TOÁN HỌC ĐẰNG SAU MÃ QR

NGUYỄN HOÀNG VŨ[1]

Mã QR đã trở nên rất quen thuộc trong đời sống hàng ngày quanh ta, mang lại sự tiện lợi lớn cho các hoạt động giao dịch cũng như trao đổi thông tin. Trong bài này, chúng ta hãy cùng tìm hiểu về vai trò của toán học trong quá trình xây dựng loại mã này.

Hình 1. *Mã QR của Tạp chí Pi.*

1. Mã sửa lỗi và sự ra đời của mã Reed – Solomon

Một vấn đề không tránh khỏi khi thông tin được truyền tải từ nơi này đến nơi khác là việc nó có thể bị sai lệch trong quá trình truyền tin. Với các tín hiệu dạng nhị phân, một bit 0 có thể biến thành 1 và ngược lại. Xác suất của việc này phụ thuộc vào các đặc tính của đường truyền cũng như môi trường truyền dẫn.

Để đảm bảo thông tin có thể được kiểm chứng đúng/sai, ta cần ghi thông tin theo một loại mã có thể giúp ta khẳng định nó có bị lỗi hay không. Loại mã này gọi là mã phát hiện lỗi.

Một ví dụ về mã phát hiện lỗi là mã số của sách (ISBN) hoặc tạp chí (ISSN). Tạp chí Pi của chúng ta có mã ISSN là $2525-2437$. Trong 8 chữ số của mã, chữ số cuối cùng được dùng để phát hiện lỗi. Đầu tiên, ta lấy 7 chữ số đầu tiên, nhân mỗi một chữ số với số thứ tự tính từ bên phải (tức là $8, 7, 6, 5, \ldots 2$) rồi tính tổng:

$$2\times 8+5\times 7+2\times 6+5\times 5+2\times 4 +4\times 3+3\times 2=114.$$

Lấy tổng này chia cho 11 rồi lại lấy 11 trừ đi số dư sẽ được chữ số cuối cùng (nếu giá trị này là 10 thì ký tự cuối cùng là X): 114 chia 11 dư 4; $11-4=7$.

Nếu chữ số hoặc ký tự cuối không khớp với 7 chữ số đầu tiên, ta có thể khẳng định rằng ISSN đã bị nhập sai. Cách làm này không chỉ phát hiện lỗi do một vị trí bị thay đổi mà còn phát hiện được trường hợp hai vị trí cạnh nhau bị hoán vị (một lỗi thường thấy khi con người nhập dữ liệu) thông qua hệ số khác

[1] *Viện Sinh thái và Môi trường Đông Dương.*

nhau của các vị trí. Số 11 được chọn do nó là một số nguyên tố, không chia hết cho bất cứ hệ số nào trong phép nhân. Nếu ta chọn số 10, nó sẽ không phát hiện được lỗi nếu chữ số ở vị trí 5 bị thay đổi một lượng chẵn.

Trong nhiều trường hợp, chỉ phát hiện được lỗi là không đủ. Thay vì yêu cầu tín hiệu được gửi lại khi có lỗi, người ta muốn sử dụng một loại mã có thể giúp sửa lỗi khi lỗi được phát hiện. Muốn vậy, khi gửi tín hiệu, ta cần thêm một số thông tin khác để trợ giúp sửa lỗi.

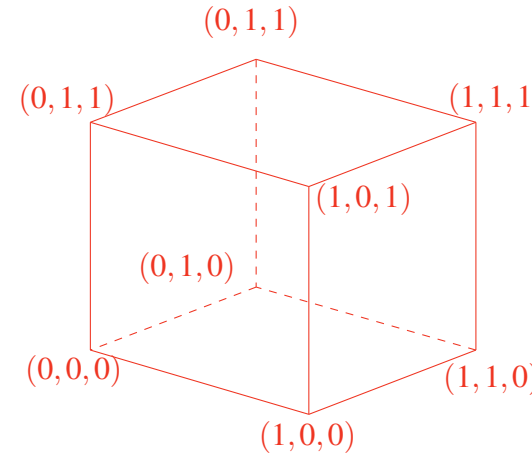
Một cách đơn giản nhất là gửi thông tin lặp. Thay vì gửi một bit có giá trị là 0 hoặc 1, ta có thể gửi nhiều bit có cùng giá trị. Chẳng hạn, nếu tín hiệu nhận được là 1101110111, với quy định là 10 bit liền nhau sẽ có cùng một giá trị, thì ta có thể khẳng định bit được gửi đi là 1. Cách làm đơn giản này tuy có thể giúp sửa lỗi nhưng nó rất lãng phí đường truyền.

Vậy cần phải truyền thêm bao nhiêu dữ liệu và dữ liệu sửa lỗi cần phải có dạng như thế nào? Đây là những vấn đề thiết yếu của ngành lý thuyết mã hóa.

Năm 1948, những nghiên cứu của nhà toán học Claude Shannon về entropy của thông tin cùng lượng thông tin cần truyền tối thiểu để sửa lỗi đã thúc đẩy nhiều nghiên cứu về các phương thức truyền thông tin có sửa lỗi.

Một loại mã sửa lỗi quan trọng ra đời vào những năm 1950 là mã Hamming (theo tên của nhà toán học R. W. Hamming). Ta hãy xét một trường hợp rất đơn giản với dữ liệu được truyền theo 3 bit một, với 8 tổ hợp: 000, 001, 010, 011, 100, 101, 110, 111. Nếu ta chọn cả 8 tổ hợp này để truyền tin thì khi có một bit bị lỗi, ta không thể tiến hành phát hiện lỗi. Một cách đơn giản là chọn các trạng thái mà khi bị lỗi, ta có thể biết được giá trị gốc của tín hiệu.

Ta hãy biểu diễn 8 tổ hợp trên 8 đỉnh của hình lập phương. Có thể thấy khi đi từ một đỉnh đến một đỉnh kề nó thì có đúng một bit bị thay đổi. Số cạnh trên đường đi từ một đỉnh đến một đỉnh khác cho ta biết số bit bị thay đổi giữa hai trạng thái. Đây được gọi là khoảng cách Hamming giữa chúng. Ví dụ khoảng cách Hamming giữa 000 và 101 là 2 còn khoảng cách giữa 000 và 111 là 3.



Hình 2. Các tổ hợp 3 bit nhị phân biểu diễn theo dạng hình lập phương. Các đỉnh kề nhau sẽ khác biệt một bit.

Giả sử trong môi trường truyền tin, tín hiệu 3 bit bị lỗi tối đa một bit. Để có thể xác định được trạng thái gốc từ trạng thái lỗi, ta chỉ có thể chọn các đỉnh có khoảng cách là 3. Do đó, chỉ có thể chọn 2 đỉnh là 000 và 111. Giả sử tín hiệu nhận được là 011, ta sẽ biết được tín hiệu gốc là 111 do khoảng cách Hamming giữa chúng là 1 còn khoảng cách Hamming giữa 011 và 000 là 2. Khi đó, 000 và 111 được gọi là các codeword (từ mã) cho việc truyền tín hiệu. Trong trường hợp này ta truyền đi tổng cộng 3 bit nhưng lượng thông tin chỉ có 2 trạng thái (ứng với 2^1 codeword) cho nên tốc độ truyền của ta là 1/3. Nếu chọn tập hợp các đỉnh có khoảng cách giữa chúng là 2 (ví dụ 4 đỉnh 000, 011, 101, 110) để làm codeword thì ta chỉ có thể phát hiện lỗi chứ không sửa lỗi (ví dụ 001 có khoảng cách đến 000 và 011 đều là 1).

Với mã Hamming, các quy tắc tính chẵn lẻ được tận dụng để thêm vào các thông tin cho phép sửa lỗi. Ta hãy xét một mã Hamming đơn giản với 7 bit $b_1, b_2, \ldots, b_7$ theo quy tắc

sau:

- $b_1 + b_3 + b_5 + b_7$ là chẵn
- $b_2 + b_3 + b_6 + b_7$ là chẵn
- $b_4 + b_5 + b_6 + b_7$ là chẵn

Có tổng cộng 16 codeword độ dài 7 bit thỏa mãn các quy tắc trên (Bảng 1). Đồng thời, các codeword này đều có khoảng cách Hamming giữa chúng là 3.

1	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0
3	1	0	0	1	1	0	0
4	0	1	1	1	1	0	0
5	0	1	0	1	0	1	0
6	1	0	1	1	0	1	0
7	1	1	0	0	1	1	0
8	0	0	1	0	1	1	0
9	1	1	0	1	0	0	1
10	0	0	1	1	0	0	1
11	0	1	0	0	1	0	1
12	1	0	1	0	1	0	1
13	1	0	0	0	0	1	1
14	0	1	1	0	0	1	1
15	0	0	0	1	1	1	1
16	1	1	1	1	1	1	1

Bảng 1. 16 codeword độ dài 7 bit.

Khi giải mã tín hiệu, các tín hiệu vi phạm những quy tắc chẵn lẻ sẽ bị coi là lỗi. Việc phát hiện lỗi nằm ở bit nào (giả sử chỉ có 1 bit bị lỗi) thì phức tạp hơn một chút. Tuy ta có thể dò xem codeword nào có khoảng cách ngắn nhất đến tín hiệu bị lỗi, việc này khá mất thời gian. Thay vì đó, người ta sử dụng dạng ma trận của các quy tắc được sử dụng để xây dựng mã (Bảng 2).

b_1	b_2	b_3	b_4	b_5	b_6	b_7
1	0	1	0	1	0	1
0	1	1	0	0	1	1
0	0	0	1	1	1	1

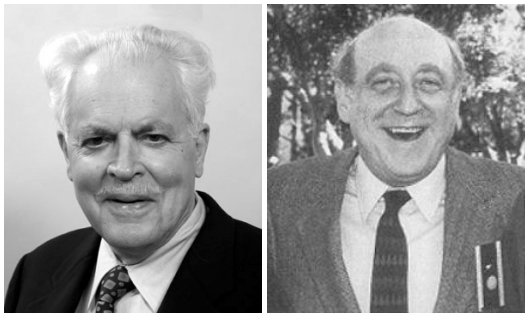
Bảng 2. Ma trận ứng với 3 quy tắc trong mã Hamming mà ta đang xét.

Ví dụ, với chuỗi tín hiệu lỗi 1101101, nó vi phạm quy tắc 1 và 3. Ta cần tìm cột trong ma trận mà khi bit ứng với cột này bị lật ngược lại thì chỉ có quy tắc 1 và 3 bị ảnh hưởng còn quy tắc 2 không thay đổi. Nói cách khác, ta cần tìm cột có dạng:

$$\begin{matrix} 1 \\ 0 \\ 1 \end{matrix}$$

tức là cột thứ 5 trong ma trận. Do đó, có thể kết luận bit thứ 5 bị lỗi và codeword gốc là 1101001.

Mã Hamming 7 bit trên có $2^4 = 16$ codeword, tức là khi truyền đi 7 bit, ta thu được lượng thông tin tương đương với 4 bit. Tốc độ truyền ở đây là $4/7$. Tùy theo nhu cầu, người ta có thể thiết kế các mã Hamming với khoảng cách giữa các codeword lớn hơn để có thể sửa lỗi cho nhiều hơn 1 bit.



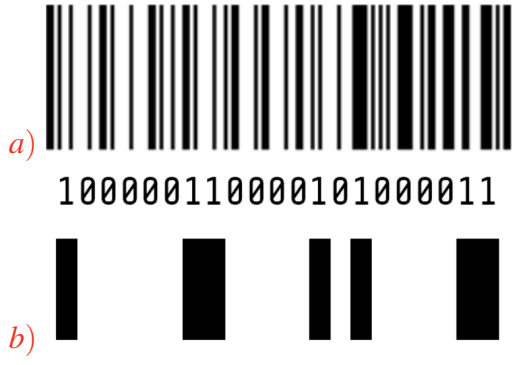
Trái: Irving Reed (1923 – 2012). Phải: Gustave Solomon (1930 – 1996).

Sau khi Hamming đưa ra mã sửa lỗi thực tiễn đầu tiên sử dụng các kỹ thuật đại số tuyến tính như trên, cuộc đua giữa các nhà toán học để tìm ra những cách mã hóa tốt hơn trở nên sôi động. Một dấu ấn quan trọng của ngành mã hóa là sự ra đời của mã Reed – Solomon, được hai nhà toán học Irving Reed và Gustave Solomon công bố năm 1960. Mã này sử dụng các cấu trúc đại số phức tạp hơn bao gồm trường Galois cùng các đa thức trên trường này (xem chi tiết ở phần phụ lục). Với nhiều lợi thế so với mã Hamming về khả năng

sửa lỗi cũng như tốc độ truyền, mã Reed – Solomon đã có nhiều ứng dụng rộng rãi trong các lĩnh vực của đời sống. Một trong số những ứng dụng đó chính là mã QR.

2. Mã QR

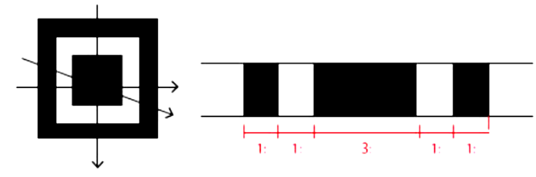
Mã QR (viết tắt của Quick Response) mà chúng ta quen thuộc được kỹ sư người Nhật, Masahiro Hara thiết kế năm 1994 khi làm tại công ty Denso Wave. Lúc đó, Masahiro được giao nhiệm vụ thiết kế một loại mã mới để thay thế mã dạng barcode (mã vạch) (Hình 3). Tuy mã barcode đã được dùng rất phổ biến trong cả sản xuất lẫn phân phối, nhưng vì nó ghi lại thông tin theo một chiều nên dung lượng thông tin của barcode (20 ký tự) đã không còn đáp ứng được nhu cầu thực tế với sự đa dạng hóa sản xuất đầu những năm 1990 ở Nhật. Để lưu trữ những thông tin dài, người ta thường phải sử dụng nhiều barcode cùng lúc và do đó phải tiến hành quét nhiều lần cho mỗi sản phẩm.



Hình 3. a) Mã barcode gồm cách vạch với độ dày và khoảng cách khác nhau. b) Thực chất của mã barcode là các cột màu trắng hoặc đen liên tiếp ứng với các bit 0 hoặc 1 để biểu diễn thông tin đã được chuyển từ dạng ký tự sang dạng bit.

Để có thể chứa được nhiều thông tin hơn, Masahiro quyết định sử dụng mã dạng hình vuông (hai chiều không gian thay vì một chiều như barcode). Vấn đề đầu tiên là làm thế nào để thiết bị có thể nhận diện được vùng mã khi quét. Masahiro đã nảy ra ý tưởng đưa thêm các hình hình học vào các góc để giúp định vị mã.

Tuy nhiên, việc chọn hình hình học này thế nào cũng là vấn đề khó bởi trong thực tế có thể có các hình hình học khác ở gần mã của ta làm phần mềm không thể nhận diện được mã một cách chính xác. Sau một thời gian nghiên cứu về tỷ lệ giữa các vùng trắng/đen trong các nội dung ảnh hoặc văn bản trên báo, tạp chí, tờ rơi, thùng carton, và nhiều tài liệu khác; nhóm làm việc của Masahiro quyết định sử dụng hình ô vuông như ta thấy ở ba góc của mã QR ngày nay. Hình ô vuông này có tỷ lệ các phần đen/trắng khi quét theo các góc khác nhau đều là 1 : 1 : 3 : 1 : 1. Theo kết quả thống kê thì tỷ lệ này ít xuất hiện nhất trên các vật liệu in thực tế nên mã QR sẽ không bị lẫn vào môi trường xung quanh và có thể được thiết bị quét nhận diện tự động nhanh chóng sau khi tiến hành quét.

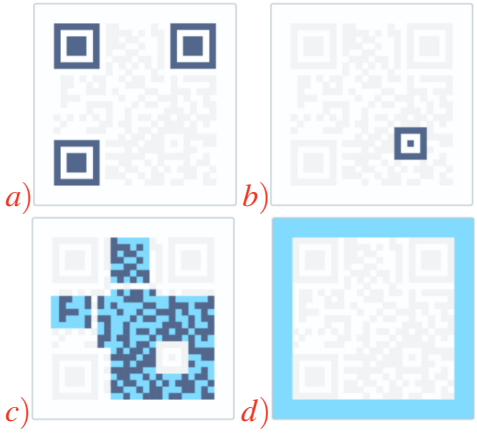


Hình 4. Ô vuông định vị cho mã QR và tỷ lệ đen/trắng khi quét.

Một vấn đề thực tế khác mà Masahiro muốn giải quyết với mã QR là tình trạng các vết bẩn làm sai lệch thông tin. Trong các môi trường như nhà xưởng, cửa hàng, ... mã đã in ra có thể dễ dàng bị dính các loại dầu, bụi đất, ... Để việc đọc mã có thể tiến hành thuận lợi trong các điều kiện này, ông sử dụng mã sửa lỗi Reed – Solomon để ghi thông tin thay vì chuyển trực tiếp từ dạng ký tự sang dạng bit như mã barcode. Do đó, thông tin bị sai lệch ở mức độ cho phép trong quá trình quét có thể được sửa lại thay vì phải quét lại hoặc làm sạch vùng chứa mã.

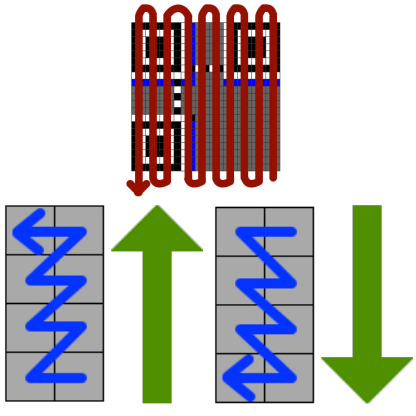
Từ khi được Masahiro thiết kế đến nay, mã QR đã trải qua nhiều phiên bản. Về cơ bản, một mã QR gồm các nội dung như trong Hình 4. Ngoài các ô vuông định vị và phần thông tin sử dụng mã Reed – Solomon, mã

QR còn gồm một số thành phần khác như ô vuông giống hàng, phần viền trắng (để tách biệt với xunh quanh) và các phần ghi một số thông tin khác về phiên bản mã và định dạng mã, ...



Hình 5. Một số thành phần của mã QR. a) Các ô định vị. b) Ô vuông giống hàng. c) Phần ghi dữ liệu. d) Phần viền trắng.

Khi mã hóa, các thông tin thực tế (chữ số, ký tự, ...) đã được chuyển thành dạng bit nhị phân theo quy tắc định trước (tùy theo phiên bản mã QR) sẽ được mã hóa thành mã Reed – Solomon dạng bit. Các bit được ghi lên trên hình vuông theo cách thức như trong Hình 6. Vị trí đen ứng với bit 1 còn vị trí trắng ứng với bit 0.



Hình 6. Thông tin sau khi được mã hóa theo mã Reed – Solomon sẽ được ghi theo các làn như hình vẽ (trên). Ứng với mỗi chiều trong làn, thông tin sẽ được ghi trong hai cột (dưới) với màu đen đại diện bit 1 còn màu trắng đại diện bit 0.

Khi thiết bị quét mã QR, phần mềm sẽ định vị mã dựa vào vị trí của các ô vuông đánh dấu và tiến hành quét phần dữ liệu. Dữ liệu sau khi quét được tiến hành giải mã theo thuật toán giải mã của mã Reed – Solomon. Do khả năng sửa lỗi của mã này nên nếu quá trình quét có một số bit bị nhận diện sai (đen thành trắng hoặc trắng thành đen), phần mềm vẫn có thể tiến hành sửa lỗi để cho ra thông tin ban đầu. Nếu số lượng lỗi vượt quá khả năng sửa lỗi của thuật toán, người dùng phải tiến hành quét lại.

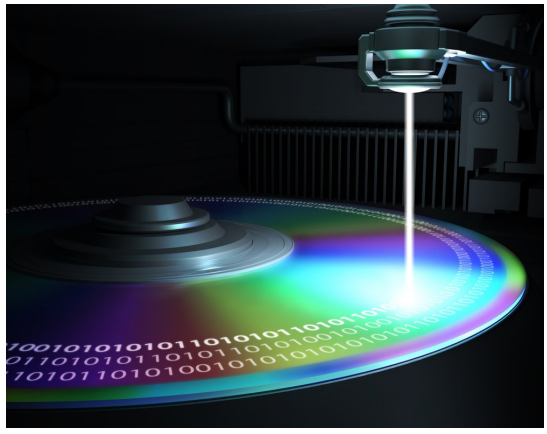
Hình 7. a) Masahiro và mã QR do ông phát minh. b) Masahiro có ý tưởng về các ô đen trắng cho mã QR từ trò chơi Go (cờ vây) mà ông hay chơi trong giờ nghỉ.

Ngày nay, với sự phổ biến của các thiết bị điện thoại thông minh, mã QR đã xuất hiện không chỉ ở trong các dây chuyền sản xuất mà còn được ứng dụng ở rất nhiều hoạt động của đời sống trên thế giới, từ thanh toán ngân hàng đến các giao dịch ở cửa hàng, siêu thị, bệnh viện, phương tiện công cộng, ... Nó cũng có vai trò quan trọng trong việc truy vết

COVID 19 ở nhiều nơi trên thế giới trong thời gian vừa qua; bản thân Masahiro cũng nói rằng ông cảm thấy rất hài lòng về sự đóng góp này của mã QR.

Hình 8. Việc sử dụng mã QR giúp tiến hành các giao dịch không cần tiền mặt chỉ với điện thoại di động.

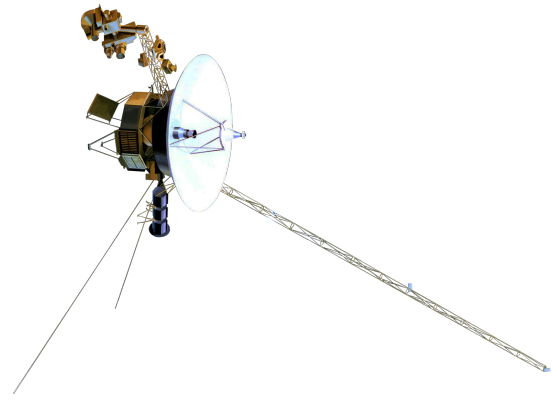
3. Một số ứng dụng khác của mã Reed – Solomon



Hình 9. Dữ liệu được mã hóa thành mã Reed - Solomon trước khi ghi trên bề mặt của các đĩa quang học.

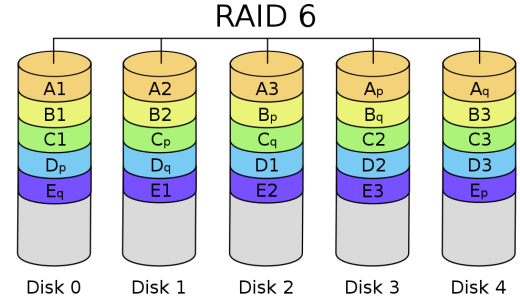
Mã Reed – Solomon đã từ lâu được sử dụng để mã hóa dữ liệu cho các loại đĩa lưu trữ thường thấy trong đời sống như CD, DVD, MiniDisc và Blu–ray. Dữ liệu nhị phân sau khi chuyển thành mã Reed – Solomon sẽ được ghi lên bề mặt đĩa tạo thành các rãnh lồi lõm. Khi đầu đọc đọc lại các bit này, chúng sẽ được giải mã theo thuật toán giải mã tương ứng để cho ra dữ liệu ban đầu. Do khả năng khôi phục lỗi của mã Reed – Solomon, ngay cả khi có lỗi lúc đọc (thường do bề mặt đĩa bị bụi, bẩn, ...), hệ thống vẫn có thể khôi phục dữ liệu gốc.

Mã Reed – Solomon cũng được sử dụng để mã hóa dữ liệu cho việc truyền tin của ngành viễn thông, theo dây cáp lẫn vô tuyến. Đặc biệt, mã này được sử dụng trên các thiết bị thăm dò không gian của NASA như tàu thăm dò Voyager 1 và nhiều tàu thăm dò sau đó.



Hình 10. Tàu thăm dò Voyager 1 của NASA.

Mã Reed – Solomon còn xuất hiện ở các hệ thống lưu trữ trong trung tâm dữ liệu. Trong kết nối ổ cứng theo dạng RAID 6, các mã Reed – Solomon để sửa lỗi cho dữ liệu của mỗi ổ cứng sẽ được hệ thống phân bố ở các ổ khác. Nếu một ổ cứng bị hỏng, dữ liệu có thể được khôi phục từ dữ liệu cùng các mã Reed – Solomon ở trong các ổ cứng còn lại.



Hình 11. Lưu trữ dữ liệu trên 5 ổ cứng theo dạng RAID 6. Dữ liệu cũng như thông tin sửa lỗi được chia ra để lưu trữ ở các ổ cứng khác nhau để cho phép phục hồi trong trường hợp một trong các ổ cứng này bị hỏng.

4. Lời kết

Sự xuất hiện của trường Galois dưới dạng cơ sở lý thuyết của mã Reed – Solomon cho thấy ngay cả những khái niệm trừu tượng và phức tạp trong toán học cũng có thể đóng vai trò quan trọng trong ứng dụng thực tiễn nếu ta tiến hành mô hình hóa bài toán một cách phù hợp. Mã Reed – Solomon cũng như các ứng dụng đa dạng của nó sẽ không xuất hiện nếu không có những nghiên cứu toán học thuần túy từ hơn một thế kỷ trước đó của Galois. Ngày nay, lĩnh vực lý thuyết mã hóa vẫn còn rất nhiều vấn đề toán học thú vị khác với nhiều cơ hội cho những ai muốn tìm tòi khám phá.

Phụ lục. Trường Galois và mã Reed – Solomon

Trong đại số, trường là một tập hợp mà trên đó có thể định nghĩa phép cộng và phép nhân với các tính chất giao hoán và kết hợp. Đồng thời, trong trường phải tồn tại phần tử 0 và 1 sao cho với phần tử a bất kì của trường thì $a+0=a$; $a\cdot 1=1$. Mỗi phần tử của trường đều phải có phần tử đối (với phép cộng) và phần tử nghịch đảo (với phép nhân, trừ trường hợp nghịch đảo của 0) cũng nằm trong trường này.

Ví dụ, tập hợp các số nguyên $\mathbb{Z}$ không phải là một trường vì nghịch đảo của một số nguyên có thể không phải là một số nguyên. Trong khi đó, tập hợp các số hữu tỷ $\mathbb{Q}$ hay tập hợp các số thực $\mathbb{R}$ đều là một trường.

Trường Galois (còn gọi là trường hữu hạn), được đặt tên theo nhà toán học Evariste Galois, là một trường có số phần tử là hữu hạn. Trường Galois $GF(q)$ (có q phần tử) luôn chứa một phần tử α sao cho cả các phần tử khác 0 của trường đều là các lũy thừa của α: $\{1,\alpha,\alpha^2,\ldots,\alpha^{q-2}\}$. Khi đó ta cũng có $\alpha^{q-1}=1$.

Thông qua các đa thức, người ta có thể liên hệ trực tiếp trường Galois $GF(2^n)$ với tập hợp các số nhị phân có n bit. Đây cũng là cách mã Reed – Solomon cùng nhiều mã sửa lỗi khác được xây dựng.

Với $n=1$, ta có trường Galois $GF(2)$ gồm hai phần tử $\{0,1\}$ cùng phép cộng và phép nhân như trong hệ cơ số nhị phân nhưng phép cộng ở đây không có nhớ: $1+1=0$.

Với $n>1$, ta hãy thử xét trường hợp $n=3$. Tập hợp các số nhị phân 3 bit sẽ gồm $\{000,001,010,100,011,110,101,111\}$. Với mỗi số nhị phân này, ta có một đa thức tương ứng trong đó các bit được dùng làm hệ số của các đơn thức trong đa thức. Ví dụ 111 ứng với x^2+x+1; 011 ứng với $x+1$, ...

Ta coi các hệ số của các đa thức trên là các phần tử của $GF(2)$. Do đó, phép cộng đa thức sẽ không có nhớ, chẳng hạn:

$$x^2+x^2=(1+1)x^2=0,$$
$$(x^2+x)+x=x^2+(1+1)x=x^2+0=x^2.$$

Để thu được trường hữu hạn, thay vì sử dụng phép nhân đa thức thông thường, người ta sử dụng phép nhân modulo một đa thức đặc trưng (tức là tiến hành nhân thông thường sau đó chia cho đa thức đặc trưng để lấy số dư). Với $GF(2^3)$, một đa thức đặc trưng của nó là x^3+x^2+1. Ta có thể biểu diễn các đa thức khác theo số dư khi chia cho đa thức này:

$$x^3\equiv x+1,$$
$$x^4\equiv x(x^3)\equiv x(x+1)\equiv x^2+x,$$
$$x^5\equiv x(x^4)\equiv x(x^2+x)\equiv x^3+x^2\equiv x^2+x+1,$$
$$x^6\equiv x(x^5)\equiv x(x^2+x+1)\equiv x^3+x^2+x$$
$$\equiv (x+1)+x^2+x\equiv x^2+1,$$
$$x^7\equiv x(x^6)\equiv x^3+x\equiv x+(x+1)\equiv 1\equiv x^0.$$

Do tính tuần hoàn của các biểu diễn này nên kết quả phép nhân luôn cho ta một đa thức trong 8 đa thức ban đầu và ta thu được một trường Galois $GF(2^3)$. Người ta chứng minh được rằng cho trường hợp $n>1$ bất kỳ, luôn có thể tìm được đa thức đặc trưng

để xây dựng trường Galois cùng phần tử α tương ứng của trường này (chú ý rằng phép toán lũy thừa trên trường cũng tuân theo quy tắc nhân modulo trên).

Trong mã Reed – Solomon, việc mã hóa được tiến hành cho từng gói gồm k phần tử của $GF(2^n)$. Nói cách khác, quá trình mã hóa được tiến hành cùng lúc cho k số nhị phân, mỗi số n bit. Ký hiệu các phần tử của gói là $m_0, m_1, \ldots, m_{k-1}$. Ta xây dựng được một đa thức:

$$P(x) = m_0 + m_1 x + \cdots + m_{k-1}x^{k-1}.$$

Chú ý rằng đa thức này khác với những đa thức đã nói ở trên. Những đa thức ở phía trên có hệ số là 0 hoặc 1 (phần tử của $GF(2)$) còn đa thức $P(x)$ của ta có các hệ số m_i là các phần tử của $GF(2^n)$. Một codeword được tạo ra bằng cách tính các giá trị của $P(x)$ cho từng phần tử của trường Galois $GF(2^n)$:

$$\begin{aligned} c &= (c_0, c_1, c_2, \ldots, c_{q-1}) \\ &= \left[P(0), P(\alpha), P(\alpha^2), \ldots, P(\alpha^{q-1})\right], q = 2^n, \end{aligned}$$

Do mỗi phần tử m_i có thể nhận một trong $q = 2^n$ giá trị nên có tổng cộng q^k codeword trong mã Reed – Solomon. Ứng với mỗi codeword, ta có một hệ phương trình:

$$\begin{aligned} P(0) &= m_0, \\ P(\alpha) &= m_0 + m_1\alpha + m_2\alpha^2 + \cdots \\ &\quad + m_{k-1}\alpha^{k-1}, \\ P(\alpha^2) &= m_0 + m_1\alpha^2 + m_2\alpha^4 + \cdots \\ &\quad + m_{k-1}\alpha^{2(k-1)}, \\ &\cdots \\ P(\alpha^{q-1}) &= m_0 + m_1\alpha^{q-1} + m_2\alpha^{2(q-1)} \\ &\quad + \cdots + m_{k-1}\alpha^{(k-1)(q-1)}. \end{aligned}$$

Chọn bất kỳ k trong số q phương trình trên, ta được một hệ phương trình mà từ đó có thể giải để tìm các m_i.

Trong trường hợp đường truyền bị lỗi, giả sử có t trong số q phương trình bị lỗi (giá trị $P(\alpha^i)$ tương ứng bị lỗi). Khi đó, nếu ta thử tất cả các tổ hợp có thể gồm k phương trình từ q phương trình, sẽ có $\binom{t+k-1}{k}$ hệ phương trình cho kết quả m_i khác với các hệ phương trình còn lại. Do đó, việc sửa lỗi có thể được tiến hành bằng cách lấy kết quả chiếm đa số, với điều kiện là $\binom{t+k-1}{k} < \binom{q-t}{k}$. Số lỗi lớn nhất có thể sửa là số nguyên t nhỏ hơn hoặc bằng $\frac{1}{2}(q-k+1)$.

Việc thử các tổ hợp hệ phương trình khác nhau theo đề xuất ban đầu của Reed và Solomon là tương đối phức tạp về mặt cài đặt trong thực tế nên sau đó người ta đã đưa ra hai hướng tiếp cận khác để tạo mã Reed – Solomon, một sử dụng đa thức sinh và một sử dụng biến đổi Fourier trên trường Galois (Wicker, 1994).

Tài liệu tham khảo

[1] Aktaş, C. (2017). *The evolution and emergence of QR codes*. Cambridge Scholars Publishing.

[2] Reed, I. S., & Solomon, G. (1960). *Polynomial Codes Over Certain Finite Fields*. Journal of the Society for Industrial and Applied Mathematics, 8(2), 300 – 304.

[3] Wicker, S. B., & Bhargava, V. K. (1994). *Reed–Solomon codes and their applications*. IEEE.