trong phần này:

3.4.1. Độ an toàn kháng lại thám mã tuyến tính và vi sai

Độ an toàn của MKV-l/k trước thám mã tuyến tính và vi sai được xem xét thông qua Mệnh đề 1, Mệnh đề 2 với sự lựa chọn cụ thể của các thành phần mật mã của mã pháp FLC-SDS. Trong đó, tầng tuyến tính D sử dụng ma trận MDS có kích thước 4×4 trên $\mathbb{F}_{2^8}$ đối với MKV-128 và ma trận MDS có kích thước 8×8 trên $\mathbb{F}_{2^8}$ đối với MKV-256. Khi đó, ta có $\mathrm{Br_d}(D_{32}) = \mathrm{Br}_l(D_{32}) = 5$ và $\mathrm{Br_d}(D_{64}) = \mathrm{Br}_l(D_{64}) = 9$. Từ đó, chúngta nhận được kết quả đánh giá số S-hộp hoạt động vi sai (tuyến tính) của MKV-128 và MKV-256 như sau:

Hệ quả 1. *Hai vòng MKV-128 có ít nhất 20 S-hộp hoạt động vi sai/tuyến tính.*

Hệ quả 2. *Hai vòng MKV-256 có ít nhất 36 S-hộp hoạt động vi sai/tuyến tính.*

Hơn nữa, các kết quả này cũng được kiểm chứng lại bằng cách sử dụng công cụ đánh giá số S-hộp hoạt động dựa trên bài toán tối ưu MILP theo cách tiếp cận của bài báo [71] và đã nhận được các đánh giá cho S-hộp hoạt động cho toàn bộ số vòng phù hợp với các kết quả lý thuyết nhận được. Như vậy, ta có số lượng các S-hộp tối thiểu qua các vòng của mã khối MKV-l đồng thời so sánh với trường hợp vệt lan rộng cho kích thước khối tương ứng, tức là mã khối được thiết kế dạng AES, Kalyna và Kuznyechik như sau:

Bảng 4 - Số lượng các S-hộp hoạt động tối thiểu qua các vòng

STT	Mã khối	1 vòng	2 vòng	3 vòng	4 vòng	5 vòng	6 vòng	7 vòng	8 vòng
Kích thước cỡ 128-bit									
1.	MKV-128	5	20	25	40	45	60	65	80
2.	Mã pháp dạng	1	5	6	25	26	30	31	50

	AES								
3.	Mã pháp dạng Kalyna	1	9	10	28	29	37	38	56
4.	Mã pháp dạng Kuznyechik	1	17	18	34	35	51	52	68
Kích thước cỡ 256-bit									
5.	MKV-256	9	36	45	72	81	108	117	144
6.	Mã pháp dạng AES	1	9	10	45	46	54	55	90
7.	Mã pháp dạng Kalyna	1	17	18	48	49	65	66	96
8.	Mã pháp dạng Kuznyechik	1	32	33	64	65	96	97	128

Với S-hộp s có độ vi sai đều cực đại $(\mathrm{DP})_{\max} = 0{,}0234$ và độ lệch tuyến tính $\epsilon = \frac{22}{128}$, cận trên của các xác suất vi sai và độ lệch tuyến tính qua các vòng của mã khối MKV-128, MKV-256 được tổng hợp trong Bảng 5.

Bảng 5 - Độ phức tạp thám vi sai và tuyến tính lên MKV

Vòng	Số lượng S-hộp hoạt động				
		Thám mã vi sai		Thám mã tuyến tính	
		Cận trên xác suất vi sai	Độ phức tạp thám mã	Cận trên độ lệch tuyến tính	Độ phức tạp thám mã
MKV-128					
1	5	$2^{-27{,}085}$	$2^{27{,}085}$	$2^{-12{,}7}$	$2^{25{,}4}$
2	20	$2^{-108{,}34}$	$2^{108{,}34}$	$2^{-50{,}8}$	$2^{101{,}6}$
3	25	$2^{-135{,}425}$	$2^{135{,}425}$	$2^{-63{,}5}$	2^{127}
4	40	$2^{-216{,}68}$	$2^{216{,}68}$	$2^{-101{,}6}$	$2^{203{,}2}$
5	45	$2^{-243{,}765}$	$2^{243.765}$	$2^{-114{,}3}$	$2^{228{,}6}$
6	60	$2^{-325{,}02}$	$2^{325{,}02}$	$2^{-152{,}4}$	$2^{304{,}8}$
7	65	$2^{-352{,}105}$	$2^{352{,}105}$	$2^{-165{,}1}$	$2^{330{,}2}$
8	80	$2^{-433{,}36}$	$2^{433{,}36}$	$2^{-203{,}2}$	$2^{406{,}4}$
MKV-256					
1	9	$2^{-48{,}753}$	$2^{48{,}753}$	$2^{-22{,}86}$	$2^{45{,}72}$
2	36	$2^{-195.012}$	$2^{195.012}$	$2^{-91{,}44}$	$2^{182{,}88}$
3	45	$2^{-243{,}765}$	$2^{243{,}765}$	$2^{-114{,}3}$	$2^{228.6}$
4	72	$2^{-390{,}024}$	$2^{390{,}024}$	$2^{-182{,}88}$	$2^{365{,}76}$
5	81	$2^{-438{,}777}$	$2^{438{,}777}$	$2^{-205{,}74}$	$2^{411{,}48}$

6	108	$2^{-585.036}$	$2^{585.036}$	$2^{-274,32}$	$2^{548,64}$
7	117	$2^{-633,789}$	$2^{633,789}$	$2^{-297,18}$	$2^{594,36}$
8	144	$2^{-780,048}$	$2^{780,048}$	$2^{-365,76}$	$2^{731,52}$

Như vậy, mã khối MKV-128 với số vòng 3 sẽ an toàn đối với thám mã vi sai và với số vòng 4 sẽ an toàn với thám mã tuyến tính. Để tấn công với các phiên bản khoá 192-bit, 256-bit, số vòng tăng thêm đảm bảo an toàn mong muốn. Còn đối với MKV-256 với số vòng 4 đạt được độ an toàn đối với cả thám mã tuyến tính và vi sai. Hơn nữa, số vòng bổ sung sẽ đảm bảo độ an toàn cho các phiên bản khoá dài hơn (384-bit, 512-bit). Bảng 6 tổng hợp được số vòng và hành lang an toàn đảm bảo kháng lại hai thám mã quan trọng này đối với mã khối MKV-l/k với tùy chọn kích thước khối và độ dài khóa như sau:

Bảng 6 - Số vòng và hành lang an toàn của MKV-l/k trước thám mã tuyến tính và vi sai

STT	Kích cỡ khối l	Độ dài khóa k	Số vòng đủ đảm bảo độ an toàn	Hành lang an toàn
Thám mã vi sai				
1.	128	128	3	3
2.		192	4	3
3.		256	6	2
4.	256	256	3	3
5.		384	5	2
6.		512	6	2
Thám mã tuyến tính				
1.	128	128	4	2
2.		192	4	3
3.		256	6	2
4.	256	256	4	2
5.		384	5	2
6.		512	6	2

3.4.2. Độ an toàn kháng lại thám mã boomerang

David Wagner đã công bố tấn công Boomerang vào năm 1999 [72] và sử dụng nó để phá vỡ mã khối COCONUT98. Tấn công xem xét một mã pháp E dưới dạng kết hợp liên tiếp của hai mã pháp con, tức là $E = E_2 \circ E_1$ (Hàm giải mã D được chia thành $D = D_1 \circ D_2$). Khi đó, hai vi sai cần đến là một vi sai tốt với đầu vào α và đầu ra β đối với quá trình mã hóa E_1 có xác suất là p, và một vi sai tốt, đối với quá trình giải mã của E_2 tức là D_2 có . Hai vi sai này "gặp nhau" tại giữa của mã pháp E và cùng nhau để phân tích đầy đủ mã pháp. Dưới giả thiết hai vệt vi sai là độc lập, tấn công này khai thác xác suất cao của tính chất vi sai sau:

$$\Pr[E^{-1}(E(P) \oplus \Delta^*) \oplus E^{-1}(E(P \oplus \Delta) \oplus \Delta^*) = \Delta] = p^2q^2$$

Bốn bản rõ $P, P^*\hat{P}, \hat{P}^*$ thỏa mãn được gọi là một bộ bốn. Nếu hai vi sai này có xác suất tương ứng là p và q, xác suất của bộ bốn này là p^2q^2. Khi một bộ bốn được tìm ra trường hợp này được mô tả ở Hình 8.1, trong đó các vi sai tìm kiếm đối với E_1 và D_2 được chỉ ra. Tại điểm này một tấn công vi sai có thể được triển khai lên mã pháp. Chú ý rằng đây là dạng vi sai bậc 2 dạng $0 \to 0$. Thật vậy:

$$0 = \Delta \oplus \Delta = \left(\hat{P}^* \oplus \hat{P}\right) \oplus (P^* \oplus P)$$

$$\to$$

$$\left(\hat{C}^* \oplus \hat{C}\right) \oplus (C^* \oplus C) = \left(\hat{C}^* \oplus C^*\right) \oplus \left(\hat{C} \oplus C\right) = \nabla \oplus \nabla = 0.$$

Trong [73], các tác giả đã chỉ ra điều kiện để mã khối an toàn trước tấn công Boomerang như sau $p^2q^2 \leq 2^{-l}$, nhắc lại l là kích thước khối.

Áp dụng vào mã khối đề xuất, Chúng ta đã xem xét đánh giá cho các trường hợp mã khối MKV-l/k cho trường hợp số vòng rút gọn. Trong trường hợp số vòng rút gọn r, ta có thể xem xét các trường hợp số vòng có thể của các mã pháp con E^1, E^2, sao cho $E^r = E^1 \cdot E^2$. Kết hợp với các kết quả đánh giá cận trên đối với các xác suất đặc trưng vi sai qua các vòng, chúng ta nhận được chi tiết các cận dưới độ phức tạp của tấn công này như sau:

Bảng 7 - Độ phức tạp của tấn công Boomerang lên các vòng rút gọn của MKV

Mã khối	Số vòng thám mã r	Số vòng của mã pháp con		Cận trên xác suất đặc trưng vi sai qua số vòng r		Cận trên giá trị $(pq)^2$	Cận dưới độ phức tạp tấn công
		E^1	E^2	Cho E^1, p	Cho E^2, q		
MKV-128	2	1	1	$2^{-27,085}$	$2^{-27,085}$	$2^{-108,34}$	$2^{108,34}$
	3	1	2	$2^{-27,085}$	$2^{-108,34}$	$2^{-270,85}$	$2^{270,85}$
MKV-256	2	1	1	$2^{-48,75}$	$2^{-48,75}$	2^{-195}	2^{195}
	3	1	2	$2^{-48,75}$	$2^{-194,4}$	$2^{-487,5}$	$2^{487,5}$

Như vậy, cả hai phiên bản khối 128-bit, 256-bit mã khối MKV-l với số vòng lớn hơn 2 an toàn trước tấn công Boomerang.

3.4.3. Độ an toàn kháng lại thám mã tích phân

Thám mã tích phân (integral cryptanalysis) [74] là một dạng thám mã vi sai bậc cao [75] được trình bày đầu tiên như một đối ngẫu của thám mã vi sai và nó được biết đến là tấn công tốt nhất lên mã khối AES nói riêng, các mã khối hướng từ nói chung. Tấn công tích phân khai thác một tính chất đặc biệt các phép biến đổi trong trường $\mathbb{F}_{2^8}$.

Sau đây, nhóm nhóm thiết kế trình bày tấn công theo ý tưởng này lên MKV. Giả sử tập Λ là một tập 256 trạng thái[19]. Khi đó, ta có các khái niệm như sau:

- các byte trạng thái khác với mọi byte trạng thái khác gọi là các byte *chủ động*.
- các byte trạng thái mà bằng với mọi byte trạng thái khác gọi là các byte *thụ động*.

Ta có: $\forall x_{i,j} \in \Lambda, 1 \leq i,j \leq 4$:

$$\begin{cases} x_j^i \neq x_l^k, \forall 1 \leq k,l \leq 4, (k,l) \neq (i,j) \text{ được gọi là chủ động} \\ x_j^i = x_l^k, \forall 1 \leq k,l \leq 4, (k,l) = (i,j) \text{ được gọi là thụ động} \end{cases}$$

Nếu ta xét một tập Ω gồm 256 bản rõ mà chúng chỉ khác nhau ở 1 vị trí byte (vị trí đó là byte chủ động). Khi đó các byte trạng thái đầu vào (bản rõ) (có giá trị nằm trong tập Λ) hoặc là hằng số hoặc thay đổi trên tất cả các giá trị có thể khác nhau, ta có:

$$\oplus_{x \in \Omega} x_{i,j} = 0, \forall i,j$$

Một đặc điểm quan trọng là khi byte đầu vào của hộp thế thay đổi trên mọi giá trị trong tập Λ thì byte đầu ra cũng thay đổi trên mọi giá trị của tập Λ. Việc áp dụng các bước thay thế S-hộp hoặc Cộng XOR khóa lên các byte trạng thái của tập Ω sẽ cho một tập Ω khác nhưng vị trí của các byte chủ động không thay đổi (vị trí của các byte không bị ảnh hưởng bởi 2 bước này). Việc áp dụng biến đổi chuyển vị cho kết quả trong tập Ω nhưng các byte chủ động bị chuyển vị bởi phép biến đổi này.

Đối với MKV-l, chúng ta xem xét chi tiết sự lan truyền của các byte chủ động và sự bảo toàn của tính chất cân bằng của các từ này qua các phép biến đổi của mã pháp. Rõ ràng, các phép biến đổi cộng khóa và S_w không làm thay đổi các byte chủ động và bảo toàn tính chất cân bằng, còn phép MixWords sẽ biến đổi các byte chủ động lan ra cho toàn bộ trạng thái con (được biểu diễn bởi cột trong ma trận trạng thái).

Cụ thể, ta xem xét sự lan truyền này qua các bước được minh họa trong Hình 8 như sau: Xét một bản rõ $P = \left(p_j^i\right)_{4\times t} \in V_l$ có một byte chủ động (không mất tổng quát giả sử là p_0^0, còn các byte còn lại là hằng số) và đầu ra tương ứng với số vòng $C = \left(c_j^i\right)_{4\times t} \in V_l$. Khi đó, các byte chủ động khi qua bước 1.1 được giữ nguyên, Bước 1.2 trạng thái con chứa byte chủ động được lan ra toàn bộ, Bước 1.3 trạng thái các byte chủ động được giữ nguyên và cuối cùng từ một trạng thái con chủ động được lan truyền sang 3 trạng thái con chủ động tại bước 1.4. Tóm lại, trạng thái đầu ra của Vòng 1, $X \in V_l$ được tính như sau:

[19] Nghĩa là 256 giá trị có thể nhận của một byte dữ liệu.

$$X = \mathrm{XWords}\Big(\mathrm{SubCells}\big(\mathrm{MixWords}\big(\mathrm{SubCells}(P \oplus \mathcal{K}_0^0)\big) \oplus \mathcal{K}_0^1\big)\Big)$$

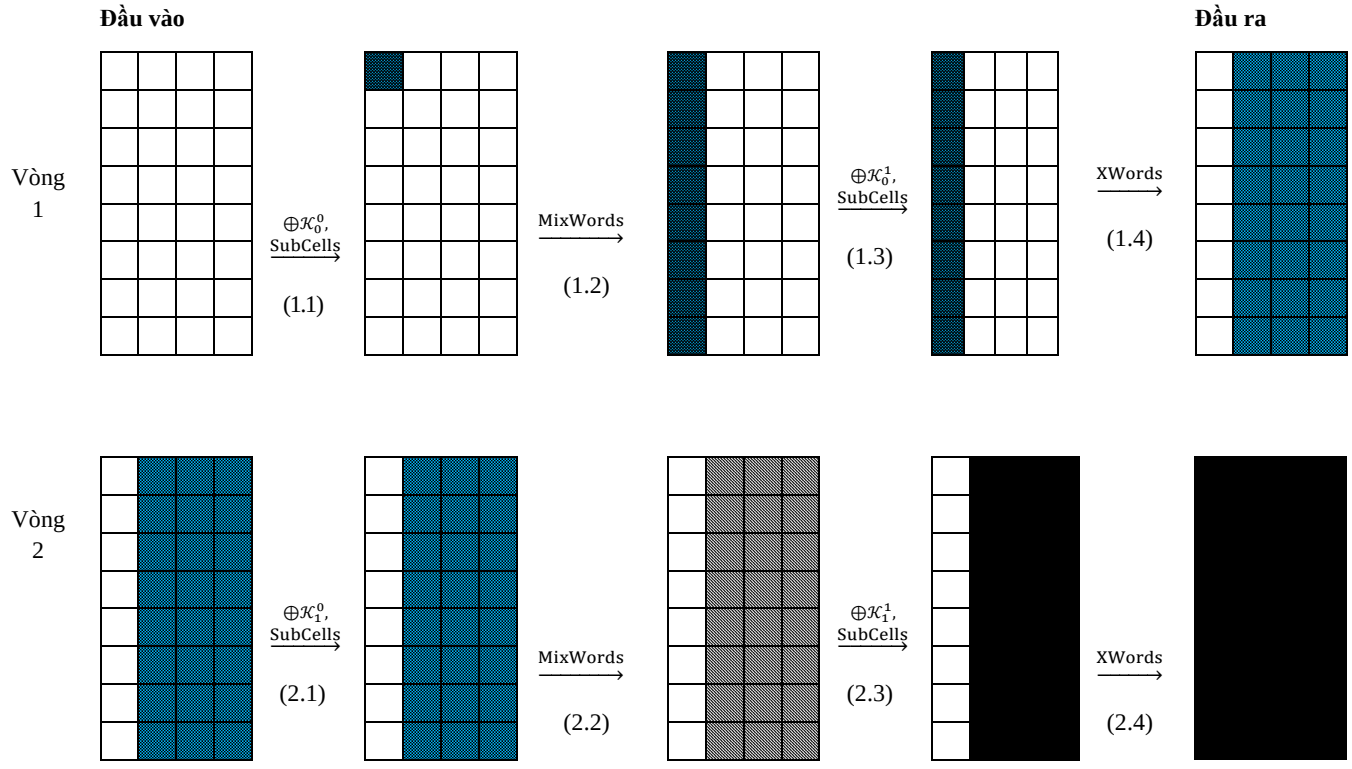
với các trạng thái con X^0, X^1, X^2, X^3 của X thỏa mãn:

- Trong trạng thái con X^0, các từ byte đều có giá trị là hằng số c, tức là $x_j^0 = c$ với $0 \leq j \leq t$.

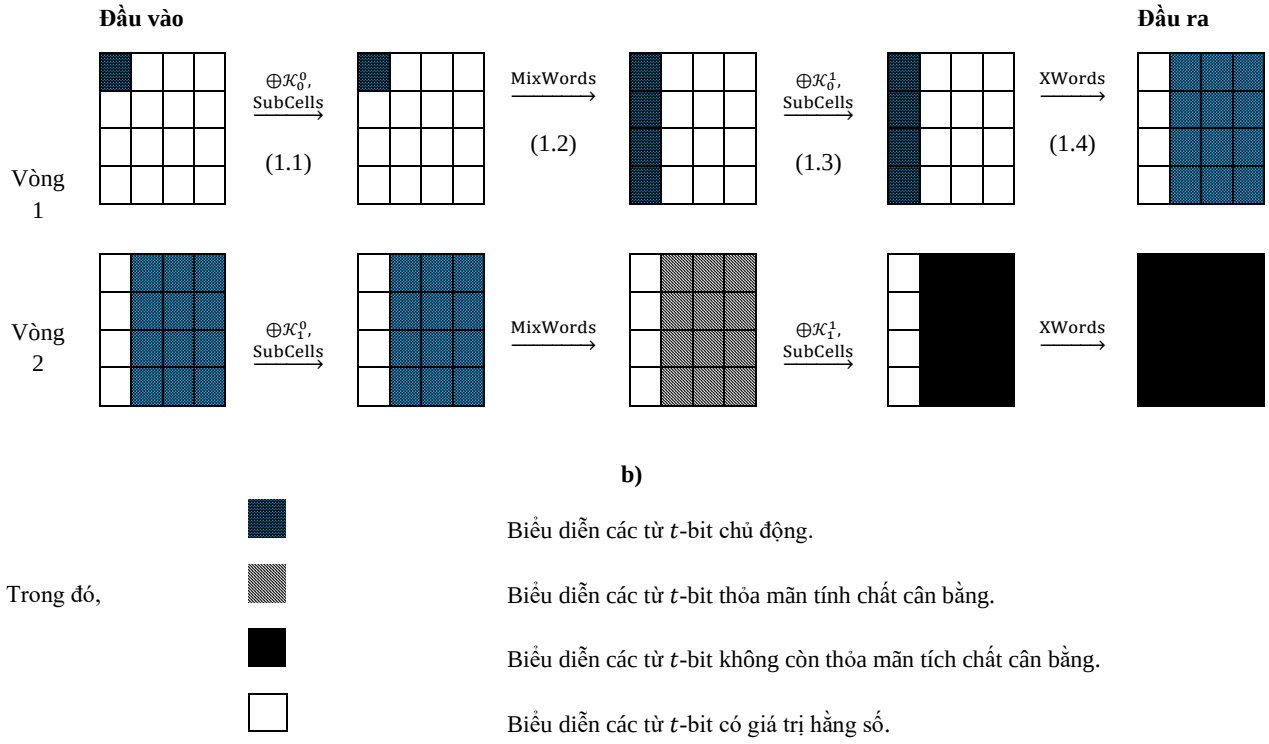
- Trong các trạng thái con X^1, X^2, X^3, các byte bị ảnh hưởng bởi byte chủ động p_0^0 nên có giá trị được xác định như sau:

$$\begin{aligned} x_j^i &= s\big(m_{j,0} \cdot s(p_0^0 \oplus \hbar_0^0) \oplus \bigoplus_{k=1}^{t} m_{j,k} \cdot c \oplus \hbar_0^4\big) \\ &= s\big(m_{j,0} \cdot s(p_0^0 \oplus \hbar_0^0) \oplus c' \oplus \hbar_0^4\big) \end{aligned}$$

với $c' = \bigoplus_{k=1}^{t} m_{j,k} \cdot c$ và $m_{i,j}$ là các hệ số của ma trận M xác định phép biến đổi MixWords. Như vậy, khi p_0^0 là một từ chủ động, tức là lần lượt nhận các giá trị từ $0 \ldots 2^8 - 1$, thì x_j^i vẫn giữa được tính chất của từ chủ động. Lưu ý rằng, đối với mỗi trạng thái con X^1, X^2, X^3 là bằng nhau do tính chất của phép XWords, song các giá trị bên trong của mỗi ô khác nhau, tức là $x_j^i \neq x_{j'}^i$, với $j \neq j'$.



a)



Hình 8 - Sự lan truyền của các ô chủ động và tính chất cân bằng trong thám mã tích phân lên 2 vòng mã pháp MKV-256 (a) và MKV-128 (b).

Các byte chủ động này vẫn được bảo đảm khi trạng thái đi qua phép cộng XOR khóa $\mathcal{K}_0^1$ và SubCells tại Bước 2.1. Đến Bước 2.2, các byte chủ động qua biến đổi MixWords thì tính chất chủ động này bị mất đi, tuy nhiên vẫn đảm bảo tính cân bằng. Thật vậy, giả sử đầu vào và đầu ra của MixWords tương ứng là Y thỏa mãn $y_j^0 = \text{const}, 0 \le j \le t-1$; và y_j^i là các byte chủ động với $1 \le i \le 3, 0 \le j \le t-1$ trong đó $y_j^i \ne y_{j'}^i$, với $j \ne j'$. Khi đó, chúng ta xét trạng thái đầu ra Z

$$z_j^i = \bigoplus_{t=1}^{7} m_{j,t} \cdot y_t^i$$

Với các p_0^0 lần lượt hết các giá trị $0 \dots 2^8 - 1$ sẽ không đảm bảo được z_j^i không nhận được lần lượt các giá trị trên tập V_t, như vậy tính chủ động không còn. Tuy nhiên, tính cân bằng vẫn được đảm bảo,

$$\bigoplus_{p_0^0 \in V_t} z_j^i = \bigoplus_{k=1}^{t} m_{j,t} \cdot \bigoplus_{p_0^0 \in V_8} y_j^i = \bigoplus_{k=1}^{t} m_{j,k} \cdot 0 = 0$$

Cuối cùng, tính cân bằng đối với các byte này không được đảm bảo khi qua phép biến đổi cộng XOR khóa $\mathcal{K}_1^1$, SubCells, XWords tại các Bước 2.3 và 2.4 của vòng 2.

Khi đó, thuật toán tấn công tích phân lên 2 vòng của mã pháp MKV-l được thực hiện qua các bước sau:

Tấn công tích phân cơ bản mã pháp MKV-l 2 vòng:

1. Tạo ra tập Ω gồm 2^8 bản rõ đặc biệt chỉ có một byte chủ động, tính các bản mã tương ứng với từng bản rõ này.
2. Thực hiện việc tuần tự duyệt từng byte khóa vòng con $\mathcal{K}_3^0$ tại vòng cuối (vòng 3) sao cho khi giải mã ở vòng cuối nhận được tập trạng thái mà tổng theo modulo 2 (XOR) bằng 0 trong các byte tương ứng của trạng thái.
3. Khóa vòng con thứ hai tại vòng thứ 2 sẽ ở bước 2 cho phép rút gọn tấn công xuống khôi phục khóa con $\mathcal{K}_2^1$ của vòng 2.

Như vậy, tấn công lên 2 vòng yêu cầu xấp xỉ $2 \times 2^8 = 2^9$ phép mã hóa của 2 vòng (bỏ qua độ phức tạp của việc tra S-hộp và sử dụng hai tập Ω) khi khôi phục được một byte của khóa $\mathcal{K}_1^1$ cho tấn công cơ bản lên 2 vòng của mã pháp MKV-l. Hơn nữa, để khôi phục toàn bộ khóa $\mathcal{K}_2^1$, chúng ta cần $2^{4t} \times 2^9 = 2^{4t+9}$.

Khi mở rộng thêm cho 0,5 vòng về phía sau đối với hai vòng cơ bản để có tấn công cho 2,5 vòng nhằm khối phục khóa $\mathcal{K}_3^0$, do một byte sẽ ảnh hưởng tới toàn bộ byte của trạng thái con này khi qua biến đổi MixColumns tiếp theo, ta phải vét cạn khóa các vị trí từ ảnh hưởng của khóa $\mathcal{K}_3^0$ do đó độ phức tạp tính toán của tấn công lên 2,5 vòng sẽ là $2^{4t+9} \times 2^w$. Đối với khóa $\mathcal{K}_3^1$, ta phải vét cạn cho ba trạng thái con tương ứng với trạng thái con của từ khóa đang xét là $2^{4t+9} \times 2^w \times 2^{3w}$ với độ phức tạp dữ liệu là $2 \times 2^8 \times l$-bit. Mở rộng sang khóa $\mathcal{K}_4^1$, ta phải vét cạn toàn bộ khóa do khi đó 3 từ ở vòng trước đã ảnh hưởng toàn bộ các từ đầu ra của 4 vòng. Chi tiết xem minh họa tại Bảng 8.

Bảng 8 - Độ phức tạp của thám mã tích phân lên MKV.

Tấn công lên phiên bản rút gọn vòng MKV-l	Độ phức tạp tính toán	Độ phức tạp dữ liệu	Khóa khôi phục
2-vòng cơ bản	2^{t+5}	$2 \times 2^t \times l$-bit	$\mathcal{K}_0, \mathcal{K}_1,$
2,5-vòng	$2^{t+5} \times 2^w$	$2 \times 2^t \times l$-bit	$\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2^0$
3-vòng	$2^{t+5} \times 2^{4w}$	$2 \times 2^t \times l$-bit	$\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2$
3,5-vòng	$2^{t+5} \times 2^{8w}$	$2 \times 2^t \times l$-bit	$\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3^0$

Tương tự, khi mở rộng thêm về phía trước đối với 2,5 vòng cơ bản, chúng ta cần xem xét các từ t-bit có ảnh hướng đến từ chủ động được xem xét ở đầu vào của tấn công cơ bản. Đối với phép biến đổi XWords, sẽ có 3 từ t-bit trong 3 trạng thái con (có biểu diễn cột trong hình vẽ mô tả) ảnh hưởng nhưng vẫn giữ được tính chủ động, ta tiếp thụ đến biến đổi P_w^{-1} thì trong mỗi trạng thái con có từ chủ động sẽ bị ảnh hưởng toàn bộ. Khi đó, độ phức tạp tấn công mở rộng thêm 1 vòng về phía trước sẽ vẫn cần $2^{4t+9} \times 2^w \times 2^{3w}$ với độ phức tạp dữ liệu là $2 \times 2^t \times l$-bit. Tóm lại, mã khối MKV-l/k từ bốn vòng trở lên an toàn trước tấn công tích phân.

3.4.4. Độ an toàn kháng lại thám mã đại số

Nguyên lý cơ bản của các tấn công đại số xuất phát từ kết quả của Shannon: "*toàn bộ hệ mật có thể biểu diễn như là một hệ các phương trình đại số đa biến lớn, mà có thể giải ra để tìm khóa bí mật*". Mỗi phương trình đại số có thể được xem như một đa thức trên các bit của khóa bí mật và đặt bằng 0. Trong loại tấn công này, kẻ tấn công tìm một hệ lớn các phương trình đa biến trên các khóa bí mật. Sau đó họ cố gắng giải hệ phương trình lớn này để thu được khóa bí mật hoặc giảm bớt không gian tìm kiếm. Về căn bản, tấn công này là tấn công bản rõ đã biết (một số trường hợp nó là tấn công bản rõ chọn lọc) và nó giống dạng đại số hơn là dạng thống kê. Tính hiệu quả của tấn công này phụ thuộc vào tính hiệu quả của thuật toán sinh và giải hệ các phương trình đa biến lớn. Các tấn công đại số được thực hiện trong hai bước chính:

- Tạo ra một tập lớn các phương trình đa biến trên các khóa bí mật.
- Giải hệ các phương trình được tạo trên để thu được khóa bí mật thực sự hoặc giảm bớt không gian tìm kiếm cho khóa.

Tại hội nghị FSE năm 2005, trong bài báo "*Small Scale Variants of the AES*" nhóm tác giả Robshow đã xem xét khả năng kháng lại tấn công đại số của các mã pháp tựa AES được mô tả dưới dạng bảng. Trong đó các tác giả đã xét một mã pháp SR*(n, r, c, e) trong đó n là số vòng mã hóa, r là số lượng các dòng trong bảng mô tả của đầu vào, c là số lượng các cột trong bảng mô tả của đầu vào, e là kích cỡ của một từ với các bước giống như AES (xem [76]) . Như vậy, AES-128 tương đương với SR*(10, 4, 4, 8), còn mã khối MKV-l với số vòng r tương đương $SR^*(2r, 4, t, 8)$ (trong đó $t = 4$ đối với phiên bản MKV-128, còn $t = 8$ đối với phiên bản MKV-256). Trong đó, các tác giả đã đưa ra các áp dụng thuật toán giải hệ phương trình như F4 và Grobner cho các thuật toán dạng AES này. Qua các đánh giá và thực nghiệm của các tác giả, $SR^*(2r, 4, t, 8)$ không thể tấn công thực hành được bằng phương pháp đại số với số vòng lớn. Hiện nay, cũng chưa có tiến bộ nào trong việc cải tiến đột phát các kết quả này về phương pháp giải cũng như công cụ cho thám mã đại số lên mã khối. Phần tiếp theo trình bày các kết quả thống kê các phương trình và số lượng các biến cần để mô tả thuật toán MKV. Các S-hộp được đề xuất sử dụng trong MKV-l được lựa chọn các S-hộp có tính chất mật mã tối ưu về mối quan hệ đại số; tuy nhiên để mô tả tổng quát, chúng sẽ cần số lượng các phương trình độc lập tuyến tính gồm e_2 phương trình bậc 2 và e_3 phương trình bậc 3 độc lập tuyến tính cùng 16 biến (8 biến đầu vào và 8 biến đầu ra) để mô tả trên trường $\mathbb{F}_2$. Trong mỗi hàm vòng F_l dạng SDS, dựa trên các biến đổi cơ bản chúng ta sẽ có đánh giá về số lượng phương trình đại số và các biến cần mô tả chúng như sau:

- Tầng S-hộp thứ nhất, thứ hai đều sử dụng $4t$ S-hộp nên cần $2te_2$ phương trình bậc 2 và $4te_3$ phương trình bậc 3 của $16t$ biến.
- Tầng tuyến tính ở giữa sẽ cần l phương trình bậc nhất giữa các biến của tầng S-hộp thứ nhất và S-hộp thứ hai (biến đầu ra của tầng 1 và biến đầu vào của tầng

2). Hơn nữa, đối với tầng tuyến tính giữa S-hộp thứ 2 với vòng tiếp theo, ta cần l phương trình bậc nhất giữa các biến đầu ra của tầng S-hộp thứ 2 này với đầu vào của vòng tiếp theo. Bổ sung thêm $2l$ biến trung gian để mô tả cho đầu ra của S-hộp tầng thứ nhất và S-hộp tầng thứ hai.

Như vậy, chúng ta sẽ cần tổng cộng sẽ là $2l$ phương trình bậc nhất, $4te_2$ phương trình bậc hai, $4te_3$ phương trình bậc 3; cùng $4l$ biến (đầu vào, đầu ra) để mô tả hàm vòng F_l. Nếu tính thêm cả lược đồ khóa, để biểu diễn một lần UpdateKS_l, chúng ta cần mỗi lần update trạng thái khóa chúng ta phải 2 lần hàm vòng SDS, trong đó ta có thể bỏ biến đổi chuyển vị và cộng hằng số. Cụ thể, tổng số các biến và phương trình mô tả một vòng của các mã pháp như sau:

Bảng 9 - Số lượng biến và phương trình cần mô tả

STT	Hàm	Số lượng biến	Số lượng phương trình		
			Phương trình bậc 1	Phương trình bậc 2	Phương trình bậc 3
1.	F_l	$4l$ biến	$2l$	$4te_2$	$4te_3$
2.	UpdateKS_l	$16l$ biến	$8l$	$16te_2$	$16te_3$
3.	1 vòng MKV-l	$20l$ biến	$10l$	$20te_2$	$20te_3$

Đối với mã khối MKV-l, các S-hộp 8-bit được lựa chọn với các tính chất tối ưu nhất của các hộp thế cùng kích cỡ. Cụ thể, S-hộp này có tối đa số lượng các phương trình biểu diễn bậc 3 độc lập tuyến tính là $e_3 = 441$ (xem [77],[4]) trong khi không có phương trình bậc 2 độc lập tuyến tính $e_2 = 0$. Do đó, số lượng biến và phương trình mô tả này nhiều hơn so với các mã khối AES và Kuznyechik đối với phiên bản cùng mức an toàn. Như vậy, mã khối MKV-l/k với các phiên bản có số vòng 6,7,8 an toàn đối với thám mã đại số.

Bảng 10 - Số lượng biến, phương trình mô tả của MKV

STT	Mã khối	Số lượng biến	Số lượng phương trình		
			Bậc 1	Bậc 2	Bậc 3
1.	MKV-128/128	13.312	9.600	0	176.400
2.	MKV-128/192	15.872	11.520	0	211.680
3.	MKV-128/256	18.432	13.440	0	246.960
4.	MKV-256/256	26.624	19.200	0	352.800
5.	MKV-256/384	31,.744	23.040	0	423.360

6.	MKV-256/512	36.864	26.880	0	493.920

Hiện nay, cũng chưa có tiến bộ nào trong việc cải tiến đột phá về phương pháp giải cũng như công cụ giải hệ phương trình cho thám mã đại số lên mã khối. Do đó, mã khối MKV-l/k với số vòng 6, 7, 8 an toàn đối với thám mã đại số.

3.4.5. Độ an toàn kháng lại thám mã vi sai không thể

Tấn công dựa trên việc tìm kiếm một vi sai không thể xảy ra (qua một phần rút gọn) của một mã khối dựa trên một sự kiện không bao giờ xảy ra nếu khoá được dự đoán sai. Bản chất của sự kiện này là của một vi sai, tức là thông thường lấy điều kiện trên sai khác đầu ra của một sai khác đầu vào cho trước với một số vòng nhất định. Sau đó, dự đoán các bit khoá bí mật và chúng dẫn đến một sự kiện không thể xảy ra được loại bỏ. Một sự khác nhau quan trọng đối với thám mã vi sai cổ điển đó là trong khi trong thám mã vi sai cổ điển, các bit khoá có liên quan được tính rõ ràng hoặc bị hạn chế, thì trong thám mã vi sai không thể chúng được xác định bằng phép loại trừ. Tấn công này nhanh hơn tấn công vét cạn khi chỉ một số bit khoá đặc biệt được dự đoán chính xác để kiểm tra việc xảy ra cả sự kiện và các bit khoá còn lại là không thích hợp.

Đối với mã pháp MKV-256, các vi sai trung gian không thể xảy ra khi tại các biến đổi tuyến tính MixWords hoặc XWords. Trong đó, đối với lỗi ở giữa tại biến đổi MixColumns giống như các mã khối dạng SPN (như AES, Kalyna, …), chúng ta sẽ dựa trên tính chất số nhánh của ma trận MKV tuy nhiên, cách tiếp cận này sẽ gặp khó khăn với sự tham gia của biến đổi XWords khiến cho lan truyền xảy ra với xác suất không chắc chắn trước hoặc sau đó. Do đó, nhóm đề tài sử dụng phương pháp tiếp cận tương tự đối với Feistel (trong bài báo [79]) để đưa ra một vi sai không thể sau hai vòng (mở rộng được cho 2,5 vòng). Cụ thể, ta xét một sai khác đầu vào $\Delta X = \left(\Delta x_j^i\right) \in V_8^{32}$ chỉ có byte hoạt động (không mất tổng quát, giả sử là $\Delta x_0^0 \neq 0$) còn lại các byte bằng không, tức là $\Delta x_j^i = 0, (i,j) \neq (0,0)$; và sai khác đầu ra $\Delta Y = \left(\Delta y_j^i\right) \in V_8^{32}$ thỏa mãn sai khác các từ $\Delta Y^3 \neq 0$ và $\Delta Y^1 = \Delta Y^2 = \Delta Y^3 = 0$. Khi đó, *vi sai $\Delta X \to \Delta Y$ là vi sai không thể đối với 2 vòng của mã pháp MKV.*

Thật vậy, chúng ta xem xét sự lan trường của byte hoạt động của sai khác ΔX khi qua vòng thứ nhất. Giả sử sai khác đầu ra của vòng thứ nhất là ΔZ, ta có:

$$\Delta Z = \mathrm{XWords}\left(\mathrm{SubCells}\left(\mathrm{MixWords}\left(\mathrm{SubCells}(\Delta X)\right)\right)\right)$$

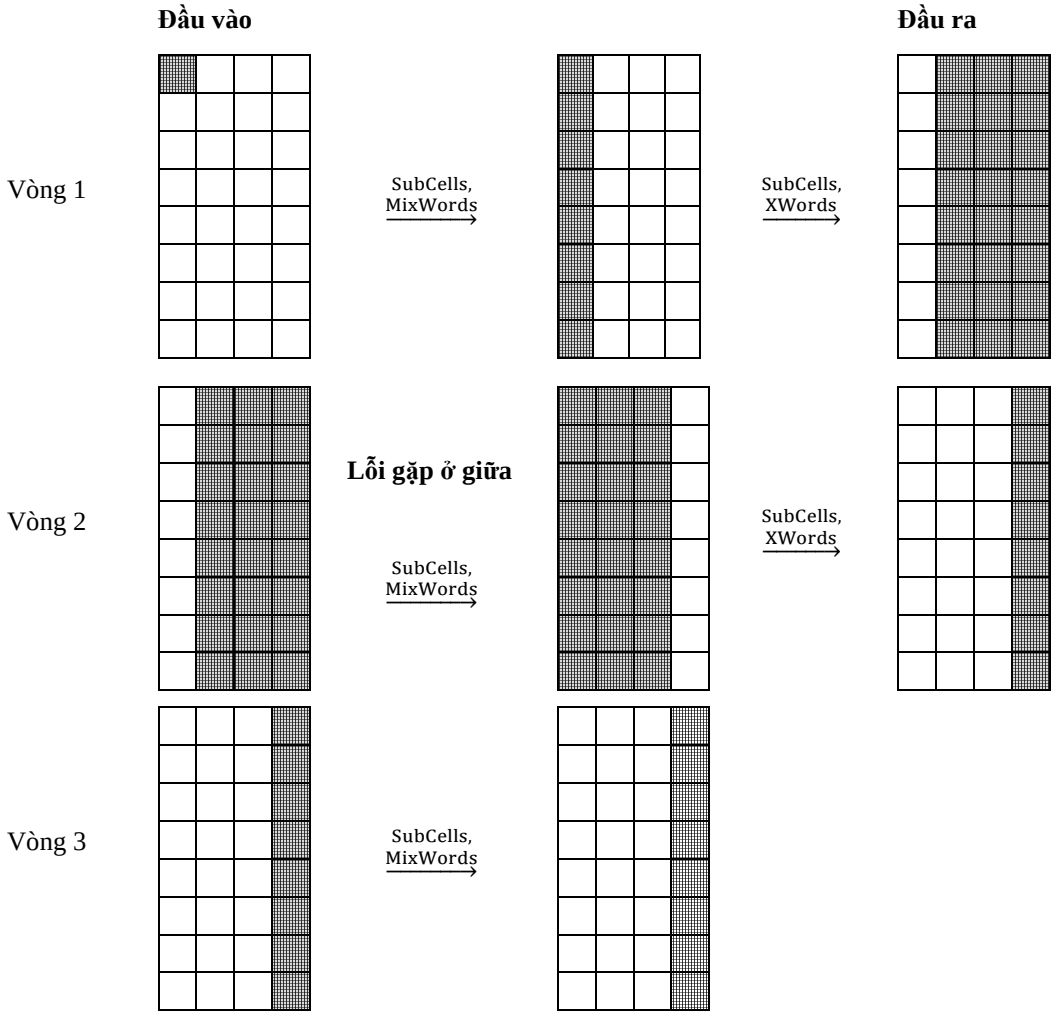
Biến đổi SubCells không làm thay đổi các byte hoạt động, sau đó các byte hoạt động này được lan truyền khi qua biến đổi MixWords cho toàn bộ trạng thái con đầu tiên, và kích hoạt 3 trạng thái con khi qua biến đổi XWords. Như vậy, sai khác $\Delta Z = (\Delta Z^0, \Delta Z^1, \Delta Z^2, \Delta Z^3)$ trong đó các trạng thái con có dạng $\Delta Z^0 = 0, \Delta Z^1 = \Delta Z^2 = \Delta Z^3 \neq 0$.

Xét sự biến đổi ngược của sai khác đầu ra ΔY cho tới gặp đầu vào của vòng thứ hai, gọi sai khác này là $\Delta Z'$, ta có:

$$\Delta Z' = \text{invSubCells}\left(\text{invMixWords}\left(\text{invSubCells}\left(\text{XWords}(\Delta Y)\right)\right)\right)$$

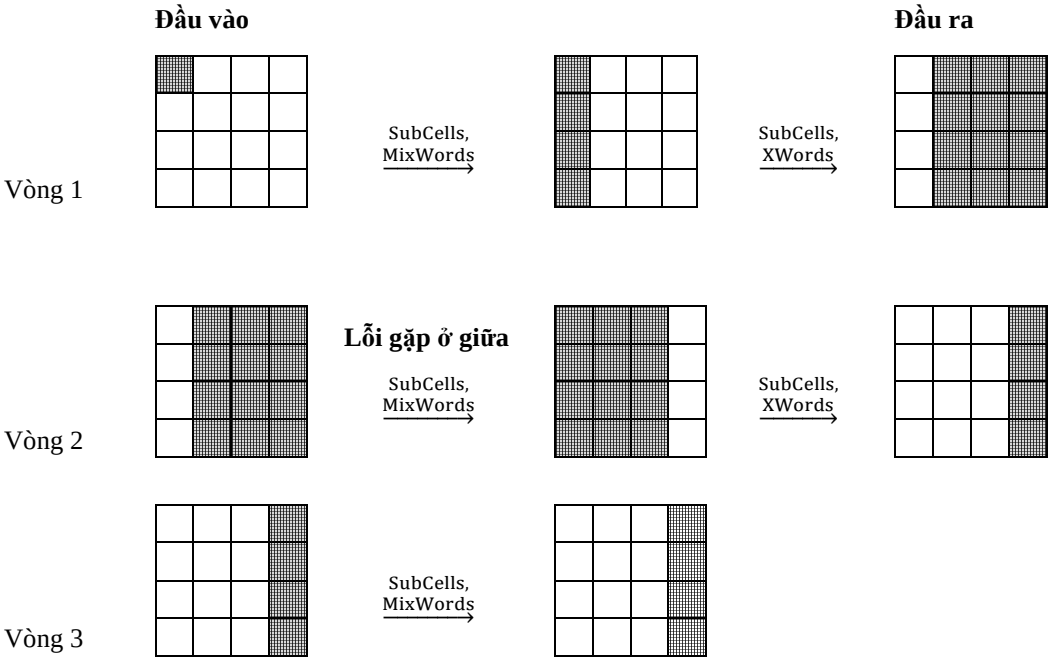
Tương tự, các biến đổi invSubCells không làm thay đổi các ô (cells) hoạt động, sau đó các ô (cell) hoạt động trong trạng thái con được lan truyền thành 3 trạng thái con hoạt động qua biến đổi invXWords, và trạng thái hoạt động này được giữ nguyên khi qua phép biến đổi invMixWords. Khi đó, sai khác $\Delta Z' = \left(\Delta Z'^0, \Delta Z'^1, \Delta Z'^2, \Delta Z'^3\right)$ thỏa mãn $\Delta Z'^0 = \Delta Z'^1 = \Delta Z'^2 \neq 0$ còn $\Delta Z'^3 = 0$.

Như vậy, ta có lỗi gặp ở giữa khi $\Delta Z = \Delta Z'$ với xác suất không. Do đó, vi sai $\Delta X \to \Delta Y$ là vi sai không thể đối với 2 vòng của mã pháp MKV-l. Vi sai không thể này có thể mở rộng thêm khi chúng ta xét thêm biến đổi SubCells, MixColumns của vòng ba và trạng thái hoạt động của ΔY có dạng không đổi. Vì vậy, vi sai $\Delta X \to \Delta Y$ cùng là vi sai không thể đối với 2 vòng của mã pháp MKV-l.



Hình 9 - Một dạng vi sai không thể của MKV-256

Tương tự, đối với mã khối MKV-128, nhóm đề tài xây dựng một vi sai không thể qua 2 vòng $\Delta X \to \Delta Y$ trong đó sai khác đầu vào $\Delta X = \left(\Delta x_j^i\right) \in V_8^{16}$ chỉ có byte hoạt động (không mất tổng quát, giả sử là $\Delta x_0^0 \neq 0$) còn lại các byte sai khác bằng không, tức là $\Delta x_j^i = 0, (i,j) \neq (0,0)$; và sai khác đầu ra $\Delta Y = \left(\Delta y_j^i\right) \in V_8^{16}$ thỏa mãn trạng thái con $\Delta Y^0 \neq 0$ và $\Delta Y^1 = \Delta Y^2 = \Delta Y^3 = 0$, như sau:



Hình 10 - Một dạng vi sai không thể của MKV-128

Tuy nhiên, để mở rộng vi sai này cho số vòng lớn hơn về phần trước và phần sau, nhóm đề tài chưa tìm thấy. Hiện tại, nhóm đề tài vẫn chưa phát hiện ra được điểm yếu đối với MKV-l có số vòng lớn hơn 3 để thám mã vi sai không thể khai thác.

Tấn công tương quan không. Thám mã tuyến tính tương quan không [80-82] là bản sao của thám mã vi sai không thể trong trường hợp thám mã tuyến tính. Hiện tại, nhóm đề tài chưa tìm được điểm yếu của mã pháp MKV-l để một số thám mã đã biết khác khai thác.

3.4.6. Độ an toàn kháng lại thám mã vi sai khóa quan hệ

Đối với thám mã này, chúng ta sẽ xem sát mối quan hệ giữa các sai khác của các khóa vòng qua các vòng. Với hai vòng lặp hàm F_l, mã pháp MKV-l/k đảm bảo số lượng các S-hộp tích cực vi sai hoạt động tối thiểu là $4 \times \mathrm{Br}_d(M)$ trong mỗi nửa khối trạng thái khóa khi được kích hoạt. Khi đó, với lược đồ V, thì qua bốn vòng thì ít nhất sẽ có 05 nửa khối này được kích hoạt. Từ đó, chúng ta sẽ nhận được số lượng tối thiểu các S-hộp qua bốn vòng của lược đồ khóa. Cụ thể,

Xét sai khác khóa khác không đầu vào $\Delta\mathcal{K} \in V_{2l}$, sai khác này là đầu vào của vòng 1. Chúng ta kí hiệu $\Delta\mathcal{K}_i = \left(\Delta\mathcal{K}_i^{\mathrm{Left}}, \Delta\mathcal{K}_i^{\mathrm{Right}}\right) \in V_{2l}$ là sai khai đầu ra của vòng

thứ i và cũng là sai khác đầu vào của vòng thứ $i+1$ $(0 < i < R)$. Khi đó, ta có sự lan truyền các sai khác diễn ra như sau:

$$\left(\Delta\mathcal{K}_0^{\text{Left}}, \Delta\mathcal{K}_0^{\text{Right}}\right) \xrightarrow{\text{Vòng 1}} \left(\Delta\mathcal{K}_1^{\text{Left}}, \Delta\mathcal{K}_1^{\text{Right}}\right) \xrightarrow{\text{Vòng 2}} \left(\Delta\mathcal{K}_2^{\text{Left}}, \Delta\mathcal{K}_2^{\text{Right}}\right)$$
$$\xrightarrow{\text{Vòng 3}} \left(\Delta\mathcal{K}_3^{\text{Left}}, \Delta\mathcal{K}_3^{\text{Right}}\right) \xrightarrow{\text{Vòng 4}} \left(\Delta\mathcal{K}_4^{\text{Left}}, \Delta\mathcal{K}_4^{\text{Right}}\right)$$

Ta thấy rằng với bất kỳ sai khác khác không đầu vào $\text{wt}_l\left(\Delta\mathcal{K}_i^{\text{Left}}\right) = \text{wt}_l\left(\Delta\mathcal{K}_i^{\text{Right}}\right) \neq 0$, hàm F_l^{KS} sẽ được kích hoạt. Dẫn đến, số lượng S-hộp hoạt động vi sai trong nửa trạng thái khóa này sẽ đạt được theo Hệ quả 1, Hệ quả 2. Theo dõi vệt vi sai qua lược đồ V ta dễ đàng thấy được sẽ ít nhất 05 khối trong tổng số 08 khối được kích hoạt. Thật vậy, qua việc khảo sát các trọng số bó theo kích cỡ l đối với từng nửa khối trạng thái như sau:

STT	$wt_l(\Delta\mathcal{K})$	$wt_l(\Delta\mathcal{K}_1)$	$wt_l(\Delta\mathcal{K}_2)$	$wt_l(\Delta\mathcal{K}_3)$	Số lượng tối thiểu các nửa trạng thái được kích hoạt
1.	11	10	01	11	6
2.	01	11	10	01	5
3.	10	01	11	10	5

Khi đó, qua 4 vòng số lượng các S-hộp hoạt động sẽ được kích hoạt sẽ tối thiểu là $5 \times 4 \times \text{Br}_d(M)$ trong lược đồ khóa. Do đó, xác suất để nhận được sai khác khoá mong muốn sẽ không vượt quá $(\text{DP}_{\max})^{5\times4\times Br_d(M)}$ khi qua bốn vòng. Cụ thể, đối với MKV-128, giá trị này cỡ 2^{-540}, còn đối với MKV-256, giá trị này cỡ là 2^{-972}. Hơn nữa, cận trên của xác suất của sai khác giữa hai khóa con $(\mathcal{K}_i^0, \mathcal{K}_i^1)$ trong một khóa vòng $\mathcal{K}_i$ là $(\text{DP}_{\max})^{4\times Br_d(M)}$. Các kết quả này đảm bảo độ an toàn của MKV trước thám mã khóa quan hệ qua bốn vòng.

3.4.7. Thám mã khác

MKV cũng đã được xem xét độ an toàn của trước một số thám mã khác như thám mã vi sai-tuyến tính, thám mã không gian con bất biến, … Đặc biệt, đối với thám mã không gian con bất biến, do cấu trúc FLC, các không gian bất biến trên không gian V_l của MKV-l sẽ được quy dẫn về các không gian bất biến trên không gian V_w trong mỗi trạng thái con. Tuy nhiên, trong mỗi trạng thái con đều sử dụng các tầng tuyến tính dựa trên ma trận MDS có kích thước toàn bộ khối, dẫn đến không tồn tại một không gian con bất biến thực sự để khai thác trong thám mã. Hiện chưa tìm được điểm yếu của mã pháp để các thám mã này khai thác.

3.5. Thảo luận độ an toàn của MKV trong thời điểm chuyển tiếp lượng tử

Hiện nay, xuất hiện những thuật toán lượng tử có thể cải thiện hiệu suất của một số kỹ thuật thám mã chống lại các mã khối như thuật toán Simon, Grover, ... Trong phần này, chúng ta sẽ thảo luận độ an toàn của các phiên bản MKV trước một số thám mã khai thác lượng tử đã biết. Đầu tiên, chúng ta sẽ nhắc lại ba thiết lập thông

dụng cho mô hình an toàn của mã khối bao gồm: An toàn cổ điển (kí hiệu QS0), an toàn hậu lượng tử (kí hiệu QS1), an toàn lượng tử (kí hiệu QS2). Hiện nay, hầu hết thám mã lượng tử đều sử dụng hai thuật toán lượng tử quan trọng là thuật toán Grover và thuật toán Simon. Thuật toán Grover có tốc độ tăng tốc bậc hai so với thuật toán tìm kiếm tuyến tính cổ điển . Do đó, độ dài của khóa K phải được tăng gấp đôi để đảm bảo mức độ bảo mật tương tự như máy tính cổ điển nhưng chống lại máy tính lượng tử. Tuy nhiên, nếu các phép toán qubit là lớn và chậm thì việc triển khai thuật toán của Grover sẽ vô ích trên một máy tính lượng tử có thể thay đổi theo tỷ lệ do sự gia tăng chi phí lượng tử. Nếu các phép toán qubit nhỏ và nhanh, thì chỉ khi đó thuật toán Grover có thể là mối đe dọa đối với nhiều hệ thống mật mã [83]. Thay vì chỉ sử dụng tìm kiếm Grover, nhiều kỹ thuật thám mã cổ điển đã được lượng tử hóa [83, 85, 86, 110] để khôi phục một khóa một phần và sau đó chạy thuật toán Grover để tìm các bit còn lại của khóa. Các kỹ thuật thám mã lượng tử như vậy chỉ được coi là thành công khi độ phức tạp về thời gian theo yêu cầu của chúng nhỏ hơn độ phức tạp về thời gian của tìm kiếm khóa vét cạn dựa trên thuật toán của Grover. Trong [83, 85, 86, 110], một số kỹ thuật thám mã cổ điển đã được lượng tử hóa như kỹ thuật thám mã vi sai, thám mã vi sai rút gọn, thám mã vi sai không thể và thám mã tuyến tính đã được phân tích. Ngoài ra, thuật toán lượng tử Simon cũng được sử dụng để đánh giá độ an toàn của một số cấu trúc mã khối, trong một số thám mã dựa trên cấu trúc và kết hợp thuật toán Grover trong một số thám mã. Cuối cùng, một lưu ý quan trọng cần nhắc lại là độ phức tạp của tấn công vét cạn khoá trong thiết lập lượng tử sẽ giảm căn bậc hai so với tấn công này trong thiết lập cổ điển, nhờ thuật toán Grover. Do đó, chúng ta cần lấy độ phức tạp này làm cơ sở để đánh giá hiệu quả của các thám mã lượng tử được xem xét, cũng như sử dụng chúng làm chuẩn ngưỡng an toàn của mã khối trong các mô hình thiết lập lượng tử.

Đối với thám mã vi sai lượng tử. Trong [83], Kaplan và cộng sự đã phát triển một phiên bản lượng tử của bộ phân biệt vi sai và tấn công vòng cuối sử dụng thuật toán Grover, với giả định rằng việc sử dụng thuật toán Ambainis cùng với tìm kiếm Grover có thể làm giảm thêm độ phức tạp thời gian của tấn công. Áp dụng vào MKV, dựa trên các kết quả đánh giá chi tiết của cận trên xác suất của các vệt vi sai qua các vòng trong Bảng 5, khi đó với ba vòng thì mã khối có độ phức tạp của tấn công vi sai lượng tử sẽ lớn hơn với tấn công vét cạn khoá khi sử dụng thuật toán lượng tử cỡ $O(2^{k/2})$, với k là kích thước khóa (tương ứng với độ an toàn $k/2$ qubit) trong cả thiết lập QS1, QS2. Để xem xét độ an toàn với phiên bản khoá có độ dài lớn hơn, chúng ta xem xét tấn công vòng cuối vi sai lượng tử dựa trên tìm kiếm Grover, theo cách tiếp cận trong [83]. Với các kết quả cho cận trên xác suất của tất cả các vệt vi sai qua ba vòng không lớn hơn 2^{-k}, dẫn đến độ phức tạp của bước này lớn hơn so với tìm kiếm vét cạn khoá dựa trên Grover. Do đó, với các vòng bổ sung cho phiên bản độ dài khoá thì MKV đảm bảo độ an toàn lượng tử mong muốn trong QS1, tuy nhiên trong QS2 các phiên bản khoá có độ dài khoá sẽ chỉ đạt được độ an toàn theo

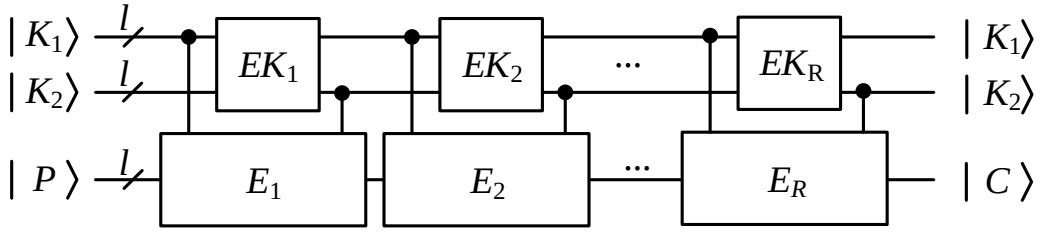
kích cỡ khối. Ngoài ra, MKV được xem xét thêm một số cải tiến của tấn công vi sai lượng tử như *tấn công vòng cuối dựa trên đếm lượng tử* [110], hay *tìm cặp vi sai lượng tử dùng thuật toán Bernstein–Vazirani* [86] với thiết lập QS1, QS2, với giả thiết kẻ tấn công có quyền truy cập vào mạch lượng tử của mã pháp. Độ phức tạp của các tấn công này đều lớn hơn mức an toàn lượng tử mong muốn.

Đối với thám mã tuyến tính lượng tử. Kaplan và cộng sự trong [83] cũng đã trình bày phiên bản lượng tử của bộ phân biệt tuyến tính và tấn công vòng cuối. Áp dụng vào MKV, dựa trên các kết quả đánh giá chi tiết của cận trên độ lệch các vệt vi sai qua các vòng, tương tự như với thám mã vi sai lượng tử, khi đó với bốn vòng thì mã khối có độ phức tạp lượng tử đang xem xét sẽ lớn hơn với tấn công vét cạn khoá khi sử dụng thuật toán lượng tử cỡ $O(2^{k/2})$ (tương ứng với độ an toàn $k/2$ qubit) trong cả thiết lập QS1, QS2. Hơn nữa, trong tấn công vòng cuối tuyến tính thiết lập QS1 và QS2, độ phức tạp dữ liệu và thời gian của MKV với độ dài khoá lớn đều đảm bảo cho độ phức tạp của tìm khoá tại các vòng cuối đều lớn hơn với độ phức tạp của thuật toán vét cạn dựa trên Grover.

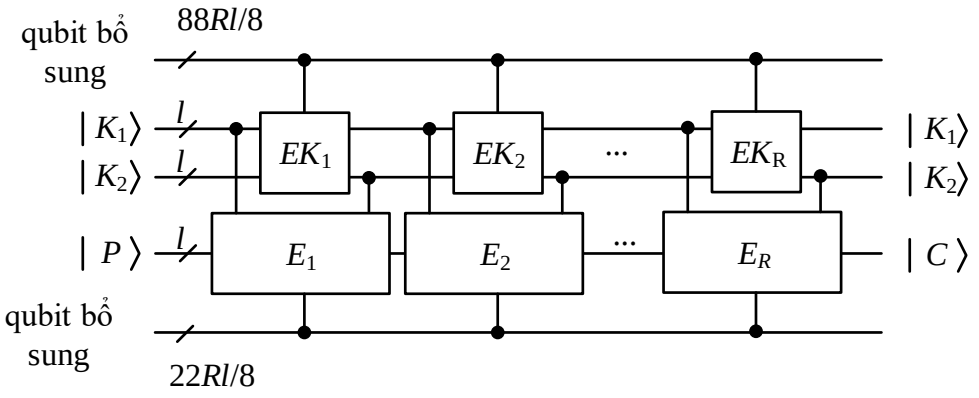
Một số thám mã lượng tử khác. Ngoài hai thám mã quan trọng ở trên, nhóm thiết kế cũng đã xem xét độ an toàn của MKV đối với một số tiếp cận của thám mã khác như tấn công vi sai cắt ngắn lượng tử (dựa trên thuật toán Ambainis [83] và dựa trên trên thuật toán Bernstein–Vazirani [87]); tấn công vi sai không thể lượng tử [86, 87] (cách tiếp cận đều sử dụng thuật toán lượng tử trong giai đoạn đầu để tìm vi phân không thể); hay kỹ thuật toán gặp ở giữa Demiric - Selçuk [88]. Các tấn công này đều yêu cầu độ phức tạp lớn hơn độ phức tạp khi tìm kiếm vét cạn dựa trên Grover. Ngoài ra, với các cách tiếp cận lượng tử hiện có cũng chưa phát hiện được điểm yếu trong MKV để các mã pháp này khai thác và cải tiến để thực hiện hiệu quả hơn so với tấn công vét cạn dựa trên Grover với số vòng đầy đủ.

Ước lượng tài nguyên lượng tử cho MKV$-l/k$.

Dựa vào các biến đổi ở trên, MKV được minh họa mạch lượng tử cho thực thi phép mã hóa một khối *l* bit của MKV-*l* như trong Hình 11 (không sử dụng qubit bổ sung) và Hình 12 (có sử dụng qubit bổ sung).



Hình 11 - Minh họa mã hóa một khối *l* bit không sử dụng qubit bổ sung



Hình 12 - Minh họa mã hóa một khối *l* bit có sử dụng qubit bổ sung

Để ước lượng tài nguyên lượng tử cần thiết, các thực nghiệm sử dụng trình mô phỏng VisuaStudio Code với thư viện mã nguồn mở ProjectQ [89]. Số lượng tài nguyên lượng tử cần thiết được cho trong Bảng 11 và Bảng 12 dưới đây. Trong mỗi ước lượng số qubit đầu vào là $3l$ gồm l qubit cho kích thước khối và $2l$ qubit cho 02 khóa con của 1 vòng.

Bảng 11 - Tài nguyên lượng tử cho các phiên bản MKV không sử dụng qubit bổ sung

Phiên bản mã pháp	Số Qbit	CNOT	8-Toffoli	Cổng X	Cổng Swap
MKV-128/128	384	916.960	229.440	1.019.779	768
MKV-128/192	384	1.069.680	267.680	1.189.660	896
MKV-128/256	384	1.222.400	305.920	1.359.543	1.024
MKV-256/256	768	1.855.760	458.880	2.039.427	1.536
MKV-256/384	768	2.164.840	535.360	2.379.164	1.792
MKV-256/512	768	2.473.920	611.840	2.718.903	2.048

Bảng 12 - Tài nguyên lượng tử cho các phiên bản MKV sử dụng qubit bổ sung

Phiên bản mã pháp	Số Qbit	CNOT	Toffoli	Cổng X	Cổng Swap	Full depth
MKV-128/128	10.944	77.920	83.520	16.579	768	3.265
MKV-128/192	12.704	90.800	97.440	19.240	896	3.805
MKV-128/256	14.464	103.680	111.360	21.902	1.024	4.346
MKV-256/256	21.888	177.680	167.040	33.027	1.536	3.612
MKV-256/384	25.408	207.080	194.880	38.344	1.792	4.210
MKV-256/512	28.928	236480	222.720	43.662	2.048	4.809

Khác với kết quả trong Bảng 11, Bảng 12 thống kê giá trị tham số full depth bởi trong cài đặt này sử dụng các cổng lượng tử cơ sở.

Như chúng ta đã biết, NIST đưa ra độ an toàn đối với thuật toán mã khối trước tấn công sử dụng thuật toán Grover [90]. Kết quả này dựa trên công trình nghiên cứu năm 2016 của nhóm Grassl và cộng sự [91]. Đến năm 2022, yêu cầu này của NIST được cập nhật [92] (xem bảng 12).

Bảng 13 - Độ phức tạp lượng tử theo yêu cầu của NIST với mã khối

Mức an toàn	NIST	
	Năm 2016 [90]	Năm 2022 [92]
1 – AES-128	2^{170}	2^{157}
3 – AES-192	2^{233}	2^{221}
5 – AES-256	2^{298}	2^{285}

Bảng 14 thống kê kết quả đánh giá độ phức tạp của MKV trước tấn công sử dụng thuật toán Grover và so sánh với một kết quả trên thế giới.

Bảng 14 - Độ phức tạp của MKV trước tấn công vét cạn khóa sử dụng thuật toán Grover và so sánh

Thuật toán	Số cặp bản rõ mã r	Số qubit (M)	Tổng số cổng (G)	Full depth (FD)	Độ phức tạp FD-G $(FD \times G)$	Độ phức tạp FD-M $(FD \times M)$
AES128 [93]	1	3.257	$1{,}17 \cdot 2^{82}$	$1{,}27 \cdot 2^{74}$	$1{,}49 \cdot 2^{156}$	$1{,}01 \cdot 2^{86}$
AES192 [93]	2	7.161	$1{,}29 \cdot 2^{115}$	$1{,}64 \cdot 2^{106}$	$1{,}03 \cdot 2^{222}$	$1{,}43 \cdot 2^{119}$
AES256 [93]	2	7.537	$1{,}84 \cdot 2^{147}$	$1{,}16 \cdot 2^{139}$	$1{,}07 \cdot 2^{287}$	$1{,}94 \cdot 2^{151}$
MKV-128/128	1	10.944	$1{,}40 \cdot 2^{80}$	$1{,}25 \cdot 2^{76}$	$1{,}75 \cdot 2^{156}$	$1{,}66 \cdot 2^{89}$
MKV-128/192	2	12.704	$1{,}65 \cdot 2^{112}$	$1{,}46 \cdot 2^{108}$	$1{,}21 \cdot 2^{223}$	$1{,}75 \cdot 2^{123}$
MKV-128/256	2	14.464	$1{,}86 \cdot 2^{145}$	$1{,}67 \cdot 2^{141}$	$1{,}55 \cdot 2^{287}$	$1{,}82 \cdot 2^{155}$
MKV-256/256	1	21.888	$1{,}44 \cdot 2^{145}$	$1{,}39 \cdot 2^{140}$	$1{,}00 \cdot 2^{287}$	$1{,}02 \cdot 2^{154}$
MKV-256/384	2	25.408	$1{,}67 \cdot 2^{210}$	$1{,}61 \cdot 2^{205}$	$1{,}39 \cdot 2^{416}$	$1{,}33 \cdot 2^{220}$
MKV-256/512	2	28.928	$1{,}32 \cdot 2^{276}$	$1{,}84 \cdot 2^{269}$	$1{,}21 \cdot 2^{546}$	$1{,}34 \cdot 2^{284}$

Như vậy, tài nguyên lượng tử để cài đặt MKV vẫn còn khá lớn. Nhóm thiết kế cũng đang trong quá trình tối ưu hoá cài đặt nhằm giảm thiểu số lượng qubit và số cổng CNOT cần dùng, song chưa có một bước tiến đáng chú ý nào mà tăng hiệu quả đáng kể trong quá trình lượng tử hoá mã pháp để sử dụng thuật toán Grover gây ảnh hưởng đến độ an toàn của MKV.

Tóm lại, MKV có phiên bản kích cỡ khối là 256-bit với mong muốn đạt được độ an toàn lượng tử trong QS2 theo đúng khuyến cáo của Liên minh Châu Âu(xem

báo cáo thứ nhất về các mật mã kháng lượng tử mới[20], báo cáo thứ 2 về các mật mã kháng lượng tử mới[21]), trong bảng 15.

Bảng 15 – Ngưỡng độ an toàn kỳ vọng của mã khối trong các thiết lập lượng tử

Thiết lập	$l = 128$			$l = 256$		
	$k = 128$	$k = 192$	$k = 256$	$k = 256$	$k = 384$	$k = 512$
QS0	128	192	256	256	384	512
QS1	64	96	128	128	192	256
QS2	64	64	64	128	128	128

Hơn nữa, việc tăng kích thước khối cũng nâng cao mức dữ liệu mã hoá với một khoá bí mật được sử dụng, khi chúng ta sử dụng mã khối trong một chế độ hoạt động an toàn. Đây cũng là yêu cầu quan trọng của một số nhà phát triển trong phần bình luận công khai về FIPS 197-AES (xem [64]).

3.6. Đánh giá ngẫu nhiên đầu ra của MKV

Một nguyên thuỷ mật mã nói chung và mã khối nói riêng cần phải đạt tính ngẫu nhiên đầu ra, tức là các tính chất không ngẫu nhiên tại đầu vào phải không tiếp tục được thể hiện tại đầu ra. Dựa trên ý tưởng này, các tác giả trong [94] đã đưa ra một phương pháp đánh giá tính ngẫu nhiên cho các mã khối và hàm băm. Ý tưởng chính là xây dựng các tập dữ liệu đầu vào không ngẫu nhiên (có độ dư thừa) và tính toán các dữ liệu đầu ra tương ứng sử dụng các nguyên thuỷ mật mã cần đánh giá tính ngẫu nhiên. Sau đó, kiểm tra tính ngẫu nhiên của các tập dữ liệu đầu ra bởi các kiểm tra thống kê. Nếu dữ liệu là ngẫu nhiên thì rõ ràng tính ngẫu nhiên xuất phát từ nguyên thuỷ mật mã, nếu không thì nguyên thuỷ mật mã đó không có tính ngẫu nhiên. Mục tiêu của chúng ta là đánh giá tính ngẫu nhiên đầu ra của mã khối đề xuất. Trong các kiểm tra, chúng ta giả sử rằng đối tượng cần kiểm tra là một ánh xạ $\mathbb{F}_2^m \times \mathbb{F}_2^n \to \mathbb{F}_2^n$, trong đó m là kích cỡ khoá và n là kích cỡ khối của mã khối đề xuất.

3.6.1. Quy trình đánh giá tính ngẫu nhiên đầu ra cho mã khối

Để đánh giá tính ngẫu nhiên đầu ra cho MKV, quy trình kiểm tra được đề xuất như sau:

Bước 1: Sinh một tập dữ liệu không ngẫu nhiên làm đầu vào cho mã khối như được mô tả trong phần sau.. Ở đây, 4 loại tập dữ liệu đầu vào được xem xét: tập dữ liệu có mật độ thấp (tức là có trọng số thấp), tập dữ liệu có mật độ cao, tập dữ liệu sai khác 1 bit, tập dữ liệu dịch vòng.

Bước 2: Đối với mỗi phiên bản rút gọn vòng của mã khối, tính toán tập dữ liệu đầu ra tương ứng với tập dữ liệu đầu vào được tạo trong bước 1 và khoá tuỳ ý (có thể xem xét trường hợp tồi nhất đó là khoá không ngẫu nhiên). Ví dụ, nếu tập dữ liệu đầu vào là

[20] https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5be0027a4&appId=PPGMS

[21] https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5cabf88a3&appId=PPGMS

LW256 thì ta cần tính 8 tập dữ liệu đầu ra tương ứng với các phiên bản mã khối MKV-256/512 có số vòng từ 1 đến 8. Mỗi tập dữ liệu đầu ra bao gồm 2796416 chuỗi đầu ra tương ứng của 2796416 chuỗi đầu vào 256-bit trong tập dữ liệu đầu vào LW256.

Bước 3: Đối với mỗi tập dữ liệu đầu ra, sử dụng các kiểm tra thống kê cho đoạn ngắn được mô tả dưới đây tính toán các giá trị p-value tương ứng cho từng chuỗi trong tập dữ liệu đầu ra.

Bước 4: Đối với mỗi kiểm tra thống kê, ta có một tập các giá trị p-value tương ứng với tập dữ liệu đầu ra. Khi đó, áp dụng kiểm tra so khớp (good of fitness) để kiểm tra xem các *p-value* có phân bố khớp với lý thuyết trên đoạn [0, 1] hay không, bằng cách chia đoạn [0, 1] thành 10 khoảng con [0.0, 0.1], (0.1, 0.2], …, (0.9, 1.0]. Gọi *m* là số lượng các dãy kiểm tra, và F_i là số lượng các dãy có giá trị *p-value* nằm trong khoảng thứ *i* với $i = 1, \cdots, 10$. Khi đó, giá trị X dưới đây tuân theo phân phối χ^2 với 9 bậc tự do và giá trị ***p-value*** mức 2 được tính như sau:

$$X = \sum_{i=1}^{10} \frac{(F_i - m \cdot p_i)^2}{m \cdot p_i}$$

và $\boldsymbol{p - value} = igamc\left(\frac{9}{2}, \frac{\chi^2}{2}\right)$ trong đó *igamc* là hàm gamma không đầy đủ

$$igamc(a,x) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt,$$

với hàm gamma $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$. Các giá trị $p_i, 1 \le i \le 10$ đã được xác định chi tiết trong phần sau.

Nếu $\boldsymbol{p - value} \ge 0{,}0001$ thì kết quả kiểm tra được coi là **vượt qua**, tức là phân phối của các giá trị p-value không khác xa so với các giá trị lý thuyết.

Bước 5: Tổng hợp các kết quả và đưa ra kết luận. Một phiên bản mã khối với số vòng r được coi là đạt tính ngẫu nhiên nếu tất cả các tập dữ liệu đầu ra tương ứng **vượt qua** đối với tất cả các kiểm tra thống kê được đề xuất sử dụng.

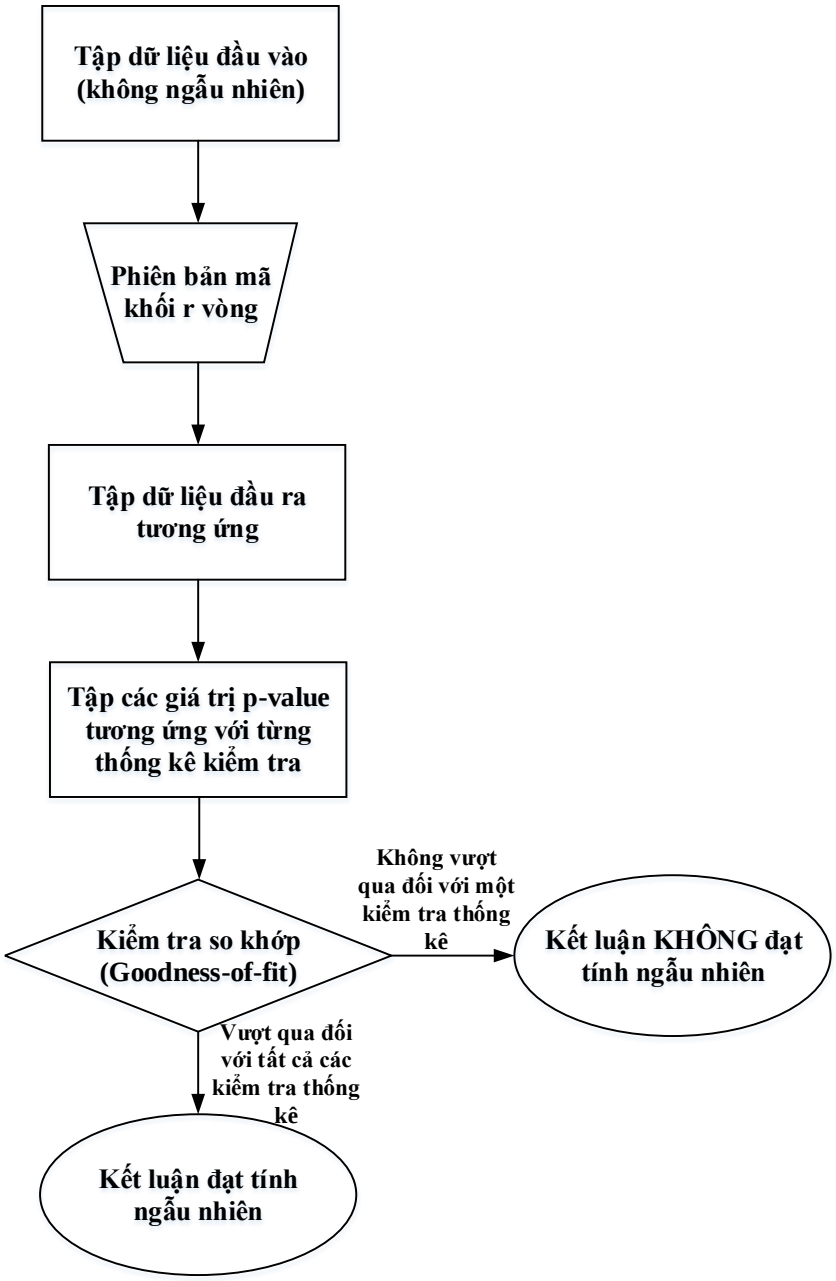
Chú ý: Số vòng R ở đây tương ứng là số vòng của mã khối.

Lý do ở đây chọn giá trị $\alpha = 0{,}0001$ như sau:

- Thứ nhất, nếu ta giả sử các kiểm tra thống kê là độc lập, khi đó xác suất sai lầm loại I (tức là xác suất để dữ liệu ngẫu nhiên được kết luận là không ngẫu nhiên) là $1 - (1 - \alpha)^5$ trong đó α là mức ý nghĩa cho mỗi kiểm tra và *s* là số lượng các kiểm tra thống kê được sử dụng. Khi ta chọn $\alpha = 0{,}0001$ thì 0,0799% dữ liệu ngẫu nhiên sẽ được kết luận là không ngẫu nhiên. Xác suất này là rất nhỏ. Trong thực tế, xác suất sai lầm loại II (tức là xác suất để dữ liệu không ngẫu nhiên được kết luận là ngẫu nhiên) cần được quan tâm hơn tuy nhiên việc xác

định xác suất này khá khó vì có vô số trường hợp dữ liệu được coi là không ngẫu nhiên.

- Thứ hai, ở đây ta đã sử dụng kiểm tra so khớp để đánh giá việc phân bố của các giá trị p-value trên khoảng [0, 1] có khớp với lý thuyết không. Và giá trị p-value mức 2 tính được chỉ rất nhỏ gần với 0 khi phân bố này khác xa so với phân bố lý thuyết, còn ngược lại thì có thể coi là chấp nhận được.



Hình 13 - Sơ đồ quy trình đánh giá tính ngẫu nhiên cho mã khối

3.6.2. Các tập bản rõ đầu vào không ngẫu nhiên được sử dụng

Các loại đặc biệt được sử dụng trong quy trình đánh giá như sau:

1. *Tập bản rõ có mật độ thấp:* Tập dữ liệu bản rõ có mật độ thấp (LW) được tạo thành bởi các chuỗi nhị phân có trọng số thấp. Độ dài bản rõ tương ứng với các thuật toán mã khối đề xuất với các phiên bản kích cỡ khối khác nhau. Trong trường hợp 128 bit, tập dữ liệu bao gồm các chuỗi nhị phân 128-bit mà trọng số không vượt quá 3. Tương tự như vậy đối với tập dữ liệu gồm các chuỗi độ dài 256 bit. Cụ thể, số lượng các bản rõ có độ dài *m* bit và mỗi bản rõ có trọng số Hamming nhỏ hơn hoặc bằng *k*, ký hiệu là N_n^k được tính như sau $N_n^k = \sum_{i=1}^{k} \binom{n}{i}$. Khi đó, số lượng bản rõ theo các độ dài khác nhau được đưa ra trong Bảng 16.

Bảng 16 - Số lượng các chuỗi cho các độ dài bản rõ khác nhau.

m (bit)	Trọng số	số lượng chuỗi
128	≤ 3	349632
256	≤ 3	2796416

2. *Tập bản rõ có mật độ cao:* Tập dữ liệu bản rõ mật độ cao (HW) được tạo thành bằng cách chọn các bản rõ (hoặc bản rõ) mật độ cao và chúng được tạo thành tương tự như trường hợp bản rõ mật độ thấp. Nói cách khác, các đầu vào mật độ cao là phần bù theo bit của các đầu vào mật độ thấp.
3. *Tập bản rõ thác 1-bit:* Để tạo thành tập dữ liệu thác bản rõ 1 bit, đầu tiên chọn ngẫu nhiên một bản rõ P có độ dài m. Sau đó, lật từng bit của P ta thu được một tập *m* bản rõ và tương ứng với đó là các bản mã. Quá trình tương tự được áp dụng cho *k* bản rõ khác nhau để thu được một tập mk chuỗi. Các giá trị của k cho các độ dài đầu vào khác nhau được đưa ra trong Bảng 17.

Bảng 17 - Lựa chọn giá trị *k* cho thác bản rõ 1-bit và bản rõ dịch vòng.

m (bit)	*k* (bản rõ ngẫu nhiên)	*mk* (dãy)
128	8192	1048576
256	4096	1048576

4. *Tập bản rõ dịch vòng:* Một bản rõ ngẫu nhiên P có độ dài *m* được chọn để tạo thành tập dữ liệu dịch vòng bản rõ (Rot), và một tập *m* bản rõ được tạo thành bằng cách dịch vòng liên tiếp 1-bit của P và tính các bản mã tương ứng. Quá trình tương tự được áp dụng cho k bản rõ khác nhau để thu được một tập mk chuỗi.

3.6.3. Các thống kế được sử dụng.

Để đánh giá tính ngẫu nhiên đầu ra cho mã khối đề xuất có độ dài đầu ra là 128 và 256. Chúng tôi đề xuất sử dụng các thống kê kiểm tra phù hợp cho độ dài 128 và 256 đã được đề xuất trong [95] bước 3 của quy trình đánh giá trong phần trên. Đặc biệt, đối với phân bố p-value của các kiểm tra này cho các dãy có độ dài ngắn đã được nghiên cứu và chỉ ra trong [96]. Cụ thể, sáu thống kê được sử dụng:

- ***Thống kê tần số.*** Thống kê tần số xem xét tần số các số 1 (hoặc 0) trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.

- ***Thống kê loạt.*** Thống kê loạt xem xét tổng số loạt trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.
- ***Thống kê loạt các số 1 dài nhất trong một khối.*** Thống kê loạt các số 1 dài nhất trong một khối xem xét loạt các số 1 dài nhất trong một khối trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.
- ***Thống kê Serial.*** Thống kê Serial xem xét tần số các bít đơn và tần số các bộ đôi trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.
- ***Thống kê entropy xấp xỉ.*** Thống kê entropy xấp xỉ xem xét tần số các khối m bit và $m + 1$ bit trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.
- ***Thống kê tổng tích luỹ.*** Thống kê tổng tích luỹ xem xét giá trị tổng tích luỹ có giá trị tuyệt đối lớn nhất trong một dãy nhị phân và so sánh với giá trị lý thuyết của một dãy ngẫu nhiên có cùng độ dài.

3.6.4. Kết quả đánh giá cho mã khối đề xuất

Các kết quả đánh giá đối với các kích cỡ khối 128-bit và 256-bit cho từng phiên bản khóa có độ dài khác nhau của MKV. Khoá được sử dụng trong các đánh giá sau là khoá toàn bit 0. Khi đó ta có kết quả cụ thể như sau:

Bảng 18 - Các giá trị p-value theo vòng cho MKV đối các tập dữ liệu đầu vào khác nhau.

Số vòng	KT tần số	KT Loạt	KT loạt 1 dài nhất trong khối 8 bit	KT Serial	KT entropy xấp xỉ	KT tổng tích luỹ
MKV-256/512						
Đối với tập dữ liệu LW256						
1	0	0	0	0	0	0
2	**0.637588**	**0.885883**	**0.627701**	**0.575822**	**0.611496**	**0.739528**
3	0.853804	0.021540	0.570203	0.197694	0.124728	0.689088
4	0.517818	0.632038	0.018662	0.757717	0.855863	0.542462
5	0.801323	0.126381	0.750872	0.085257	0.020516	0.593425
6	0.001538	0.260201	0.299307	0.078327	0.035701	0.486843
7	0.074360	0.847565	0.499330	0.100360	0.229766	0.013339
8	0.057885	0.701374	0.168971	0.071027	0.192889	0.517303
Đối với tập dữ liệu HW256						
1	0.039950	0	0	0	0	0
2	**0.231923**	**0.467467**	**0.069662**	**0.269381**	**0.176631**	**0.838731**
3	0.800637	0.409481	0.912805	0.265458	0.316531	0.178969
4	0.077584	0.819651	0.032601	0.628762	0.561010	0.051166
5	0.110754	0.987176	0.481075	0.984519	0.625497	0.490765
6	0.962606	0.605383	0.803213	0.954250	0.964876	0.553244
7	0.647937	0.502105	0.052866	0.808923	0.608734	0.816760
8	0.761734	0.311062	0.367059	0.309850	0.128289	0.478075
Đối với tập dữ liệu AV1_256						
1	**0.637456**	**0.001021**	**0.009130**	**0.032346**	**0.027770**	**0.379104**

2	0.476152	0.308137	0.754310	0.064378	0.181366	0.365511
3	0.333216	0.237989	0.066591	0.669162	0.694834	0.904723
4	0.524918	0.032895	0.119184	0.119027	0.197354	0.522938
5	0.540068	0.131351	0.984853	0.374780	0.642997	0.631082
6	0.088257	0.204429	0.912824	0.640490	0.747277	0.494743
7	0.352370	0.578437	0.015156	0.297785	0.507374	0.422715
8	0.366700	0.851687	0.023159	0.065043	0.500796	0.330422
Đối với tập dữ liệu Rot256						
1	**0.834471**	**0.462732**	**0.880623**	**0.851429**	**0.555777**	**0.352624**
2	0.449675	0.085826	0.101010	0.367657	0.454699	0.209828
3	0.585633	0.042689	0.166935	0.269872	0.179258	0.751224
4	0.302518	0.532760	0.160069	0.365266	0.404796	0.644725
5	0.471470	0.173831	0.618488	0.632051	0.604709	0.386322
6	0.202080	0.689249	0.363087	0.797760	0.729357	0.627502
7	0.389749	0.753258	0.270045	0.847001	0.676933	0.168381
8	0.387948	0.404771	0.630413	0.982166	0.895388	0.608606
MKV-256/384						
Đối với tập dữ liệu LW256						
1	0	0	0	0	0	0
2	**0.087622**	**0.267901**	**0.175411**	**0.084664**	**0.107863**	**0.722707**
3	0.267003	0.957192	0.432325	0.196702	0.151346	0.323633
4	0.841448	0.581082	0.915552	0.328237	0.686552	0.219350
5	0.326371	0.468235	0.535267	0.564386	0.537477	0.142288
6	0.070318	0.348022	0.582738	0.662676	0.781297	0.688489
7	0.687349	0.468216	0.955489	0.227318	0.292005	0.243668
Đối với tập dữ liệu HW256						
1	0.003658	0.004132	0.000000	0.349582	0.000171	0.000000
2	**0.480727**	**0.848614**	**0.021636**	**0.931175**	**0.886318**	**0.472914**
3	0.577770	0.770115	0.770893	0.624027	0.766782	0.986478
4	0.224772	0.033290	0.695426	0.415690	0.505864	0.710427
5	0.055812	0.760094	0.912853	0.586638	0.425567	0.856008
6	0.568540	0.669009	0.747493	0.528391	0.322376	0.123285
7	0.860372	0.647055	0.978559	0.415972	0.733038	0.498740
Đối với tập dữ liệu AV1_256						
1	**0.746692**	**0.616374**	**0.200161**	**0.459191**	**0.410764**	**0.829058**
2	0.531174	0.126637	0.254078	0.833544	0.527192	0.381156
3	0.262467	0.948017	0.594833	0.210888	0.225130	0.517176
4	0.604308	0.591107	0.705190	0.383022	0.356352	0.879238
5	0.080660	0.635456	0.327082	0.466575	0.679793	0.038434
6	0.328013	0.590589	0.797084	0.870610	0.639506	0.642574
7	0.646655	0.475839	0.462868	0.193457	0.283506	0.739817
Đối với tập dữ liệu Rot256						
1	**0.451253**	**0.142038**	**0.889937**	**0.082376**	**0.034144**	**0.537789**
2	0.673755	0.976528	0.652550	0.940901	0.870369	0.975220
3	0.517002	0.167780	0.070140	0.502211	0.409473	0.584819
4	0.170534	0.558521	0.202545	0.388297	0.830975	0.408688
5	0.678481	0.870681	0.607004	0.513010	0.698369	0.696009
6	0.891053	0.121805	0.153608	0.309580	0.306362	0.917063
7	0.403314	0.204981	0.776258	0.989393	0.946937	0.267970
MKV-256/256						
Đối với tập dữ liệu LW256						
1	0.000000	0.000000	0.000000	0.021295	0.031694	0.000000
2	**0.917989**	**0.770039**	**0.906822**	**0.295862**	**0.055443**	**0.957152**

3	0.802197	0.585258	0.337728	0.813780	0.435974	0.154304
4	0.267722	0.981134	0.883871	0.699523	0.853658	0.157893
5	0.284887	0.913052	0.929633	0.200720	0.309334	0.359216
6	0.531074	0.492328	0.417744	0.523436	0.667007	0.467473
Đối với tập dữ liệu HW256						
1	0.000003	0.000000	0.000000	0.000000	0.000000	0.000000
2	**0.295692**	**0.904653**	**0.529029**	**0.389228**	**0.182918**	**0.349827**
3	0.062563	0.749029	0.109213	0.348450	0.093514	0.472215
4	0.025939	0.036469	0.697312	0.120577	0.084348	0.013216
5	0.073243	0.611298	0.775373	0.326974	0.490927	0.089903
6	0.615532	0.627621	0.078165	0.330001	0.250354	0.268783
Đối với tập dữ liệu AV1_256						
1	**0.005814**	**0.473966**	**0.053974**	**0.557281**	**0.230276**	**0.092797**
2	0.077314	0.938447	0.891967	0.738264	0.772361	0.816095
3	0.074090	0.832986	0.709963	0.566301	0.552579	0.680966
4	0.770195	0.343445	0.431432	0.345569	0.308426	0.711050
5	0.364868	0.466354	0.860464	0.610297	0.378788	0.040360
6	0.069301	0.949091	0.554466	0.974981	0.697766	0.711866
Đối với tập dữ liệu Rot256						
1	**0.408827**	**0.999544**	**0.478012**	**0.973111**	**0.469662**	**0.893785**
2	0.088960	0.155755	0.474666	0.447841	0.670805	0.223805
3	0.724639	0.633233	0.137764	0.889883	0.963257	0.225497
4	0.922043	0.943392	0.243487	0.866594	0.497716	0.737674
5	0.874228	0.377650	0.031560	0.486818	0.611424	0.482822
6	0.314808	0.254190	0.160371	0.331664	0.094937	0.324718
MKV-128/256						
Đối với tập dữ liệu LW128						
1	0	0	0	0	0	0
2	**0.341961**	**0.881471**	**0.581879**	**0.077515**	**0.054115**	**0.505021**
3	0.925465	0.093544	0.343547	0.950143	0.941760	0.050184
4	0.112494	0.786904	0.963752	0.761791	0.772480	0.276378
5	0.194318	0.689087	0.929756	0.873957	0.715049	0.572462
6	0.368204	0.488986	0.309853	0.517183	0.612567	0.153969
7	0.590412	0.176323	0.021114	0.029127	0.041467	0.598207
8	0.131382	0.361427	0.134039	0.928302	0.916801	0.629374
Đối với tập dữ liệu HW128						
1	0	0	0	0	0	0
2	**0.138035**	**0.023602**	**0.395366**	**0.176674**	**0.345755**	**0.016023**
3	0.720015	0.063595	0.956407	0.389306	0.607223	0.162551
4	0.731306	0.974046	0.720939	0.220640	0.115533	0.193755
5	0.521474	0.132196	0.179004	0.143356	0.513462	0.981614
6	0.592476	0.657092	0.187838	0.652538	0.616866	0.083167
7	0.433865	0.127731	0.604174	0.701891	0.925096	0.028838
8	0.921906	0.102697	0.499860	0.268786	0.279704	0.781357
Đối với tập dữ liệu AV1_128						
1	0.073511	0.293401	0	0.005080	0.003404	0
2	**0.313099**	**0.938870**	**0.964403**	**0.641755**	**0.631739**	**0.702489**
3	0.545321	0.909961	0.292224	0.264543	0.540355	0.325202
4	0.491049	0.735247	0.259927	0.386525	0.557418	0.229110
5	0.328671	0.296077	0.578997	0.045419	0.147066	0.367722
6	0.549721	0.040831	0.516081	0.833018	0.834953	0.436863
7	0.741618	0.758267	0.358422	0.702100	0.560356	0.381170
8	0.990455	0.583392	0.835328	0.586689	0.173932	0.966947

Đối với tập dữ liệu Rot128						
1	**0.033851**	**0.693331**	**0.318519**	**0.502839**	**0.453034**	**0.156166**
2	0.580415	0.040563	0.646421	0.677559	0.634257	0.834595
3	0.619028	0.247480	0.851522	0.998375	0.996672	0.757900
4	0.935691	0.504534	0.830121	0.976586	0.977695	0.664942
5	0.923695	0.423804	0.386366	0.512928	0.509673	0.487186
6	0.074319	0.156521	0.684070	0.142935	0.139392	0.016960
7	0.552360	0.872659	0.058365	0.945651	0.891311	0.235148
8	0.058901	0.145024	0.143935	0.148078	0.578345	0.410879
MKV-128/192						
Đối với tập dữ liệu LW256						
1	0	0	0	0	0	0
2	0.526343	0.000000	0.004469	0.003806	0.001017	0.000253
3	**0.462663**	**0.460336**	**0.906088**	**0.557431**	**0.874558**	**0.948640**
4	0.839202	0.441223	0.921978	0.368198	0.507275	0.903389
5	0.038073	0.648640	0.542496	0.353858	0.390668	0.063937
6	0.671712	0.701105	0.287511	0.163175	0.203361	0.846811
7	0.973685	0.292102	0.416413	0.616053	0.751745	0.477069
Đối với tập dữ liệu HW128						
1	0	0	0	0	0	0
2	0	0.015377	0	0	0	0
3	**0.165231**	**0.010316**	**0.989828**	**0.288726**	**0.635352**	**0.321053**
4	0.400892	0.628483	0.462262	0.154799	0.650407	0.047096
5	0.015571	0.861240	0.236808	0.981332	0.909450	0.235465
6	0.392661	0.155664	0.810090	0.350313	0.639821	0.775621
7	0.316098	0.694250	0.437562	0.703643	0.829537	0.182986
Đối với tập dữ liệu AV1_128						
1	**0.022492**	**0.450493**	**0.494482**	**0.053023**	**0.088812**	**0.002839**
2	0.559146	0.039579	0.039579	0.330434	0.218111	0.634655
3	0.719668	0.292576	0.113822	0.066457	0.051794	0.069578
4	0.863535	0.183012	0.193444	0.962271	0.972365	0.744939
5	0.976379	0.874296	0.592558	0.088971	0.076201	0.964603
6	0.185128	0.026612	0.065699	0.904222	0.956369	0.972944
7	0.559572	0.428312	0.812335	0.666377	0.415805	0.583907
Đối với tập dữ liệu Rot128						
1	**0.592821**	**0.574972**	**0.230417**	**0.349560**	**0.103837**	**0.325428**
2	0.653708	0.006441	0.145272	0.243243	0.119210	0.218878
3	0.899381	0.530569	0.457770	0.218050	0.671767	0.415553
4	0.118215	0.640640	0.711723	0.271641	0.402039	0.464728
5	0.757944	0.968466	0.788442	0.153093	0.460542	0.484623
6	0.314321	0.662449	0.048064	0.282332	0.214065	0.568306
7	0.532484	0.180076	0.310545	0.738704	0.765641	0.290328
MKV-128/128						
Đối với tập dữ liệu LW128						
1	0	0	0	0	0	0
2	**0.395383**	**0.429381**	**0.380252**	**0.411558**	**0.660426**	**0.296496**
3	0.053401	0.031311	0.427009	0.583750	0.585849	0.964438
4	0.650671	0.084407	0.712145	0.794333	0.472194	0.626571
5	0.799927	0.478243	0.928732	0.521663	0.591224	0.903262
6	0.407673	0.000201	0.536549	0.018137	0.078004	0.373471
Đối với tập dữ liệu HW128						
1	0	0	0	0	0	0
2	**0.621576**	**0.595336**	**0.784399**	**0.825933**	**0.736407**	**0.614321**

3	0.962577	0.905248	0.233679	0.442117	0.502203	0.244754
4	0.148567	0.181076	0.644228	0.131079	0.085075	0.221038
5	0.360096	0.488155	0.599905	0.696378	0.929287	0.577494
6	0.812055	0.888544	0.020241	0.542003	0.693055	0.589136
Đối với tập dữ liệu AV1_256						
1	0	0	0	0.000486	0.002923	0
2	**0.061416**	**0.748789**	**0.539786**	**0.996998**	**0.979119**	**0.716080**
3	0.622102	0.323858	0.058580	0.712911	0.129759	0.212257
4	0.579680	0.844132	0.598973	0.163586	0.149704	0.427930
5	0.769563	0.822993	0.999567	0.805645	0.316422	0.994790
6	0.280076	0.604687	0.029004	0.353730	0.192992	0.688562
Đối với tập dữ liệu Rot128						
1	**0.376898**	**0.255349**	**0.502785**	**0.559883**	**0.324686**	**0.220848**
2	0.056503	0.959853	0.916716	0.470922	0.642988	0.031775
3	0.334003	0.194604	0.073649	0.062064	0.049836	0.473674
4	0.225563	0.250969	0.323160	0.379643	0.465918	0.159390
5	0.159778	0.186926	0.422523	0.993472	0.833115	0.293890
6	0.509752	0.793347	0.911990	0.800172	0.461518	0.096826

Như vậy, chúng ta có kết quả đánh giá qua các vòng được tổng hợp trong bảng sau:

Thuật toán	Số vòng đầy đủ	Số vòng đạt tính ngẫu nhiên	Các tập dữ liệu đầu vào
MKV-256/512	8	≥ 2	LW, HW, Av1, Rot
MKV-256/384	7	≥ 2	LW, HW, Av1, Rot
MKV-256/256	6	≥ 2	LW, HW, Av1, Rot
MKV-128/256	8	≥ 2	LW, HW, Av1, Rot
MKV-128/192	7	≥ 3	LW, HW, Av1, Rot
MKV-128/128	6	≥ 2	LW, HW, Av1, Rot

Kết quả thu được cho thấy các mã khối đề xuất đều đạt tính ngẫu nhiên đầu ra với ranh giới an toàn (số vòng đầy đủ) lớn.

3.7. Một số kết quả cài đặt thực thi MKV trên các nền tảng thông dụng

Trong phần này, một số kết quả thực thi thuật toán cả trên phần cứng và phần mềm được trình bày, chi tiết phương pháp và kỹ thuật cài đặt được sử dụng xem [7, 8].

3.7.1. Cài đặt phần mềm

MKV đã được cài đặt thực thi tất cả các phiên bản bằng ngôn ngữ C++ theo hai phương pháp sử dụng các bảng tính trước và không sử dụng bảng tính trước. Sau đây là một số kết quả thống kê về thực thi và so sánh với các mã khối thông dụng khác. Các kết quả chạy này đều được chạy trên máy có năng lực tính toán là CPU Intel® Xeon E3-1225 v5 với RAM 8G với hệ điều hành Windows 10 với trình biên dịch Visual Studio 2017 chế độ Release, các mã khối không hề dùng một lệnh Assembler và sử dụng chế độ ECB với các kết quả trong Bảng 19 và Bảng 20 theo hai phương pháp đã nêu.

Bảng 19 - Cài đặt phần mềm sử dụng các bảng tính trước

STT	Thuật toán	Kích thước bảng tra (mã hóa + giải mã) KBytes	Tốc độ mã hoá (Mb/s)	Tốc độ giải mã (Mb/s)	Nguồn cài đặt
Kích thước khối 128-bit					
1	MKV-128/128	8	**1529**	**1518**	Của nhóm thiết kế
	MKV-128/192	8	**1320**	**1299**	
	MKV-128/256	8	**1156**	**1151**	
2	AES-128/128	8	1783	1779	Gladman[22] (trên nền tảng 32-bit)
	AES-128/192	8	1526	1520	
	AES-128/256	8	1323	1315	
3	Kuznyechik	128	843	804	Oliynykov [23]
4	Kalyna-128/128	32	1749	1716	
	Kalyna-128/256	32	1331	1329	
Kích cỡ khối 256-bit					
1	MKV-256/256	16	**1599**	**1612**	Của nhóm thiết kế
	MKV-256/384	16	**1386**	**1390**	
	MKV-256/512	16	**1225**	**1226**	
2	Kalyna-256/256	16	1616	n/a	Oliynykov
	Kalyna-256/512	16	1299	n/a	

Bảng 20 - Cài đặt phần mềm không sử dụng các bảng tính trước

STT	Thuật toán	Tốc độ mã (Mb/s)	Tốc độ giải mã (Mb/s)
Kích thước khối 128-bit			
1	MKV-128/128	**245**	**244**
	MKV-128/192	**210**	**211**
	MKV-128/256	**184**	**185**
2	AES-128/128	270	204
	AES-128/192	225	221
	AES-128/256	192	187
Kích thước khối 256-bit			
1	MKV-256/256	186	189
	MKV-256/384	153	155
	MKV-256/512	142	142
2	Kalyna-256/256	144	26
	Kalyna-256/512	113	20

22 Trong trang web: http://brg.a2hosted.com//oldsite/cryptography_technology/rijndael/index.php
23 Trong trang web: https://github.com/Roman-Oliynykov/ciphers-speed

3.7.2. Cài đặt phần cứng

MKV-128 đã được cài đặt trên chip FPGA Kintex-7 (xc7k160tfbg676-3) sử dụng công cụ Vivado 2015. Hơn nữa, nhóm thiết kế cũng cài đặt mã khối AES-128 trên cùng môi trường. Ngoài ra, một số cài đặt của các tác giả khác về quá trình mã hóa của AES được nhắc lại để so sánh với các cài đặt nhận được. Để đánh giá độ hiệu quả cài đặt, thông số Mbps/Slice được sử dụng.

Bảng 21 - Cài đặt phần cứng của quá trình mã hóa.

Cài đặt	Thiết bị	Số Slice	Tần số (MHz)	Số Clock	Tốc độ (Mbits/s)	Mbps/Slice
Cài đặt của nhóm thiết kế						
MKV-128/128	Kintex-7 xc7k160tfbg676-3	370	309	18	2197	5,93
MKV-128/192	Kintex-7 xc7k160tfbg676-3	387	309	21	1883	4,86
MKV-128/256	Kintex-7 xc7k160tfbg676-3	393	309	24	1648	4,19
AES-128	Kintex-7 xc7k160tfbg676-3	446	429	31	1771	3,97
Các cài đặt khác cho AES-128						
P. B. Ghewari 2010 [99]	XCV600	1853	140.390	51	352	0,19
Z. Yuan 2011 [100]	Virtex-5 XCVLX30	885	130.3	55	300	0,338
T. Hoang và V. L. Nguyen 2012 [101]	Altera APEX20KC	895	120.656	13	1188	1,327
Y. J. Minal, và M. A. Sayyad 2014 [102]	Spartan-3 XC3S400-FG456	1403	160.875	10	2059	1,467
P. N. Khose and V. G. Raut 2015 [103]	spatran-6 XC6SLX16-3-CSG324	554	277.369	177	200	0,361
E. Kavitha 2016 [104]	spartan3e XC3S500E	2227	101.47	30	433	0,194
Alshaima Q. Al-Khafaji	Virtex-6 xc6vlx195t-3	423	191.795	10	2457	5,8

2019 [105]						

Bảng 22 - Cài đặt phần cứng của quá trình giải mã.

Cài đặt	Thiết bị	Số Slice	Tần số (MHz)	Số Clock	Tốc độ (Mbits/s)	Mbps/Slice
Cài đặt của nhóm thiết kế						
MKV-128/128	Kintex-7 xc7k160tfbg676-3	370	309	18	2197	5,93
MKV-128/192	Kintex-7 xc7k160tfbg676-3	387	309	21	1883	4,86
MKV-128/256	Kintex-7 xc7k160tfbg676-3	393	309	24	1648	4,19
AES-128	Kintex-7 xc7k160tfbg676-3	456	334	31	1379	3

Kết quả cho thấy MKV-128 khi cài đặt sử dụng cùng một kỹ thuật với AES-128 đều cho chỉ số Mbps/Slice cao hơn. Hơn nữa, kết quả này cũng có thể so sánh được với các cài đặt AES-128 tiên tiến khác trên thế giới. Tiếp theo, phiên bản MKV-256 cũng được cài đặt trên chip FPGA Kintex-7 (xc7k160tfbg676-3) sử dụng công cụ Vivado 2015.

Bảng 23 - Quá trình mã hóa/giải mã của MKV-256

Cài đặt	Thiết bị	Số Slice	Tần số (MHz)	Số Clock	Tốc độ (Mbits/s)	Mbps/Slice
Cài đặt của nhóm thiết kế						
MKV-256/256	Kintex-7 xc7k160tfbg676-3	1791	250	18	3555	1,98
MKV-256/384	Kintex-7 xc7k160tfbg676-3	1863	250	21	3047	1,63
MKV-256/512	Kintex-7 xc7k160tfbg676-3	1896	250	24	2666	1,4

Kết quả cho thấy MKV-256 có tốc độ thực thi khá cao và có chi phí phù hợp triển khai trên các nền tảng phần cứng trong các ứng dụng thực tế. Hơn nữa, tài nguyên khi tổng hợp trên một số Kit thông dụng được thống kê trong Bảng 24.

Bảng 24 - Cài đặt phần cứng MKV-256 trên một số Kit thông dụng.

Cài đặt	Board	Số LUT	Số FF	Số BRAM
MKV-256	ZedBoard Zynq board (xc7z020clg484-1)	8%	2%	28%
	Artix-7 AC701 board (xc7a200tfbg676-2)	3%	1%	11%
	Kintex-7 KC705 board (xc7k325tffg900-2)	2%	1%	9%
	Virtex-7 VC709 board (xc7vx690tffg1761-2)	1%	1%	3%
	ZYNQ-7 ZC706 Board (xc7z045ffg900-2)	2%	1%	7%

4. Giải thích nội dung TCVN

Nội dung chi tiết của Dự thảo được trình bày trong bản Dự thảo riêng kèm theo Thuyết minh này. Dự thảo tiêu chuẩn bao gồm 04 Điều và 01 phụ lục, cụ thể như sau:

LỜI NÓI ĐẦU

LỚI GIỚI THIỆU

1 PHẠM VI ÁP DỤNG

2 THUẬT NGỮ VÀ ĐỊNH NGHĨA

3 CÁC KÍ HIỆU VÀ THUẬT NGỮ VIẾT TẮT

4 MÃ KHỐI MKV

PHỤ LỤC A (QUY ĐỊNH) VÉC TƠ KIỂM TRA CHO MKV

5. Khuyến nghị áp dụng TCVN

Tiêu chuẩn này áp được khuyến cáo áp dụng cho khu vực kinh tế xã hội với đối tượng là mọi tổ chức, cá nhân, công dân Việt nam và công dân nước ngoài làm việc tại Việt nam để bảo mật thông tin không thuộc phạm vi bí mật nhà nước.

Tiêu chuẩn này sẽ giúp các cơ quan chính phủ, các doanh nghiệp và đặc biệt là các tổ chức đánh giá (ví dụ các phòng đo kiểm quốc gia) có thể dựa vào đó để thực hiện đánh giá được chất lượng của sản phẩm mật mã đang sử dụng trong nước. Nó cũng là một hướng dẫn giúp cho các doanh nghiệp trong việc phát triển các sản phẩm đảm bảo các yêu cầu về an toàn và bảo mật thông tin.

Góp phần tạo sự duy nhất và thống nhất trong các hệ thống bảo mật thông tin trên toàn lãnh thổ Việt Nam. Đảm bảo cho việc chỉ đạo, lãnh đạo, điều hành thống nhất, xuyên suốt và có độ tin tưởng cao theo khía cạnh bảo mật thông tin.

Là bước tiến trong quá trình làm chủ khoa học - công nghệ lõi, bởi trong bất kỳ sản phẩm bảo mật thông tin nào, thuật toán mật mã luôn đóng vai trò là hạt nhân. Việc

làm chủ công nghệ lõi tạo thuận lợi cho quá trình kiểm định chất lượng sản phẩm. Góp phần hoàn thiện xu thế "Make in Vietnam" đã được Thủ tướng Chính phủ chỉ đạo.

Tạo niềm tin cho các doanh nghiệp, tổ chức xã hội về việc thông tin của họ được mã hóa bởi một tiêu chuẩn mật mã của chính Việt Nam. Trả về cho người chủ sở hữu thông tin đúng nghĩa đen của việc họ là người chủ sở hữu thực sự thông tin đó.

Tạo sự thống nhất cho các thiết bị, hệ thống bảo mật thông tin nhập khẩu từ nước ngoài sẽ sử dụng chính tiêu chuẩn mật mã của Việt Nam.

Khẳng định vị thế của Việt Nam trên bản đồ mật mã quốc tế.

TCVN về mã khối là bước khởi đầu cho việc xây dựng một khung mật mã hoàn chỉnh (gồm các nguyên thủy mật mã khác) với mục tiêu tiến tới chuẩn hóa phục vụ hội nhập khu vực Asean và quốc tế.

PHỤ LỤC A

THUYẾT MINH CHI TIẾT XÂY DỰNG S-HỘP VÀ MA TRẬN MDS

1. Các tính chất mật mã của S-hộp 8 bit được sử dụng

Các tính chất mật mã của S-hộp được tính toán bằng công cụ PEIGE trong [106] như sau:

Bảng A.1 – Một số tính chất mật mã của S-hộp 8-bit

Cipher	MKV	Kalyna (pi1/pi2/ pi3/pi0)	Streebog	AES	BelT	SMS4	Kuznyechik	ARIA_s2	Camellia	SEED_S1	E2	CLEFIA_S0
Permutation	**TRUE**											
Involution	**FALSE**											
Diff	6	8/8/8/8	8	4	8	4	8	4	4	4	10	10
DiffFreq	94	9/7/9/15	25	255	28	255	25	255	255	255	1	9
Diff1	4	6/4/6/15	4	2	4	2	4	4	4	2	4	0
CardD1	32	27	29	24	32	30	29	29	28	35	20	0
Lin	44	48	56	32	52	32	56	32	32	32	56	56
LinFreq	58	28	14	1275	10	1275	14	1275	1275	1275	9	52
Lin1	36	44	36	32	32	32	36	24	28	28	28	0
CardL1	59	58	59	60	60	61	59	60	60	57	51	0
max_degree	7	7	7	7	7	7	7	7	7	7	7	6
min_degree	7	7	7	7	6	7	7	7	7	7	6	6
MaxDegreeFreq	255	255	255	255	254	255	255	255	255	255	254	255
MinDegreeFreq	255	255	255	255	1	255	255	255	255	255	1	255
Max_ProductDegrees	**[7|7|7|7|7|7|7|8]**											[6|7|7|7|7|7|7|8]
LS_number	**0**											
max_v (v, w)-linear	**(1, 7)**											
max_w (v, w)-linear	**(1, 7)**											
inv_max_degree	7	7	7	7	7	7	7	7	7	7	7	6
inv_min_degree	7	7	7	7	6	7	7	7	7	7	6	6
inv_MaxDegreeFreq	255	255	255	255	254	255	255	255	255	255	254	255
inv_MinDegreeFreq	255	255	255	255	1	255	255	255	255	255	1	255
inv_Max_ProductDegrees	**[7|7|7|7|7|7|7|8]**											[6|7|7|7|7|7|7|8]
inv_LS_number	**0**											
inv_max_v (v, w)-linear	**(1, 7)**											

2. Xây dựng ma trận MDS cho tầng tuyến tính *D*

Để xây dựng một ma trận cho một tầng tuyến phục vụ tính khuếch tán của một mã khối, chúng ta có nhiều phương pháp khác nhau. Trong đó, các biến đổi tuyến tính có ma trận biểu diễn là MDS sẽ đảm bảo tính khuếch tán cao nhất thường được lựa chọn,

chúng được thể hiện ở các dạng khác nhau, như ma trận MDS dịch vòng, Hadamard hay ma trận MDS là lũy thừa của ma trận đồng hành, ... Một thực tế tồn tại là khi tăng kích thước ma trận, việc kiểm soát các hệ số của nó để đảm bảo cài đặt hiệu quả vẫn là bài toán được nhiều nhà mật mã quan tâm. Ví dụ như trong AES, hệ số trong các ma trận ở biến đổi MixColumns và InvMixColumns đem đến sự mất cân bằng trong cài đặt dạng bitslice ở quá trình mã hóa và giải mã của thuật toán này. Vấn đề tương tự cũng xuất hiện trong thuật toán Kalyna. Các ma trận MDS dạng Hadamard cuộn, Hadamard được đề xuất để giải quyết vấn đề trên nhưng điểm cố hữu của nó là chúng tồn tại nhiều điểm bất động, đây có thể là điểm yếu có thể khai thác của những thám mã trong tương lai. Một cách khác giải quyết vấn đề mất cân bằng thực thi giữa quá trình mã hóa và giải mã là sử dụng các ma trận MDS là lũy thừa của ma trận đồng hành. Điều kiện đặt ra là các ma trận đồng hành với đa thức liên kết có dạng đối xứng. Một ví dụ điển hình cho cách giải quyết theo hướng này có thể tìm thấy trong tầng tuyến tính của thuật toán Kuznyechik trong chuẩn GOST R 34.12-2015 của Liên bang Nga [9]. Sử dụng cách tiếp cận này, nhóm thiết kế đã thực hiện xây dựng hai ma trận MDS trường $\mathbb{F}_{2^8}$ trong đó một có kích thước 4×4 cho MKV-128 và một có kích thước 8×8 cho MKV-256.

2.1. Đối với ma trận MDS 4×4 của MKV-128

Như đã đề cập ở trên, nhóm thiết kế dùng dạng ma trận đồng hành có trong [69] và phương pháp duyệt lần lượt trên tập phần tử có lợi thế cài đặt, và đã nhận được ma trận A_4 và ma trận MDS $M_4 = A_4^4$. Để chắc chắn rằng ma trận M_4 là MDS, nhóm thiết kế sử dụng thuật toán kiểm tra được đề xuất trong [107]. Kết quả kiểm tra bằng thực nghiệm cho thấy đây là một ma trận MDS. Ngoài ra nhóm thiết kế cũng kiểm tra được rằng hạng của ma trận $(M - I)$ bằng 4. Do vậy, khi sử dụng ma trận này để xây dựng tầng tuyến tính sẽ không cho điểm bất động (ngoại trừ điểm bất động tầm thường là véc tơ 0). Trên thực tế, ma trận MDS là lũy thừa ma trận đồng hành tồn tại với số lượng lớn. Tuy nhiên nhóm thiết kế lựa chọn ma trận dạng này vì lý do liên quan không chỉ độ an toàn, mà còn cả khả năng cài đặt của chúng trên các môi trường khác nhau. Thật vậy,

Đối với cài đặt không tra bảng (nonlockup tables). Đây là phương pháp cài đặt không dùng các bảng tính trước, mà thay vào đó thực hiện các phép toán logic trực tiếp trên dữ liệu byte, bit. Phương pháp cài đặt này thường được áp dụng để tích hợp nguyên thủy mật mã trên môi trường với tài nguyên hạn chế. Thuật toán mã khối AES với ma trận trong biến đổi MixColumn khi cài đặt trên môi trường thanh ghi 8 bit cũng là dạng cài đặt bitslice [108]. Tham số đánh giá trong phép cài đặt này là số phép toán XOR các từ 8-bit và phép Xtime. Phép Xtime là phép nhân với phần tử 2 hoặc 2^{-1} trên trường hữu hạn. Sở dĩ phép này được sử dụng là do chúng được cài đặt đơn giản. Với trường cơ sở $\mathbb{F}_{2^8}$ có đa thức sinh nguyên thủy $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$, phép nhân phần tử a với 2 và 2^{-1} được thực hiện như sau: Cụ thể phép nhân phần tử a

với 2 được thực hiện sau 2 bước như giả code sau, trong đó giá trị 101011_2 và 10010101_2 tương ứng với $left_f = x^5 \oplus x^3 \oplus x \oplus 1$ và $right_f = (x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1) \gg 1 = x^7 \oplus x^4 \oplus x^2 \oplus 1$:

Giả code nhân với 2 *trên trường* $\mathbb{F}_{2^8}$.

1. if $(x\& 10000000_2) != 0)$ return $\left((x \ll 1) \oplus 101011_2\right)\& 11111111_2$
2. else return $(x \ll 1)\& 11111111_2$

Giả code nhân với 2^{-1} *trên trường* $\mathbb{F}_{2^8}$.

1. if $(x\&00000001) != 0$ return $(x \gg 1)\& 10010101_2$
2. else return $(x \gg 1)$.

Trong mã khối đề xuất, khi kích thước khối là 128 bit, phép biến đổi MixWords được thực hiện như sau $Y = A^4 \times X$, cụ thể:

$$\underbrace{\begin{pmatrix} y_0 & y_4 & y_8 & y_{12} \\ y_1 & y_5 & y_9 & y_{13} \\ y_2 & y_6 & y_{10} & y_{14} \\ y_3 & y_7 & y_{11} & y_{15} \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix}^4}_{A^4} \times \underbrace{\begin{pmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{pmatrix}}_{X} \quad \text{(A.1)}$$

Xét phép nhân

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \Leftrightarrow \begin{cases} z_0 = x_1 \\ z_1 = x_2 \\ z_2 = x_3 \\ z_3 = x_0 \oplus 2x_1 \oplus x_2 \oplus 3x_3 \end{cases}$$

Như vậy:

$$\begin{pmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix}^2 \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix} \times \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

$$\Leftrightarrow \begin{cases} t_0 = z_1 = x_2 \\ t_1 = z_2 = x_3 \\ t_2 = z_3 = x_0 \oplus 2x_1 \oplus x_2 \oplus 3x_3 \\ t_3 = z_0 \oplus 2z_1 \oplus z_2 \oplus 3z_3 = x_1 \oplus 2x_2 \oplus x_3 \oplus 3t_2 \end{cases}$$

Thực hiện tương tự ta có

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 2 & 1 & 3 \end{pmatrix}^4 \times \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \Leftrightarrow \begin{cases} y_0 = x_0 \oplus 2x_1 \oplus x_2 \oplus 3x_3 \\ y_1 = x_1 \oplus 2x_2 \oplus x_3 \oplus 3y_0 \\ y_2 = x_2 \oplus 2x_3 \oplus y_0 \oplus 3y_1 \\ y_3 = x_3 \oplus 2y_0 \oplus y_1 \oplus 3y_2 \end{cases}$$

$$\Leftrightarrow \begin{cases} y_0 = x_0 \oplus x_2 \oplus x_3 \oplus 2(x_1 \oplus x_3) \\ y_1 = x_1 \oplus x_3 \oplus y_0 \oplus 2(x_2 \oplus y_0) \\ y_2 = x_2 \oplus y_0 \oplus y_1 \oplus 2(x_3 \oplus y_1) \\ y_3 = x_3 \oplus y_1 \oplus y_2 \oplus 2(y_0 \oplus y_2) \end{cases} \quad \text{(A.2)}$$

Thực hiện tương tự ta sẽ tính được các phần tử khác của ma trận đầu ra *Y*, chúng sẽ có dạng truy hồi phần tử sau qua phần tử trước như trong biểu thức (1.4). Độ phức tạp của cài đặt (A.2) là 16 phép XOR hai phần tử trong trường $\mathbb{F}_{2^8}$ và 4 phép Xtime. Do vậy cài đặt (A.1) yêu cầu $4 \times 16 = 48$ phép XOR và $4 \times 4 = 16$ phép Xtimes. Yêu cầu về tài nguyên này là giống như cài đặt phép biến đổi MixColumns của AES [108].

Quay trở lại với phép biến đổi InvMixWords trong mã khối dân sự phiên bản 128 bit. Thấy rằng, để thực hiện phép này ta cần tính:

$$\underbrace{\begin{pmatrix} y_0 & y_4 & y_8 & y_{12} \\ y_1 & y_5 & y_9 & y_{13} \\ y_2 & y_6 & y_{10} & y_{14} \\ y_3 & y_7 & y_{11} & y_{15} \end{pmatrix}}_{Y} = \underbrace{\begin{pmatrix} 2 & 1 & 3 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}^4}_{A^{-4}} \times \underbrace{\begin{pmatrix} x_0 & x_4 & x_8 & x_{12} \\ x_1 & x_5 & x_9 & x_{13} \\ x_2 & x_6 & x_{10} & x_{14} \\ x_3 & x_7 & x_{11} & x_{15} \end{pmatrix}}_{X} \quad \text{(A.3)}$$

Tương tự như trên ta có:

$$\begin{cases} y_3 = x_1 \oplus x_2 \oplus x_3 \oplus 2(x_0 \oplus x_2) \\ y_2 = x_0 \oplus x_1 \oplus x_2 \oplus 2(y_3 \oplus x_1) \\ y_1 = y_3 \oplus x_0 \oplus x_1 \oplus 2(y_2 \oplus x_0) \\ y_0 = y_2 \oplus y_3 \oplus x_0 \oplus 2(y_1 \oplus y_3) \end{cases} \quad \text{(A.4)}$$

Từ biểu thức (A.4) thấy rằng phép (A.3) được cài đặt với độ phức tạp tương tự như của phép (A.1). Như vậy, với dạng ma trận đồng hành đề xuất, phép MixWords và InvMixWords có độ phức tạp cài đặt trên môi trường thanh ghi 8 bit là như nhau. Nói cách khác, tốc độ thực thi cho cả quá trình mã hóa và giải mã là giống nhau. Đây là tính chất mà AES không có được bởi dạng ma trận sử dụng trong biến đổi MixColumns của nó là dạng dịch vòng và không có tính cuộn (tính tự nghịch đảo). Cụ thể trong phép biến đổi InvMixColumns của AES, các hệ số của ma trận MDS không thuận tiện cho việc thực hiện nhân nhanh trên trường hữu hạn, và do vậy nó tốn tài nguyên cài đặt hơn. Nói cách khác, tốc độ trong quá trình mã hóa nhanh hơn so với giải mã của AES.

Mặt khác trong [109] đã chỉ ra rằng, dạng ma trận mà AES sử dụng trong tầng tuyến tính của nó có 2^{16} điểm bất động, còn với ma trận đề xuất là không có.

Đối với cài đặt trên môi trường thanh ghi 32 bit. Thứ tự biến đổi trong một vòng mã của mã khối dân sự (không tính biến đổi cộng khóa *AddRoundKeys*) được thực hiện từ trên xuống dưới như sau:

SubCells
MixWords
MixColums
XWords

Tương tự như nguyên tắc cài đặt của AES khi mà biến đổi tuyến tính và phi tuyến được kết hợp với nhau và được biểu diễn dưới dạng các bảng tra [108]. Ở đây, nhóm thiết kế cũng sẽ kết hợp phép biến đổi SubCells và biến đổi MixWords theo nguyên tắc như vậy. Với các tiếp cận này, cho phép thuật toán cài đặt hiệu quả trên môi trường thanh ghi 32 bit. Ký hiệu *e* là dữ liệu đầu ra qua biến đổi *MixWords* khi thực hiện đối với khối đầu vào *a ở biến đổi SubCells.* Khi đó cột thứ *j* trong *e* được tính như sau:

$$\begin{bmatrix} e_{0j} \\ e_{1j} \\ e_{2j} \\ e_{3j} \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{23} & m_{33} \end{bmatrix} \begin{bmatrix} c_{0j} \\ c_{1j} \\ c_{2j} \\ c_{3j} \end{bmatrix}, j = 0 \div 3, \quad \text{(A.5)}$$

trong đó *c* – kết quả của biến đổi *SubCells* và $M = \left(m_{ij}\right)_{4\times4}, 0 \le i,j \le 3$ là ma trận MDS mà chúng ta đề xuất (xem biểu thức (A.1)). Vì $c_{ij} = S[a_{ij}]$

Biến đổi *XORWord*s và *SubCells* có thể viết dưới dạng nên từ (11) ta có:

$$\begin{bmatrix} e_{0j} \\ e_{1j} \\ e_{2j} \\ e_{3j} \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} & m_{02} & m_{03} \\ m_{10} & m_{11} & m_{12} & m_{13} \\ m_{20} & m_{21} & m_{22} & m_{23} \\ m_{30} & m_{31} & m_{23} & m_{33} \end{bmatrix} \begin{bmatrix} S[a_{j0}] \\ S[a_{j1}] \\ S[a_{j2}] \\ S[a_{j3}] \end{bmatrix}, j = 0 \div 3$$

Biểu thức này có thể viết dưới dạng:

$$\begin{bmatrix} e_{0j} \\ e_{1j} \\ e_{2j} \\ e_{3j} \end{bmatrix} = S[a_{j0}] \begin{bmatrix} m_{00} \\ m_{10} \\ m_{20} \\ m_{30} \end{bmatrix} \oplus S[a_{j1}] \begin{bmatrix} m_{01} \\ m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \oplus S[a_{j2}] \begin{bmatrix} m_{02} \\ m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} \oplus S[a_{j3}] \begin{bmatrix} m_{03} \\ m_{13} \\ m_{23} \\ m_{33} \end{bmatrix},$$

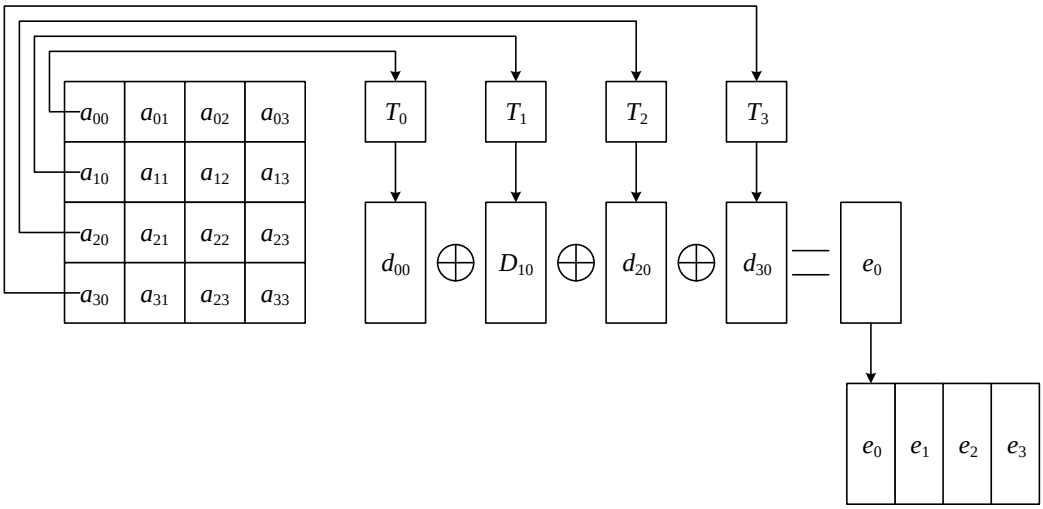
trong đó $S[a_{ij}]$- tương ứng với giá trị 8 bit nhận được qua các S-hộp. Ký hiệu

$$T_0[a] = \begin{bmatrix} S[a]\cdot m_{00} \\ S[a]\cdot m_{10} \\ S[a]\cdot m_{20} \\ S[a]\cdot m_{30} \end{bmatrix}, T_1[a] = \begin{bmatrix} S[a]\cdot m_{01} \\ S[a]\cdot m_{11} \\ S[a]\cdot m_{21} \\ S[a]\cdot m_{32} \end{bmatrix}, T_2[a] = \begin{bmatrix} S[a]\cdot m_{02} \\ S[a]\cdot m_{12} \\ S[a]\cdot m_{22} \\ S[a]\cdot m_{32} \end{bmatrix}, T_3[a] = \begin{bmatrix} S[a]\cdot m_{03} \\ S[a]\cdot m_{13} \\ S[a]\cdot m_{23} \\ S[a]\cdot m_{33} \end{bmatrix}$$

là các bảng tra, trong đó $a \in \mathbb{F}_{2^8}$. Do vậy mỗi bảng gồm 256 phần tử, mỗi phần tử là các số 32 bit. Bộ nhớ cần thiết để lưu toàn bộ 4 bảng này là $4 \times 256 \times 4byte = 4096byte = 4Kbyte$. Như vậy 2 biến đổi *SubCells* và *MixWords* trong một vòng mã có thể thực hiện theo biểu thức sau:

$$e_j = T_0[a_{j0}] \oplus T_1[a_{j1}] \oplus T_2[a_{j2}] \oplus T_3[a_{j3}]$$

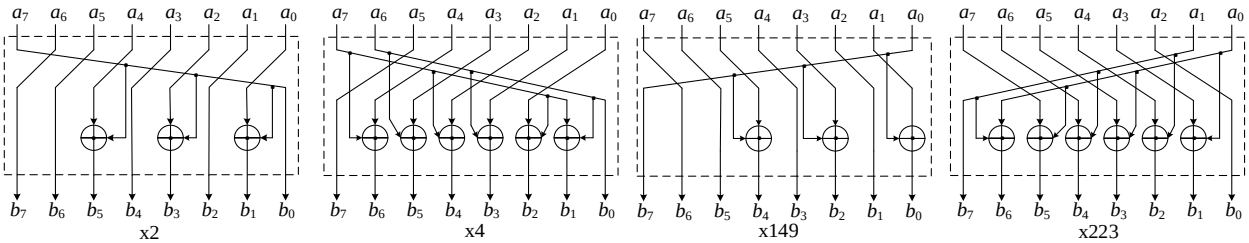
Hình A.2 là minh họa cho quá trình này



Hình A.2 - Minh họa cài đặt trên thanh ghi 32 bit của cấu trúc đề xuất khi tính cột đầu tiên của ma trận trạng thái

Như vậy ta thấy việc lập bảng và gộp các biến đổi lại cho phép cài đặt hiệu quả trên các thanh ghi 32 bit. Bộ nhớ yêu cầu để lưu các bảng này là 4 Kbyte.

Đối với Cài đặt phần cứng. Thấy rằng độ phức tạp trong cài đặt mỗi biểu thức này chính là phép nhân với phần tử trên trường hữu hạn. Trong [5] đã chỉ ra một số lợi thế của việc lựa chọn đa thức sinh của trường hữu hạn cho phép thực thi nhanh một số phép nhân với phần tử của trường. Cụ thể trong nghiên cứu này đã chỉ ra rằng, với trường hữu hạn $\mathbb{F}_{2^8}$, đa thức sinh là $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$, phép nhân với 2, $2^2 = 4$, $2^{-1} = 149$ và $2^{-2} = 223$ là dễ dàng thực hiện trong 1 xung nhịp (clock). Hình 1.3 là minh họa cho quá trình này, trong đó phép cộng $\oplus$ trong mỗi hình là phép cộng 2 bit đầu vào tạo ra 1 bit đầu ra.



Hình A.3 - Minh họa phép nhân với 2, 4, 149 và 223 trên $\mathbb{F}_{2^8}$ với $f(x) = x^8 \oplus x^5 \oplus x^3 \oplus x \oplus 1$

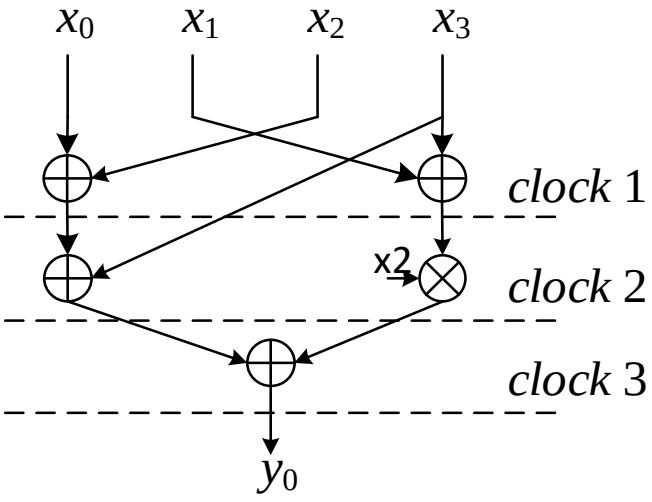
Trong biểu thức (A.1) và (A.4) chúng ta chỉ quan tâm đến phép nhân với 2. Tuy nhiên với các đề xuất ma trận khác, việc đưa về phép nhân với 4, 149 hoặc 223 cũng sẽ mang lại lợi thế cài đặt cho những ma trận này.

Để trực quan chúng ta viết lại các biểu thức này, trong đó biểu thức (A.6) tương ứng được sử dụng để tính cột đầu tiên trong ma trận trạng thái ở biến đổi MixWords, còn (A.7) tương ứng để tính cột đầu tiên trong ma trận trạng thái ở biến đổi InvMixWords. Lưu ý là phép cộng $\oplus$ ở mỗi biểu thức này là phép cộng theo modulo 2 của hai từ 8 bit và tạo ra 1 từ 8 bit.

$$\begin{cases} y_0 = x_0 \oplus x_2 \oplus x_3 \oplus 2(x_1 \oplus x_3) \\ y_1 = x_1 \oplus x_3 \oplus y_0 \oplus 2(x_2 \oplus y_0) \\ y_2 = x_2 \oplus y_0 \oplus y_1 \oplus 2(x_3 \oplus y_1) \\ y_3 = x_3 \oplus y_1 \oplus y_2 \oplus 2(y_0 \oplus y_2) \end{cases} \quad \text{(A.6)}$$

$$\begin{cases} y_3 = x_1 \oplus x_2 \oplus x_3 \oplus 2(x_0 \oplus x_2) \\ y_2 = x_0 \oplus x_1 \oplus x_2 \oplus 2(y_3 \oplus x_1) \\ y_1 = y_3 \oplus x_0 \oplus x_1 \oplus 2(y_2 \oplus x_0) \\ y_0 = y_2 \oplus y_3 \oplus x_0 \oplus 2(y_1 \oplus y_3) \end{cases} \quad \text{(A.7)}$$

Sơ đồ cài đặt cho mỗi phương trình trong các biểu thức này là giống nhau, ví dụ hình A.4 là minh họa sơ đồ tính y_0, với $y_0 = x_0 + x_2 + x_3 + 2(x_1 + x_3)$:



Hình A.4: Minh họa tính giá trị y_0.

Thấy rằng để tính được y_0 cần 4 phép XOR 2 số 8 bit, 1 phép nhân 2 và cần 3 xung nhịp. Tổng hợp lại cần $4 \times 8 + 3 = 35$ phép XOR 2 bit và 3 xung nhịp. Để tính được toàn bộ 1 cột của ma trận trạng thái cần $35 \times 4 = 140$ phép XOR 2 bit và $3 \times 4 = 12$ xung nhịp. Để tính toàn bộ ma trận trạng thái qua biến đổi MixWords cần $140 \times 4 = 560$ phép XOR 2 bit và 12 xung nhịp (vì quá trình tính 4 cột trong ma trận trạng thái là độc lập). Độ phức tạp cài đặt này cũng tương tự đối với phép InvMixWords ở biểu thức (A.7). Trên thực tế triển khai cài đặt phần cứng có thể sử dụng bộ cộng với nhiều

đầu vào hơn, như thế có thể tăng tốc quá trình tính toán đầu ra của biến đổi. Ở đây nhóm thiết kế chỉ đưa ra phân tích lý thuyết sơ bộ.

2.2. Đối với ma trận MDS 8×8 cho MKV-256

Tương tự như đối với phiến bản 128-bit, nhóm thiết kế dùng dạng ma trận đồng hành có trong [69] và cũng dùng phương pháp duyệt lần lượt trên tập phần tử có lợi thế cài đặt và đã nhận được ma trận A_8 và ma trận MDS $M_8 = A_8^8$. Khi sử dụng các ma trận này trong biến đổi MixWords và nghịch đảo của nó sẽ cho số nhánh bằng 9. Nhóm thiết kế cũng tính số điểm bất động của tầng tuyến tính sử dụng các ma trận này, kết quả cho thấy hạng của ma trận $M_8 - I$ bằng 8, do vậy sẽ không có điểm bất động nào (ngoại trừ 1 điểm là véc tơ 0 tầm thường). Tiếp theo, nhóm thiết kế sẽ phân tích khả năng cài đặt của ma trận này.

Đối với cài đặt không tra bảng. Quá trình phân tích tương tự đối với ma trận cho MKV-128, chú ý rằng

$$\begin{cases} 219 = 223 \oplus 4 = 149 \otimes 149 \oplus 4 = 2^{-1} \otimes 2^{-1} \oplus 2^2 \\ 12 = 8 \oplus 4 = 2^3 \oplus 2^2 \\ 20 = 16 \oplus 4 = 2^4 \oplus 2^2 \end{cases}$$

Nên phép nhân với các phần tử của ma trận đồng hành A_8 đều đưa về phép Xtime và phép XOR với sự thực thi đơn giản. Ví dụ để tính một cột trong ma trận trạng thái đầu ra của biến đổi MixWords, ta thực hiện theo biểu thức sau:

$$\begin{cases} y_0 = x_0 \oplus 2^2\left(x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus 2(x_3 \oplus x_5) \oplus 2^2 x_4\right) \oplus 2^{-2}(x_2 \oplus x_6) \\ y_1 = x_1 \oplus 2^2\left(x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus y_0 \oplus 2(x_4 \oplus x_6) \oplus 2^2 x_5\right) \oplus 2^{-2}(x_3 \oplus x_7) \\ y_2 = x_2 \oplus 2^2\left(x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus y_0 \oplus y_1 \oplus 2(x_5 \oplus x_7) \oplus 2^2 x_6\right) \oplus 2^{-2}(x_4 \oplus y_0) \\ y_3 = x_3 \oplus 2^2\left(x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus y_0 \oplus y_1 \oplus y_2 \oplus 2(x_6 \oplus y_0) \oplus 2^2 x_7\right) \oplus 2^{-2}(x_5 \oplus y_1) \\ y_4 = x_4 \oplus 2^2\left(x_5 \oplus x_6 \oplus x_7 \oplus y_0 \oplus y_1 \oplus y_2 \oplus y_3 \oplus 2(x_7 \oplus y_1) \oplus 2^2 y_0\right) \oplus 2^{-2}(x_6 \oplus y_2) \\ y_5 = x_5 \oplus 2^2\left(x_6 \oplus x_7 \oplus y_0 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus 2(y_0 \oplus y_2) \oplus 2^2 y_1\right) \oplus 2^{-2}(x_7 \oplus y_3) \\ y_6 = x_6 \oplus 2^2\left(x_7 \oplus y_0 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus 2(y_1 \oplus y_3) \oplus 2^2 y_2\right) \oplus 2^{-2}(y_0 \oplus y_4) \\ y_7 = x_7 \oplus 2^2\left(y_0 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus 2(y_2 \oplus y_4) \oplus 2^2 y_3\right) \oplus 2^{-2}(y_1 \oplus y_5) \end{cases}$$

(A.8)

Để tính được (A.8) cần $12 \times 8 = 96$ phép XOR hai từ 8 bit và $7 \times 8 = 56$ phép Xtime. Để tính được toàn bộ ma trận trạng thái qua phép MixWords cần $96 \times 4 = 384$ phép XOR và $56 \times 4 = 224$ phép Xtimes. Độ phức tạp cài đặt này cũng chính là độ phức tạp cho cài đặt phép InvMixWords vì ma trận tuyến tính của chúng có các phần tử giống nhau.

Đối với cài đặt trên môi trường thanh ghi 64 bit. Đây cũng là giải pháp cài đặt sử dụng kỹ thuật tra bảng. Tuy nhiên ma trận tuyến tính trong phần này là 8×8 nên mỗi phần tử trong bảng tra sẽ có kích thước 64 bit. Áp dụng tương tự phần trước, để cài đặt biến đổi MixWords cần 8 bảng tra, mỗi bảng tra 256 phần tử, mỗi phần tử 8 byte. Tổng sẽ cần $8 \times 256 \times 8\ byte = 16\ Kbyte$. Cả quá trình mã hóa và giải mã sẽ cần 32 Kbyte. Kỹ thuật sử dụng bảng tra này là tương tự như ở mục trên và giống với kỹ thuật cài đặt các thuật toán AES, GOST R 34.12.2015, Kalyna, LED, ... nên nhóm thiết kế không trình bày chi tiết.

Đối với cài đặt phần cứng. Cài đặt phần cứng cũng sử dụng biểu thức (A.8) như đối với MixWords trong phiên bản mã khối với kích thước khối 128 bit. Rõ ràng biểu thức (A.8) với các phép nhân đơn với phần tử của trường và phép cộng XOR là thuận tiện cho cài đặt cứng hóa. Phép nhân với 2 cần 3 cổng XOR 2 bit, nhân với 4 hoặc 224 cần 6 cổng XOR 2 bit do vậy, số phép XOR 2 bit cần sử dụng để thực thi MixWords là $384 \times 8 + 224 \times 3 = 3744$.

TÀI LIỆU THAM KHẢO

[1]. Cuong Nguyen, Anh Nguyen, Phong Trieu, Long Nguyen, and Lai Tran. *Analysis of a new practically secure SPN-based scheme in the Luby-Rackoff model*. in *The 9th International Conference on Future Data and Security Engineering*. 2022. Springer.

[2]. Cuong Nguyen, Nam Tran, and Long Nguyen. *FLC: A New Secure and Efficient SPN-Based Scheme for Block Ciphers*. in *2022 9th NAFOSTED Conference on Information and Computer Science (NICS)*. 2022. IEEE.

[3]. Tran Sy Nam, Nguyen Van Long, and Nguyen Bui Cuong. *An Optimized Bit-Slice Implementation of Secure 8-Bit Sbox Based on Butterfly Structure*. in *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*. 2023. IEEE.

[4]. Nguyễn Bùi Cương, Nguyễn Văn Long, and Trần Duy Lai, *Một số đặc trưng đại số của các S-hộp 4×4 bit tối ưu chống thám mã lượng sai và tuyến tính.* Tạp chí ứng dụng toán học, 2012. **10** (1): p. 1-18.

[5]. Nguyễn Ngọc Điệp, *Một đề xuất ma trận MDS 4x4 an toàn, hiệu quả cho tầng tuyến tính của các mã pháp dạng AES.* Tạp chí Nghiên cứu Khoa học và Công nghệ Quân sự, 2016. **Số 46/12-2016**: p. tr. 133-142.

[6]. Bui Cuong Nguyen and Tuan Anh Nguyen, *Evaluating pseudorandomness and superpseudorandomness of the iterative scheme to build SPN block cipher.* Journal of Science and Technology on Information security, 2017. **40**(2): p. 40.

[7]. Trần Sỹ Nam, Nguyễn Văn Long, and Nguyễn Bùi Cương, *Xây dựng tầng tuyến tính có cài đặt hiệu quả cho mã khối 128-bit có cấu trúc FLC*, in *Hội thảo nghiên cứu ứng dụng mật mã và an toàn thông tin*. 2022: Hà Nội.

[8]. Tran Sy Nam, Nguyen Van Long, and Nguyen Bui Cuong, *Đề xuất tầng tuyến tính và đánh giá khả năng cài đặt trong xây dựng mã khối 256-bit có cấu trúc FLC.* Journal of Science and Technology on Information security, 2022.

[9]. Jaechul Sung, *Differential cryptanalysis of eight-round SEED.* Information Processing Letters, 2011. **111**(10): p. 474-478.

[10]. Alex Biryukov, Léo Perrin, and Aleksei Udovenko. *Reverse-engineering the S-box of Streebog, Kuznyechik and STRIBOBr1*. in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. 2016. Springer.

[11]. Riham AlTawy, Onur Duman, and Amr M Youssef, *Fault analysis of Kuznyechik.* Математические вопросы криптографии, 2016. **7**(2): p. 21-34.

[12]. Riham AlTawy, Ahmed Abdelkhalek, and Amr M Youssef, *A meet-in-the-middle attack on reduced-round Kalyna-b/2b.* IEICE TRANSACTIONS on Information and Systems, 2016. **99**(4): p. 1246-1250.

[13]. Donghoon Chang, Mohona Ghosh, Aarushi Goel, and Somitra Kumar Sanadhya. *Single key recovery attacks on 9-round Kalyna-128/256 and Kalyna-256/512*. in *International Conference on Information Security and Cryptology*. 2015. Springer.

[14]. Li Lin and Wenling Wu, *Improved meet-in-the-middle attacks on reduced-round Kalyna-128/256 and Kalyna-256/512.* Designs, Codes and Cryptography, 2018. **86**(4): p. 721-741.
[15]. Philipp Jovanovic and Ilia Polian. *Fault-based attacks on the Bel-T block cipher family*. in *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. 2015. IEEE.
[16]. Muhammad ElSheikh, Ahmed Abdelkhalek, and Amr M Youssef. *On MILP-based automatic search for differential trails through modular additions with application to Bel-T.* in *Progress in Cryptology–AFRICACRYPT 2019: 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9–11, 2019, Proceedings 11*. 2019. Springer.
[17]. Wentao Zhang, Wenling Wu, Dengguo Feng, and Bozhan Su. *Some new observations on the SMS4 block cipher in the Chinese WAPI standard*. in *International Conference on Information Security Practice and Experience*. 2009. Springer.
[18]. Zhiqiang Liu, Dawu Gu, and Jing Zhang, *Multiple linear cryptanalysis of reduced-round SMS4 block cipher.* Chinese Journal of Electronics, 2010. **19**(3): p. 389-393.
[19]. Bo-Zhan Su, Wen-Ling Wu, and Wen-Tao Zhang, *Security of the SMS4 block cipher against differential cryptanalysis.* Journal of Computer Science and Technology, 2011. **26**(1): p. 130-138.
[20]. Jonathan Etrog and Matt JB Robshaw. *The cryptanalysis of reduced-round SMS4*. in *International Workshop on Selected Areas in Cryptography*. 2008. Springer.
[21]. Joo Yeon Cho and Kaisa Nyberg. *Improved linear cryptanalysis of SMS4 block cipher*. in *Symmetric Key Encryption Workshop*. 2011.
[22]. Bin Zhang and ChenHui Jin, *Practical security against linear cryptanalysis for SMS4-like ciphers with SP round function.* Science China Information Sciences, 2012. **55**(9): p. 2161-2170.
[23]. Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu, *Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties.* Cryptology ePrint Archive, Report, 2014. **747**: p. 2014.
[24]. Jiazhe Chen. *A Note on the Impossible Differential Attacks on Block Cipher SM4*. in *Computational Intelligence and Security (CIS), 2016 12th International Conference on*. 2016. IEEE.
[25]. Donghoon Chang, Mohona Ghosh, Aarushi Goel, and Somitra Kumar Sanadhya. *Improved Meet-in-the-Middle Attacks on 7 and 8-Round ARIA-192 and ARIA-256*. in *Progress in Cryptology--INDOCRYPT 2015: 16th International Conference on Cryptology in India, Bangalore, India, December 6-9, 2015, Proceedings*. 2015. Springer.
[26]. Henri Gilbert, Helena Handschuh, Antoine Joux, and Serge Vaudenay. *A statistical attack on RC6*. in *International Workshop on Fast Software Encryption*. 2000. Springer.

[27]. Lars R Knudsen and Willi Meier. *Correlations in RC6 with a reduced number of rounds*. in *International Workshop on Fast Software Encryption*. 2000. Springer.
[28]. Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. *LEA: A 128-bit block cipher for fast encryption on common processors*. in *Information Security Applications: 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers 14*. 2014. Springer.
[29]. Cihangir Tezcan. *The improbable differential attack: Cryptanalysis of reduced round CLEFIA*. in *Progress in Cryptology-INDOCRYPT 2010: 11th International Conference on Cryptology in India, Hyderabad, India, December 12-15, 2010. Proceedings 11*. 2010. Springer.
[30]. Mitsuru Matsui. *New block encryption algorithm MISTY*. in *International Workshop on Fast Software Encryption*. 1997. Springer.
[31]. Lars Knudsen and David Wagner. *Integral cryptanalysis*. in *International Workshop on Fast Software Encryption*. 2002. Springer.
[32]. Seyyed Arash Azimi, Siavash Ahmadi, Zahra Ahmadian, Javad Mohajeri, and Mohammad Reza Aref, *Improved impossible differential and biclique cryptanalysis of HIGHT*. International Journal of Communication Systems, 2018. **31**(1): p. e3382.
[33]. Farzaneh Abed, Eik List, Stefan Lucks, and Jakob Wenzel. *Differential cryptanalysis of round-reduced Simon and Speck*. in *International Workshop on Fast Software Encryption*. 2014. Springer.
[34]. Ling Song, Zhangjie Huang, and Qianqian Yang. *Automatic differential analysis of ARX block ciphers with application to SPECK and LEA*. in *Information Security and Privacy: 21st Australasian Conference, ACISP 2016, Melbourne, VIC, Australia, July 4-6, 2016, Proceedings, Part II*. 2016. Springer.
[35]. Céline Blondeau and Kaisa Nyberg. *Links between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities*. in *Eurocrypt*. 2014. Springer.
[36]. Richard P Feynman, *Simulating physics with computers*, in *Feynman and computation*. 2018, CRC Press. p. 133-153.
[37]. Reza Azarderakhsh. *PQSecure Technologies." PQSecure Technologies*. WP Release 2/12/2020 Available from: www.pqsecurity.com/white-paper-release-2-12-2020.
[38]. Global Risk Institute. *Quantum Threat Timeline Report 2020* 1 Feb. 2021; Available from: www.pqsecurity.com/white-paper-release-2-12-2020.
[39]. Manoj Kumar and Pratap Pattnaik. *Post quantum cryptography (PQC)-An overview*. in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. 2020. IEEE.
[40]. Ohood Saud Althobaiti and Mischa Dohler, *Cybersecurity challenges associated with the Internet of Things in a post-quantum world*. IEEE Access, 2020. **8**: p. 157356-157381.
[41]. Craig Gidney and Martin Ekerå, *How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits*. Quantum, 2021. **5**: p. 433.

[42]. Petros Wallden and Elham Kashefi, *Cyber security in the quantum era.* Communications of the ACM, 2019. **62**(4): p. 120-120.
[43]. Jiabo Wang, Ling Liu, Shanxiang Lyu, Zheng Wang, Mengfan Zheng, Fuchun Lin, Zhao Chen, Liuguo Yin, Xiaofu Wu, and Cong Ling, *Quantum-safe cryptography: crossroads of coding theory and cryptography.* Science China Information Sciences, 2022. **65**(1): p. 111301.
[44]. Tianqi Zhou, Jian Shen, Xiong Li, Chen Wang, and Jun Shen, *Quantum cryptography for the future internet and the security analysis.* Security and Communication Networks, 2018. **2018**: p. 1-7.
[45]. Tmilinovic. *Quantum computing timeline by Gartner*. 2018; Available from: tmilinovic.wordpress.com/2019/01/18/quantum-computing-timeline-by-gartner.
[46]. Global Risk Institute. *A Methodology for Quantum Risk Assessment* 8 Nov. 2017; Available from: https://globalriskinstitute.org/publication/a-methodology-for-quantum-risk-assessment/?lang=fr%2F.
[47]. Lukas Malina, Petr Dzurenda, Sara Ricci, Jan Hajny, Gautam Srivastava, Raimundas Matulevičius, Abasi-Amefon O Affia, Maryline Laurent, Nazatul Haque Sultan, and Qiang Tang, *Post-quantum era privacy protection for intelligent infrastructures.* IEEE Access, 2021. **9**: p. 36038-36077.
[48]. Michele Mosca and Privacy, *Cybersecurity in an era with quantum computers: will we be ready?* IEEE Security, 2018. **16**(5): p. 38-41.
[49]. Ward Beullens, Jan-Pieter D'Anvers, Andreas T Hülsing, Tanja Lange, Lorenz Panny, Cyprien de Saint Guilhem, and Nigel P Smart, *Post-Quantum Cryptography: Current state and quantum mitigation.* 2021.
[50]. Daya Sagar Gupta, Sangram Ray, Tajinder Singh, and Madhu Kumari, *Post-quantum lightweight identity-based two-party authenticated key exchange protocol for internet of vehicles with probable security.* Computer Communications, 2022. **181**: p. 69-79.
[51]. William Buchanan and Alan Woodward, *Will quantum computers be the end of public key encryption?* Journal of Cyber Security Technology, 2017. **1**(1): p. 1-22.
[52]. Liu-Jun Wang, Kai-Yi Zhang, Jia-Yong Wang, Jie Cheng, Yong-Hua Yang, Shi-Biao Tang, Di Yan, Yan-Lin Tang, Zhen Liu, and Yu Yu, *Experimental authentication of quantum key distribution with post-quantum cryptography.* Journal quantum information, 2021. **7**(1): p. 67.
[53]. Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, and David A Buell, *Quantum supremacy using a programmable superconducting processor.* Nature, 2019. **574**(7779): p. 505-510.
[54]. Lukas Malina, Lucie Popelova, Petr Dzurenda, Jan Hajny, and Zdenek Martinasek, *On feasibility of post-quantum cryptography on small devices.* IFAC-PapersOnLine, 2018. **51**(6): p. 462-467.
[55]. Kazutoshi Kan and Masashi Une, *Recent trends on research and development of quantum computers and standardization of post-quantum cryptography.* IMES Discussion Paper Series 21-E-05, Institute for Monetary and Economic Studies, Bank of Japan, 2021.
[56]. Muhammad ElSheikh, Mohamed Tolba, and Amr M Youssef. *Integral attacks on round-reduced Bel-T-256.* in *Selected Areas in Cryptography–SAC 2018:*

25th International Conference, Calgary, AB, Canada, August 15–17, 2018, Revised Selected Papers. 2019. Springer.
[57]. Elaine Barker and Quynh Dang, *Nist special publication 800-57 part 1, revision 5*, in *NIST, Tech. Rep*. 2020.
[58]. Cryptographic Mechanisms BSI, *Recommendations and Key Lengths.* Technical Guideline, 2020.
[59]. Michel Abdalla, Benedikt Gierlichs, Kenneth G Paterson, Vincent Rijmen, Ahmad-Reza Sadeghi, Nigel P Smart, Martijn Stam, Michael Ward, Bogdan Warinschi, and Gaven Watson, *Algorithms, key size and protocols report.* ECRYPT-CSA, Tech. Rep. H2020-ICT-2014–Project, 2018. **645421**.
[60]. ANSSI, *ANSSI views on the Post-Quantum Cryptography transition*. 2022.
[61]. Ward Beullens, Jan-Pieter D'Anvers, Andreas T Hülsing, Tanja Lange, Lorenz Panny, Cyprien de Saint Guilhem, and Nigel P Smart. *Post-Quantum Cryptography: Current state and quantum mitigation*. 2021.
[62]. NCSC, *Preparing for Quantum-Safe Cryptography.* 2022.
[63]. Kerry A McKay, Larry Bassham, Meltem Snmez Turan, and Nicky Mouha, *NISTIR 8114 report on lightweight cryptography.* National Institute of Standards and Technology (NIST), Gaithersburg, 2017.
[64]. NIST, *PUBLIC COMMENTS ON FIPS 197 - Advanced Encryption Standard (AES).* 2021: https://csrc.nist.gov/CSRC/media/Projects/crypto-publication-review-project/documents/initial-comments/fips-197-initial-public-comments-2021.pdf.
[65]. Joan Daemen and Vincent Rijmen, *AES proposal: Rijndael.* 1998.
[66]. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger, *Biclique cryptanalysis of the full AES*, in *Advances in Cryptology–ASIACRYPT 2011*. 2011, Springer. p. 344-371.
[67]. Denis Bonislavovich Fomin, *New classes of 8-bit permutations based on a butterfly structure.* Математические вопросы криптографии, 2019. **10**(2): p. 169-180.
[68]. Jiqiang Lu, Orr Dunkelman, Nathan Keller, and Jongsung Kim. *New impossible differential attacks on AES*. in *International Conference on Cryptology in India*. 2008. Springer.
[69]. Shengbao Wu, Mingsheng Wang, and Wenling Wu. *Recursive diffusion layers for (lightweight) block ciphers and hash functions*. in *Selected Areas in Cryptography*. 2013. Springer.
[70]. Hüseyin Demirci and Ali Aydın Selçuk. *A meet-in-the-middle attack on 8-round AES.* in *International Workshop on Fast Software Encryption*. 2008. Springer.
[71]. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. *Differential and linear cryptanalysis using mixed-integer linear programming*. in *International Conference on Information Security and Cryptology*. 2011. Springer.
[72]. David Wagner. *The boomerang attack.* in *International Workshop on Fast Software Encryption*. 1999. Springer.
[73]. Eli Biham, Orr Dunkelman, and Nathan Keller. *New results on boomerang and rectangle attacks*. in *International Workshop on Fast Software Encryption*. 2002. Springer.

[74]. Lars Knudsen and David Wagner. *Integral cryptanalysis*. in *Fast Software Encryption*. 2002. Springer.
[75]. Xuejia Lai, *Higher order derivatives and differential cryptanalysis*, in *Communications and Cryptography*. 1994, Springer. p. 227-233.
[76]. Carlos Cid, Sean Murphy, and Matthew JB Robshaw. *Small scale variants of the AES*. in *Fast Software Encryption*. 2005. Springer.
[77]. Gregor Leander and Axel Poschmann, *On the classication of 4 bit s-boxes*, in *Arithmetic of Finite Fields*. 2007, Springer. p. 159-176.
[78]. Alex Biryukov and Dmitry Khovratovich. *Related-key cryptanalysis of the full AES-192 and AES-256*. in *International Conference on the Theory and Application of Cryptology and Information Security*. 2009. Springer.
[79]. Yuechuan Wei, Ping Li, Bing Sun, and Chao Li. *Impossible differential cryptanalysis on Feistel ciphers with SP and SPS round functions*. in *International Conference on Applied Cryptography and Network Security*. 2010. Springer.
[80]. Andrey Bogdanov and Vincent Rijmen, *Linear hulls with correlation zero and linear cryptanalysis of block ciphers.* Designs, codes and cryptography, 2014. **70**(3): p. 369-383.
[81]. Andrey Bogdanov and Meiqin Wang. *Zero correlation linear cryptanalysis with reduced data complexity*. in *Fast Software Encryption*. 2012. Springer.
[82]. Andrey Bogdanov, Gregor Leander, Kaisa Nyberg, and Meiqin Wang. *Integral and multidimensional linear distinguishers with correlation zero*. in *International Conference on the Theory and Application of Cryptology and Information Security*. 2012. Springer.
[83]. Marc Kaplan, Gaëtan Leurent, Anthony Leverrier, and María Naya-Plasencia, *Quantum differential and linear cryptanalysis.* arXiv preprint arXiv:1510.05836, 2015.
[84]. Alex Biryukov, Dmitry Khovratovich, and Ivica Nikolić, *Distinguisher and related-key attack on the full AES-256*, in *Advances in Cryptology-CRYPTO 2009*. 2009, Springer. p. 231-249.
[85]. Denis Denisenko, *Quantum differential cryptanalysis.* Journal of Computer Virology and Hacking Techniques, 2022. **18**(1): p. 3-10.
[86]. Huiqin Xie and Li Yang, *Using Bernstein–Vazirani algorithm to attack block ciphers.* Designs, Codes and Cryptography, 2019. **87**(5): p. 1161-1182.
[87]. Huiqin Xie and Li Yang, *Quantum impossible differential and truncated differential cryptanalysis.* arXiv preprint arXiv:1712.06997, 2017.
[88]. Jian Guo, Jérémy Jean, Ivica Nikolić, and Yu Sasaki. *Meet-in-the-middle attacks on generic Feistel constructions*. in *International Conference on the Theory and Application of Cryptology and Information Security*. 2014. Springer.
[89]. Damian S Steiger, Thomas Häner, and Matthias Troyer, *ProjectQ: an open source software framework for quantum computing.* Quantum, 2018. **2**: p. 49.
[90]. National Institute of Standards and Technology NIST, *Submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016).* https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf, 2016.

[91]. Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. *Applying Grover's algorithm to AES: quantum resource estimates*. in *Post-Quantum Cryptography*. 2016. Springer.
[92]. National Institute of Standards and Technology NIST, *Call for additional digital signature schemes for the post-quantum cryptography standardization process (2022).* https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf, 2022.
[93]. Kyungbae Jang, Anubhab Baksi, Hyunji Kim, Gyeongju Song, Hwajeong Seo, and Anupam Chattopadhyay, *Quantum analysis of aes.* Cryptology ePrint Archive, 2022.
[94]. Ali Doganaksoy, Baris Ege, Onur Koçak, and Fatih Sulak, *Cryptographic Randomness Testing of Block Ciphers and Hash Functions.* IACR Cryptology ePrint Archive, 2010: p. 564.
[95]. Lawrence E Bassham III, Andrew L Rukhin, Juan Soto, James R Nechvatal, Miles E Smid, Elaine B Barker, Stefan D Leigh, Mark Levenson, Mark Vangel, and David L Banks, *SP 800-22 Rev. 1a.* A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, National Institute of Standards & Technology, Gaithersburg, MD, 2010.
[96]. Fatih Sulak, Ali Doğanaksoy, Barış Ege, and Onur Koçak. *Evaluation of randomness test results for short sequences*. in *International Conference on Sequences and Their Applications*. 2010. Springer.
[97]. J. Sevilla. *Assessing the impact of quantum cryptanalysis*. July 22, 2020; Available from: www.pqsecurity.com/white-paper-release-2-12-2020.
[98]. Taizo Shirai. *Differential, linear, boomerang and rectangle cryptanalysis of reduced-round Camellia*. in *Proceedings of the Third NESSIE Workshop, Munich, Germany*. 2002.
[99]. Pravin B Ghewari, J Patil, and A Chougule, *Efficient hardware design and implementation of AES cryptosystem.* International journal of engineering science and technology, 2010. **2**(3): p. 213-219.
[100]. Yuan-Quan Tan, Sheng-Qiang Li, and Hou-Jun Wang, *Analysis on data format of Mode 5 in western Mark XIIA.* Journal of University of Electronic Science and Technology of China, 2011. **40**(4): p. 532-536.
[101]. V. L. Nguyen T. Hoang, *An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm.* 2012 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, March 2012, 2012.
[102]. YJ Minal and MA Sayyad, *Implementation of AES on FPGA.* IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), 2014. **4**(5): p. 65-69.
[103]. Pritamkumar N Khose and Vrushali G Raut. *Implementation of AES algorithm on FPGA for low area consumption*. in *2015 International Conference on Pervasive Computing (ICPC)*. 2015. IEEE.
[104]. E Kavitha, *FPGA implementation of area optimized AES Algorithm for secure communication applications.* International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2016. **5**(4).
[105]. Alshaima Q Al-Khafaji, Mohammed Falih Al-Gailani, and Hikmat N Abdullah. *FPGA design and implementation of an AES algorithm based on iterative*

looping architecture. in *2019 IEEE 9th international conference on consumer electronics (ICCE-Berlin)*. 2019. IEEE.
[106]. Zhenzhen Bao, Jian Guo, San Ling, and Yu Sasaki, *PEIGEN–a Platform for Evaluation, Implementation, and Generation of S-boxes.* IACR Transactions on Symmetric Cryptology, 2019: p. 330-394.
[107]. Kishan Chand Gupta and Indranil Ghosh Ray, *On constructions of MDS matrices from companion matrices for lightweight cryptography*, in *Security Engineering and Intelligence Informatics*. 2013, Springer. p. 29-43.
[108]. ОС Зензин and МА Иванов, *Стардарт криптографической защиты-AES. Конечные поля*. 2002: КУДРИЦ-ОБРАЗ М.
[109]. Muhammad Reza Z'aba, *Analysis of linear relationships in block ciphers.* Luận án tiến sĩ của Queensland University of Technology, 2010.
[110]. Qing Zhou, Songfeng Lu, Zhigang Zhang, and Jie Sun, *Quantum differential cryptanalysis.* Quantum Information Processing, 2015. **14**(6): p. 2101-2109.