Chương 2

CÁC MÃ TUYẾN TÍNH

2.1 CÁC KHÁI NIỆM CƠ BẢN

2.1.1. Các kỹ thuật mã hóa

Mã hóa thông tin nhằm giải quyết tình trạng lỗi trong quá trình lưu trữ, truyền tải thông tin trong môi trường truyền tin. Có thể phân biệt hai kỹ thuật mã hóa chủ yếu:

- *Mã hóa dùng nguồn*

  Ý tưởng chính của phương pháp này là nén dữ liệu từ chính nguồn của nó trước khi truyền đi, giúp cho việc truyền tin có hiệu quả hơn.

  Chẳng hạn, ta thường dùng công cụ "zip" nén dữ liệu để giảm dung lượng dữ liệu phải truyền trên internet, hoặc khi sao chép.

- *Mã hóa kênh truyền*

  Kỹ thuật này giúp cho việc truyền thông tin chính xác hơn trong môi trường nhiễu loạn của kênh truyền thông. Trong chuyên đề này, chúng ta chủ yếu nghiên cứu các vấn đề liên quan đến kỹ thuật mã hóa kênh truyền.

Vấn đề mã hóa trên kênh truyền dựa trên ý tưởng thêm các ký hiệu mới (thường là các bit với mã nhị phân) vào trong dữ liệu được truyền, để có thể xây dựng các loại mã nhằm mục đích có thể phát hiện lỗi và sửa sai các lỗi. Đó là nội dung chủ yếu của lý thuyết mã hóa.

Có 2 vấn đề chính đặt ra với lý thuyết mã hóa thông tin:

1. Xây dựng các mã sao cho có thể khắc phục tối đa các lỗi, trong khi chỉ sử dụng tối thiểu các ký hiệu bổ xung.

2. Xây dựng các mã với thủ tục mã hóa và giải mã hiệu quả.

Trong chương này, chúng ta sử dụng các khái niệm và kết quả của Toán học đã trình bày ở chương trước để xây dựng và nghiên cứu các mã thông dụng dùng để phát hiện sai và các mã sửa sai cùng ứng dụng của chúng. Trong khi trình bày, một số khái niệm toán học phức tạp có thể được đơn giản hóa cho dễ hiểu.

2.1.2. Các đặc trưng của mã

Như ta đã biết, các thông điệp có thể được mã hóa thành các xâu ký tự trên bảng ký hiệu mã, gọi là các từ mã. Một tập hợp các từ mã được gọi là một mã. Một số đặc trưng của mã được định nghĩa như sau:

Định nghĩa 2.1.

Cho bảng ký hiệu mã $A = \{a_1, a_2, ..., a_q\}$.

- *Một mã C với độ dài n trên A là một tập con của* A^n*, tức là tập các vector trong* A^n.

- *Một phần tử $w \in C$ gọi là một từ mã, có dạng $w = x_1x_2.... x_n$, trong đó $x_i \in$ A,*
- *Số phần tử của C, ký hiệu | C |, gọi là kích thước của mã.*
- *Nếu bảng ký hiệu mã A gồm q ký hiệu, các mã trên A gọi là mã q-phân (q-ary codes), đặc biệt với q = 2, ta có các mã nhị phân (binary codes).*

Như vậy, một mã C với độ dài n là tập các từ mã, là các xâu gồm n ký hiệu trong A. Số từ mã trong mã C gọi là kich thước của mã (*side of code*). Một mã với độ dài *n* và kích thước M có thể được ký hiệu là một *(n, M)-code*, hay C(n, M)

Thí dụ 2.1

Với A = {0, 1}, một mã có độ dài *n* = 4 là một tập con của $A^4 = \{0, 1\}^4 = \{0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111\}$.

Chẳng hạn, C = {0001, 1000, 0110} là một mã nhị phân (4, 3)-code, rõ ràng $C \subset A^4 = \{0 , 1\}^4$

Các khái niệm về trọng số và khoảng cách liên quan đến mã C được đưa ra như sau:

Định nghĩa 2.2.

1.*Cho các từ mã trong C: $x = x_1 x_2.... x_n$ và $y = y_1 y_2.... y_n$ trên bảng ký hiệu mã A, ta ký hiệu* $d(x_i, y_i) = \begin{cases} 1 & if\ x_i \neq y_i \\ 0 & if\ x_i = y_i \end{cases}$, khi đó khoảng cách Hamming giữa 2 từ mã x , y ∈ C được *ký hiệu và xác định như sau:*

$$d(x, y) = \sum_{i=1}^{n} d(x_i, y_i) \qquad (2.1)$$

2. *Trọng số Hamming (weight) của từ mã x là số ký hiệu khác không trong x, ký hiệu là w(x).*

Như vậy khoảng cách Hamming giữa hai từ mã $\boldsymbol{x} = x_1 x_2.... x_n$ và $\boldsymbol{y} = y_1 y_2.... y_n$ là tổng số các vị trí tương ứng mà chúng khác nhau, tức là số các i ($i = 1,2,...,n$) sao cho $x_i \neq y_i$.

Thí dụ 2.2

1. d(10100, 10001) = 2 ; d(011010,101101) = 5 ; d(01201, 21011) = 3

2. w(10101) = 3 ; w(01020) = 2.

Khoảng cách Hamming thỏa mãn tất cả các tính chất của hàm khoảng cách (metric), như được chứng tỏ trong định lý sau:

Định lý 2.1

Với mọi từ mã x, y, z có độ dài n trên bảng ký hiệu mã A, khoảng cách Hamming có các tính chất sau:

1. $0 \leq d(x,y) \leq n$, $\forall x,y$.
2. $d(x,y) = 0$ *khi và chỉ khi* $x = y$.
3. $d(x,y) = d(y,x)$ *với* $\forall x,y$.
4. $d(x,y) \leq d(x,z) + d(z,y)$ *với* $\forall x,y,z$. *(bất đẳng thức tam giác)*

Chứng minh:

- Các tính chất 1,2,3 được suy ra ngay lập tức từ định nghĩa của khoảng cách Hamming.
- Để chứng minh tính chất 4 ta sử dụng nhận xét: *d(x,y)* là số các thay đổi của các ký tự trong xâu x cần thiết để biến đổi xâu *x* thành xâu *y*. Nhận thấy rằng, đối với xâu *z* có độ dài *n*, số các thay đổi các ký tự cần thiết để biến đổi xâu *x* thành xâu *y* không vượt quá số các thay đổi các ký tự để biến đổi xâu *x* thành xâu *z* và sau đó biến đổi *z* thành *y*.

Bổ đề 2.1

Nếu x và y là các xâu nhị phân cùng độ dài, thì d(x, y) = w(x + y).

Sinh viên tư chứng minh bổ đề này xem như bài tập.

Định nghĩa 2.3

Khoảng cách Hamming cực tiểu của mã C là khoảng cách Hamming nhỏ nhất giữa hai từ mã khác nhau bất kỳ trong C, được ký hiệu và xác định như sau:

$$d(C) = min\{d(x,y) | \forall x, y \in C, x \neq y\} \quad (2.2)$$

Như vậy, mỗi mã C có ba đặc tính chủ yếu là:

1. Chiều dài của các từ mã trong C
2. Tổng số các từ mã trong C.
3. Khoảng cách Hamming cực tiểu của mã C

Ta ký hiệu (*n, M, d*)-code là mã có độ dài *n,* chứa M từ mã và có khoảng cách cực tiểu *d = d(C)*

Chú ý 2.1: *Mã khối là một tập hợp bao gồm nhiều từ mã để mã hóa dữ liệu theo từng khối.*

Khi cần truyền nhiều dữ liệu bằng mã khối, người gửi chia dữ liệu thành nhiều phần nhỏ. Mỗi phần nhỏ được gọi là một thông điệp và mã hóa mỗi thông điệp thành một mã tự, còn được gọi là một khối trong mã hóa khối. Người gửi sẽ gửi tất cả các khối cho người nhận, sau đó người nhận sử dụng phục hồi lại thông điệp ban đầu từ các khối nhận dược có thể có lỗi. Bằng việc đồng nhất mỗi từ mã với 1 khối trong mã khối, ta có thể không cần phân biệt mã hay mã khối.

2.2 CÁC MÃ TUYẾN TÍNH

Chú ý mở đầu:

Các mã tuyến tính được xây dựng dựa trên các kết quả của Đại số tuyến tính. Sinh viên cần nắm vững các khái niệm về không gian vector, không gian con, sự độc lập tuyến tính, các vector trực giao, các cấu trúc đại số, đặc biệt là cấu trúc trường…đã được giới thiệu ở hầu hết các giáo trình Toán cao cấp (Giải tích và Đại số tuyến tính).

Các khái niệm về mã tuyến tính được trình bày trên trường *Galois*, là một trường F có số phần tử hữu hạn. Ta ký hiệu trường này là F_q (hay GF(q)) nếu | F | = q.

Để tiện theo dõi bài giảng, ta nhắc lại một số khái niệm về trường Galois :

- Với q là số nguyên tố, trường Galois GF(q) là tập ký hiệu F = {0, 1, 2, ..., q – 1}, trên đó có hai phép toán cộng ⊕ và nhân ⊗ modulo q.

- Đặc biệt khi F là bảng mã nhị phân {0 , 1} với phép toán cộng bit và nhân bit (theo modulo 2) đã biết thì trường Galois GF(2), thường ký hiệu ngắn gọn là F_2 :

  Trường F_2 là tập F = {0, 1}, trên đó có hai phép toán 'cộng bit' và 'nhân bit' .

Để cho thuận tiện, nếu không gây nhầm lẫn, trong các phần sau có thể sử dụng ký hiệu '+' và '.' thông thường thay cho các ký hiệu ⊕ bit và ⊗ bit, và được hiểu tùy theo ngữ cảnh. Trong giáo trình này, ta chỉ xét các mã nhị phân, vì vậy hầu hết các khái niệm và kết quả chỉ trình bày cho các mã nhị phân trên trường F_2. Các khái niệm và kết quả cho các trường GF(q) được mở rộng một cách tương tự.

2.2.1 Mã tuyến tính

Mã tuyến tính là một lớp mã được dùng rất phổ biến trong việc chống nhiễu (phát hiện sai và sửa sai). Mã tuyến tính C có mục đích mã hóa các khối tin *k* bit thành các từ mã *n* bit trước khi truyền đi, nói cách khác, trong *n* bit của từ mã truyền đi có *k* bit chứa thông tin, còn *n – k* bit là thêm vào (mã hóa) để chống nhiễu.

Ta nhắc lại vài khái niệm liên quan đến mã tuyến tính:

Tập tất cả các từ mã độ dài n trên trường GF(q) làm thành một không gian vector n chiều, ký hiệu là F_q^n , khi đó mỗi vector của không gian này là một từ mã q- phân độ dài *n*.

Định nghĩa 2.5

Cho C là một tập con cuả không gian các từ mã độ dài n trên trường F_q^n, C là một mã tuyến tính nếu và chỉ nếu các từ mã của C làm thành một không gian con k chiều của không gian n chiều F_q^n, tức là dim(C) = k, với $k \leq n$.

Hai tham số quan trọng nhất của một mã tuyến tính q-phân là độ dài từ mã n và số chiều của mã.

- Mã tuyến tính C với độ dài từ mã *n* và *dim(C) = k* ký hiệu là *C[n, k]-code (hay C[n, k])*
- Nếu *dim(C) = k* thì các từ mã trong C có *k* bit chứa thông tin và *n – k* bit thêm vào để chống nhiễu.
- Do F_q^n luôn là không gian con của chính nó, do đó tập mã C = F_q^n gồm p^n từ mã độ dài *n* cũng là một mã tuyến tính.

Áp dụng kết quả về không gian con của đại số tuyến tính, ta có thể xác định mã một mã tuyến tính nhờ bổ đề sau:

Bổ đề 2.2

Mã C được gọi là mã tuyến tính trên trường GF(q) nếu nó thỏa mãn 2 điều kiện sau:

(1) $x + y \in C, \ \forall x, y \in C$

(2) $a \,.\, x \in C, \ \forall x \in C, \ \forall a \in GF(q)$.

Chú ý rằng, trên trường GF(2) = {0, 1}, từ điều kiện *(2)* trong định nghĩa 2.5 ta thấy : Do mọi bit trong mã nhị phân nhân với 0 đều bằng 0, nhân với 1 đều cho chính bit đó, như vậy điều kiện *(2)* trong định nghĩa 2.5 luôn thỏa mãn, và ta chỉ cần kiểm tra điều kiện *(1)*.

Vậy mã nhị phân C là mã tuyến tính khi và chỉ khi tổng của 2 từ mã bất kỳ trong C cũng là một từ mã thuộc C.

Thí dụ 2.3.

1. Xét mã nhị phân C = {000, 001, 100, 101}.

Khi đó, thực hiện các phép tính + .bit đối với cặp từ mã 000 và 001, ta có: 000 + 001 = 001 ∈ C.

Bằng cách tương tự, thực hiện các phép tính + .bit đối với từng cặp các từ mã khác của C, ta dễ dàng suy ra tổng các cặp từ mã (thuộc C) sẽ thuộc C. Vậy C là một mã tuyến tính.

2. Xét mã B = {000, 100, 001, 111}. Mã này không phải là mã tuyến tính vì khi thực hiện các phép tính +.bit đối với cặp từ mã 100 và 001, ta có: 100 + 001 = 101 ∉ B.

Với các mã q-ary codes, q ≠ 2 (không phải mã nhị phân), thì C là mã tuyến tính khi và chỉ khi mỗi từ mã của C là tổ hợp tuyến tính của các từ mã còn lại trong C.

Thí dụ 2.4. Trên trường F_3, cho mã C = {0000, 1011, 0112, 2022, 0221, 1120, 2210, 1202, 2101}. Chứng tỏ rằng C là một mã tuyến tính.

Chứng minh: Ta cần chỉ ra rằng mỗi từ mã trong C là tổ hợp tuyến tính của 2 từ mã nào đó trong C, hay cũng vậy tổ hợp tuyến tính bất kỳ 2 từ mã cũng là một từ mã thuộc C (chú ý ràng tổ hợp tuyến tinh chỉ với các hệ số 0, 1, 2 ∈ F_3). Chẳng hạn: 1011 + 0112 = 1120 ∈ C, 1011 + 2*0112 = 1202 ∈ C, 2*1011 + 0112 = 2101 ∈ C, …etc.

Khái niệm về mã đối ngẫu:

Vì mỗi bộ mã tuyến tính C là một không gian con trong A^n, nên theo lý thuyết về không gian vector, sẽ tồn tại không gian con trực giao của không gian con C, đó là tập vector gồm mọi vector trong A^n mà trực giao với mọi vector của C. Không gian con trực giao của C gọi là mã đối ngẫu của C. Để cho đơn giản. trong giáo trình này chúng ta chỉ xét các mã đối ngẫu của các mã tuyến tính nhị phân, trên trường F_2.

Định nghĩa 2.6.

> *Cho C là một mã tuyến độ dài n tính trên trường F_2 , mã đối ngẫu của C, ký hiệu $C^\perp$, là không gian con trực giao của C trong không gian n chiều F_2^n.*

Nói cách khác, mã đối ngẫu $C^\perp$ gồm các từ mã *n* bit trực giao với các từ mã của C.

$$C^\perp = \{ v \in \{0, 1\}^n \mid v.w = 0, \ \forall w \in C \},$$

Trong đó $v = (v_1\ v_2\ \dots\ v_n)$; $w = (x_1\ x_2\ \dots\ x_n)$, các $x_i, v_i \in \{0, 1\}$, ký hiệu *v.w* là tích vô hướng xác định bởi:

$$v.w = \sum_{i=1}^{n} x_i v_i \text{ , tổng này thực hiện với các phép +.bit và x.bit.}$$

Thí dụ 2.5

1. Trên trường F_2, ta có tích vô hướng của các từ mã: (1001).(1101) = 1.1+0.1+0.0+1.1 = 1+0+0+1 = 0. Vậy các từ mã 1001 và 1101 là trực giao nhau.

- Tương tự: ta có (1111).(1110) = 1. Vậy các từ mã 1111 và 1110 là không trực giao.

2 Trên trường F_3, chứng minh rằng:

a/. Các từ mã 12101 và 21010 là không trực giao (gợi ý: tính tích vô hướng của 2 từ mã)

b/. Các từ mã 12101 và 21111 là trực giao. (gợi ý: tính tích vô hướng của 2 từ mã).

Từ các kết quả của đại số tuyến tính, có thể chứng minh định lý sau:

Định lý 2.2

1. Nếu C là một mã tuyến tính trên trường GF(q) thì | C | = $q^{dim(C)}$.

2. Nếu C là một mã tuyến tính độ dài n trên trường GF(q) thì $C^{\perp}$ cũng là một mã tuyến tính trên trường này và ta có:

$$dim(C) + dim\ (C^{\perp}) = n.$$

3. $(C^{\perp})^{\perp} = C$.

4. Mọi mã tuyến tính C đều chứa từ mã 00 ... 0.

Mã tuyến tính C độ dài *n*, có *dim(C) = k* được ký hiệu là *[n , k]-code*, hoặc C[n , k]. Để chỉ rõ mã có khoảng cách Hamming cực tiểu là d, có thể dùng ký hiệu [n , k , d]-code, hoặc C[n , k , d]

Thí dụ 2.6. Cho mã nhị phân C = {000, 001, 100, 101}, trong thí dụ 2.3 đã được chứng minh đây là mã tuyến tính . Khi đó ta có:

1. Từ C ta thấy chỉ có tối đa 2 vector mã độc lập tuyến tính, vậy dim (C) = 2.

   Hoặc có thể tính dim(C) từ công thức | C | = $2^{dim(C)}$, ta có | C | = 4, nên $2^{dim(C)} = 4$.
   Vậy dim(C) = 2.

2. Do độ dài từ mã *n* = 3, số chiều dim(C) = 2, vậy C là một [3 , 2]-code, hoặc một [3 , 2 , 1]-code, nếu chú ý đến khoảng cách Hamming cực tiểu *d = 1* của mã C.

3. Tìm mã đối ngẫu của C.

Giải: Trong không gian vector 3 chiều V = $\{0 , 1\}^3$ = {000, 001, 010, 011, 100, 101, 110, 111}, tìm tập các vector trực giao với mọi vector của C, đó là mã đối ngẫu của C. Ta được: $C^{\perp}$ = {000, 010}, chỉ có 1 vector độc lập tuyến tính, vậy dim($C^{\perp}$) = 1.

Nhận xét: Từ các kết quả của đại số tuyến tính, ta có kết quả sau:

1. Tập các vector mã nhị phân cùng độ dài là độc lập tuyến tính khi và chỉ khi tổng của chúng là một vector mã khác không (có chứa ít nhất một bit '1')
2. Tập các vector mã nhị phân cùng độ dài là phụ thuộc tuyến tính khi và chỉ khi tổng của chúng là một vector mã bằng không (chứa toàn bit '0')
3. Mọi tập vector mã có chứa vector không đều phụ thuộc tuyến tính
4. Hệ gồm chỉ một vector mã khác không là một hệ độc lập tuyến tính.

Sinh viên tự chứng minh các kết quả trên xem như bài tập.

2.2.2 Ma trận sinh của mã tuyến tính

Mã tuyến tính C[*n* , *k*] là một không gian con *k* chiều của không gian *n* chiều các từ mã *n* thành phần trên trường F_2 , do đó tồn tại *k* từ mã độc lập tuyến tính, chẳng hạn $\{g_0, g_1,\ldots, g_{k-1}\}$ làm thành một cơ sở của C, sao cho mỗi từ mã trong C là tổ hợp tuyến tính của *k* từ mã này,. Tức là:

$$\forall\ w \in C,\ w = a_0g_0 + a_1g_1 + \ldots + a_{k-1}g_{k-1}\ ,\ \text{với các hệ số } a_i \text{ thuộc } \{0, 1\} \qquad (2.3)$$

Xếp *k* từ mã này thành *k* hàng, tạo nên một ma trận cấp *k x n*, gọi là ma trận sinh của mã C.

Định nghĩa 2.8.

Cho C là một mã tuyến tính [n , k]-code trên trường F_2,

1. *Ma trận G = $[g_{ij}]$ cấp k x n mà các hàng của nó hình thành một cơ sở của mã tuyến tính C, thì G được gọi là ma trận sinh của mã C.*
2. *Ma trận sinh H cấp (n-k)× n của mã đối ngẫu $C^{\perp}$ gọi là ma trận kiểm tra của mã C*

Chú ý:

1. Nếu C là một mã tuyến tính [*n , k*]-code thì có thể có nhiều ma trận sinh khác nhau, mọi ma trận sinh đều có cấp $k \times n$.

2. Bất kỳ *k* từ mã độc lập tuyến tính nào cũng có thể dùng làm ma trận sinh cho mã C[*n , k*].

3. Nếu ma trận G cấp $k \times n$ là một ma trận sinh của mã tuyến tính C[n , k] thì các hàng của G là các mã độc lập tuyến tính trong C.

4. Nếu H là ma trận kiểm tra của mã C[*n , k*] có ma trận sinh là G, thì G là ma trận kiểm tra của mã đối ngẫu $C^{\perp}$, và ngược lại.

Thí dụ 2.7 Cho mã tuyến tính C = {000,001,100,101} trong thí dụ 2.5 .

1/. Tìm ma trận sinh của mã C.

2/. Tìm ma trận kiểm tra của mã C

Giải:

1/. Ta có dim(C) = 2, vậy đây là một mã tuyến tính C[3 , 2], n = 3; k = 2. Ma trận sinh G có cấp k × n. Vậy ma trận sinh của mã C là một ma trận cấp *2 x 3*. Trong C có hai vector mã độc lập tuyến tính là 001 và 100, ma trận sinh G tạo bởi hai từ mã này:

$$G = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Các tin nguồn gồm 2 bit sẽ được mã hóa thành các từ mã 3 bit trong C theo ma trận sinh G.

2/. Ma trận kiểm tra H là ma trận sinh của mã đối ngẫu $C^{\perp}$.

Ta tìm được mã đối ngẫu (xem thí dụ 2.6): $C^{\perp}$ = {000, 010}, chỉ có 1 vector độc lập tuyến tính là 010, vậy ma trận sinh của mã $C^{\perp}$ là:

$$H = (010)$$

Ma trận H chính là ma trận kiểm tra của mã C.

Chú ý rằng nếu ma trận sinh G có cấp k × n, thì ma trận kiểm tra H sẽ có cấp (n-k) × n, tức là cấp 1 × 3 trong thí dụ trên.

2.2.3 Mã hóa dùng ma trận sinh

Nếu $u = (a_0, a_1, \ldots, a_{k-1})$ là một tin cần được mã hóa, thì từ mã *w* tương ứng với *u* nhận được bằng cách nhân vector hàng *u* (cấp $1 \times k$) với ma trận sinh $G_{k \times n}$. Tức là *u* được mã hóa bởi:

$$w = u.G$$

Vì mọi từ mã tương ứng vói các tin được sinh ra bởi ma trận G, nên G được gọi là ma trận sinh của bộ mã.

Thí dụ 2.8

Cho ma trận sinh G của một mã tuyến tính C[7, 4] như sau:

$$G_{4\times 7} = \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Nếu $u = (1101)$ là thông tin cần mã hoá thì từ mã tương ứng là

theo công thức (2.3), $w = u.G = a_0g_0 + a_1g_1 + \ldots + a_{k-1}g_{k-1} = 1.g_0 + 1.g_1 + 0.g_2 + 1.g_3$

$$w = (1\ 1\ 0\ 1) \cdot \begin{pmatrix} 1\ 1\ 0\ 1\ 0\ 0\ 0 \\ 1\ 0\ 1\ 1\ 1\ 0\ 0 \\ 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 0\ 0\ 0\ 1 \end{pmatrix} = (1100\ 1\ 0\ 1)$$

Như vậy, một tin nguồn gồm 4 bit đã được mã hóa thành một mã gồm 7 bit, 4 bit chứa thông tin 3 bit thêm vào là để chống nhiễu.

2.2.4 Giải mã nhờ ma trận sinh

Giả sử $u = (a_0, a_1, \ldots, a_{k-1})$ là một tin đã được mã hóa thành từ mã $w = (b_0, b_1, \ldots, b_{n-1})$ thuộc mã tuyến tính C[n , k], nhờ ma trận sinh G.

Vấn đề dặt ra là khi nhận được từ mã w (giả thiết là không có nhiễu), ta cần giải mã để tìm tin nguồn u, nói cách khác ta cần xác định các $a_0, a_1, \ldots, a_{k-1}$ từ các giá trị $b_0, b_1, \ldots, b_{n-1}$.

Giả sử có ma trận sinh $G = \begin{pmatrix} g_0 \\ g_1 \\ \cdots \\ g_{k-1} \end{pmatrix} = \begin{pmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ & & \ldots\ldots & \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix}$

Từ công thức mã hóa $u.G = w$, ta có:

$$(a_0, a_1, \ldots, a_{k-1}) \cdot \begin{pmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ & & \ldots\ldots & \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{pmatrix} = (b_0, b_1, \ldots, b_{n-1})$$

ta có hệ n phương trình để xác định k ẩn $a_0, a_1, \ldots, a_{k-1}$, ($k < n$):

$$\sum_{i=0}^{k-1} g_{i,j} a_i = b_j \ , \ (j = 0, 1, \ldots, n-1) \qquad (2.4)$$

Do hạng của ma trận G bằng k, hệ luôn có nghiệm, chỉ cần chọn k phương trình đơn giản nhất để xác định k ẩn số a_i.

Thí dụ 2.9. Giả sử mã tuyến tính C[7 , 4] có ma trận sinh G:

$$G = \begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,0\,0\,1\,1 \\ 1\,0\,1\,0\,0\,0\,1 \end{pmatrix} \qquad (2.5)$$

Giả sử từ mã nhận được là w = $(b_0, b_1, b_2, b_3, b_4, b_5)$. Hãy xác định tin nguồn u_1?

Giải: Lập hệ phương trình (2.2) với ma trận G và w trên , ta có hệ:

$$\begin{cases} 1a_0 + 1a_1 + 0a_2 + 1a_3 = b_0 \\ 1a_0 + 0a_1 + 1a_2 + 0a_3 = b_1 \\ 0a_0 + 1a_1 + 0a_2 + 1a_3 = b_2 \\ 1a_0 + 1a_1 + 0a_2 + 0a_3 = b_3 \\ 0a_0 + 1a_1 + 0a_2 + 0a_3 = b_4 \\ 0a_0 + 0a_1 + 1a_2 + 0a_3 = b_5 \\ 0a_0 + 0a_1 + 1a_2 + 1a_3 = b_6 \end{cases}$$

Chọn 4 phương trình đơn giản nhất (chứa nhiều hệ số 0). Giải ra ta có:

$$\begin{cases} a_0 = b_3 + b_4 \\ a_1 = b_4 \\ a_2 = b_5 \\ a_3 = b_5 + b_6 \end{cases} \qquad (2.6)$$

Ta nhận được công thức giải mã (2.6) cho mã tuyến tính có ma trận sinh G trong (2.5). Khi đó với mỗi từ mã w nhận được, ta sẽ giải mã được tin nguồn u.

- Giả sử từ mã nhận được là w_1 = (1 1 0 0 1 0 1), theo công thức giải mã (2.6), ta xác định được tin nguồn là u_1 = (1 1 0 1).
- Giả sử từ mã nhận được là w_2 = (1 0 0 1 0 1 1), theo công thức giải mã (2.6), ta xác định được tin nguồn là u_2 = (1 0 1 0).

Bạn đọc có thể mã hóa u_2 bằng ma trận sinh G, để xem kết quả có trùng với từ mã w_2?

Chú ý: Việc giải mã dùng ma trận sinh như trên chỉ có tính lý thuyết, chỉ được thực hiện khi *w* là một từ mã (không có lỗi). Trong thực tế, các kênh truyền thường có nhiễu, khi gửi đi một từ mã *w*, ta nhận được không phải là *w*, mà là $w' \neq w$, do đó việc giải mã gồm 2 giai đoạn: trước hết, từ w' phải xác định được đúng từ mã *w* đã gửi đi, giai đoạn 2 là xác định tin u từ *w* như đã trình bày ở trên. Các khái niệm giải mã trong các phần sau được hiểu theo nghĩa xác định chính xác từ mã đã gửi đi.

Sai lầm SV thường mắc phải: khi yêu cầu giải mã một xâu nhận được *x*, SV thường coi *x* là một từ mã và dùng ma trận sinh để giải mã như trên, điều này sai vì *x* chưa chắc đã là từ mã. Điều cần làm là khi nhận được *x* phải xác định từ mã nào đã được gửi đi (sẽ học trong các phần sau)

2.3 CÁC MÃ TƯƠNG ĐƯƠNG

2.3.1 Sự tương đương của các mã

Chúng ta đã biết rằng, mỗi mã C có ba đặc trưng chủ yếu là:

1. Chiều dài của các từ mã trong C: *n*
2. Tổng số các từ mã trong C: *M*
3. Khoảng cách Hamming cực tiểu của mã C: *d = d(C).*

Ta ký hiệu (*n, M, d*)-code là mã có độ dài *n,* chứa M từ mã và có khoảng cách cực tiểu *d = d(C).*

Các tham số trên có ý nghĩa gì trong lý thuyết mã hóa?

Chiều dài từ mã n: từ mã càng dài thì việc truyền tin càng kém hiệu quả, tuy nhiên việc mã hóa chống nhiễu là việc thêm vào các thông điệp gốc một số bit, cần thêm vào một số bit tối thiểu để đạt được mục tiêu chống nhiễu. Nói chung chiều dài từ mã càng ngắn càng tốt.

Tổng số từ mã M: Mã càng có nhiều từ mã càng tốt, vì nó có thể truyền tải được nhiều thông điệp gốc, tức là chứa được nhiều thông tin.

Khoảng cách Hamming cực tiểu của mã d: là sự sai khác nhau tối thiểu giữa các từ mã, do đó, khoảng cách này càng lớn thì sự khác biệt giữa các từ mã càng lớn, do đó khả năng phát hiện lỗi và sửa lỗi càng cao.

Vì vậy, các tham số trên là các đặc trưng cơ bản của một bộ mã. Ta sẽ còn phân tích chi tiết hơn các đặc trưng này trong các phần sau.

Dựa trên các đặc trưng trên, hai mã tương đương là hai mã có cùng các tham số đặc trưng *n, M* và *d*. Hai mã tương đương sẽ có hiệu quả truyền tin như nhau, và quan trọng hơn là số các lỗi có thể phát hiện và sửa được là như nhau. Vì vậy thay vì sử dụng một bộ mã C đã cho, ta có thể sử dụng một bộ mã tương đương với C có dạng đơn giản nhất để việc tính toán thuận lợi, mà vẫn đạt được mọi hiệu quả như bộ mã ban đầu.

Định nghĩa 2.9.

> *Hai mã (n , M)-code trên cùng bảng ký hiệu mã, được gọi là tương đương nếu một trong hai mã đạt được từ mã còn lại bằng việc hoán vị tọa độ các từ mã, hoặc nhân một tọa độ các từ mã với một số khác không.*

Nhận xét:

Từ định nghĩa trên ta thấy việc hoán vị tọa độ hoặc nhân một tọa độ với 1 số khác không sẽ không làm thay đổi các tham số đặc trưng của các mã. Tức là nếu C là một *(n, M, d)-code* thì C‘ cũng là một *(n, M, d)-code*. Theo ý nghĩa này, các mã được coi là tương đương.

Chú ý

1. Từ định nghĩa trên, ta thấy rằng hai mã là tương đương thì trước hết phải phải là các mã cùng độ dài và có cùng số từ mã.

2. Hai mã C và C’ có cùng độ dài và có cùng số từ mã gọi là tương đương nếu chúng có cùng khoảng cách Hamming cực tiểu, tức là d(C) = d(C’)

3. Với mã nhị phân, việc nhân một tọa độ với một số vô hướng khác không trên trường F_2 là nhân với 1, mã sẽ không đổi. Do đó hai mã nhị phân tương đương là các mã chỉ khác nhau về hoán vị tọa độ của các từ mã.

4. Các mã nhị phân tuyến tính tương đương cũng theo định nghĩa trên, đó là các mã tuyến tính có cùng các tham số (n, M, d).

Thí dụ 2.10

Cho mã nhị phân tuyến tính C = {000, 101, 010, 111}, mã C’= {000, 011, 100, 111} nhận được từ mã C bằng cách đổi chỗ tọa độ thứ nhất với tọa độ thứ hai của các từ mã. Vậy hai mã C và C’ là tương đương.

2.3.2 Ma trận sinh của các mã tuyến tính tương đương

Định nghĩa 2.10

Cho C[n, k] và C'[n, k] là hai mã tuyến tính tương đương trên trường GF(q), khi đó:

1. *Ma trận sinh G và G' của 2 mã tương đương là các ma trận sinh tương đương.*
2. *Ma trận kiểm tra H và H' của 2 mã tương đương là các ma trận tương đương.*

Nếu G và G' là hai ma trận sinh tương đương ta ký hiệu G ~ G', tương tự, các ma trận kiểm tra tương đương được ký hiệu: H ~ H'. Định lý sau trình bày các phép biến đổi các ma trận sinh để được các ma trận sinh tương đương. Các ma trận kiểm tra cũng được biến đổi tương tự để nhận được các ma trận kiểm tra tương đương.

Định lý 2.2

Cho G và G' là hai ma trận sinh của hai mã tuyến tính tương đương C[n, k] và C'[n, k] trên trường GF(q), khi đó mỗi ma trận có thể nhận được từ ma trận còn lại nhờ áp dụng các luật biến đổi sau:

R1. Đổi chỗ các hàng cho nhau,

R2. Nhân một hàng với 1 số khác 0 trên trường GF(q),

R3. Nhân một hàng với 1 số trên trường GF(q), rồi cộng vào hàng khác.

C1. Đổi chỗ các cột cho nhau,

C2. Nhân một cột với số khác 0 trên trường GF(q).

Việc chứng minh định lý khá đơn giản, dành cho SV như là bài tập.

Các luật trên được gọi là các luật biến đổi tương đương cho các ma trận sinh để nhận được một ma trận sinh mới tương đương với ma trận ban đầu. Với các ma trận kiểm tra cũng áp dụng các luật biến đổi này.

Nhận xét:

Trên trường F_2, luật (R2) thực chất là không thực hiện gì, vì nhân 1 hàng với số khác 0 (là 1) thì không thay đổi, còn luật (R3) là thực hiện việc cộng hàng này với hàng khác của ma trận.

Tương tự, luật C2 cũng không thực hiện gì. *Như vậy, các luật biến đổi tương đương cho các ma trận sinh trên trường F2 chỉ còn luật (R1), (R3) và (C1).*

Chú ý quan trọng : Khi sử dụng các phép biến đổi tương đương trên đây, ta nhận được một ma trận sinh G' của một mã tuyến tính C' tương đương với mã C, nhưng nói chung C' khác với C. Nếu chỉ dùng các phép biến đổi tương đương trên các hàng (R1, R2, R3) thì sẽ nhận được ma trận sinh G' của chính mã C, khi đó có thể dùng ma trận G' thay cho ma trận G của cùng mã C.

Định nghĩa 2.11.

1. *Ma trận sinh G của mã tuyến tính C[n, k] gọi là ở dạng chuẩn nếu G có k cột đầu tiên là ma trận đơn vị, tức là* $G = (I_k \mid A)$ *với* I_k *là ma trận đơn vị cấp* $k \times k$ *còn A là ma trận cấp* $k \times (n-k)$.
2. *Ma trận kiểm tra H của mã tuyến tính C[n , k] gọi là ở dạng chuẩn nếu H có n – k cột cuối là ma trận đơn vị, tức là* $H = (B \mid I_{n-k})$, *với* I_{n-k} *là ma trận đơn vị cấp n-k, còn B là ma trận cấp* $(n-k) \times k$.

Các ma trận sinh ở dạng chuẩn có ưu điểm là việc biểu diễn và tính toán đơn giản. Do đó, với mã tuyến tính C[n, k] có ma trận sinh G, người ta thường biến đổi G để nhận được ma trận sinh tương đương G' ở dạng chuẩn và thay thế mã C bởi mã C' sinh bởi G'. Nếu G'nhận được từ G nhờ các biến đổi R1, R2, R3 thì G' là ma trận sinh dạng chuẩn của chính mã C.

Việc thay thế các ma trận kiểm tra H bởi các ma trận kiểm tra tương đương H' ở dạng chuẩn cũng được thực hiện tương tự.

Thí dụ 2.11

Cho mã nhị phân tuyến tính C = {000, 101, 010, 111} trong thí dụ 2.9. Mã này có tối đa 2 từ mã độc lập tuyến tính, đó là (0 1 0) và (1 1 1), vậy ma trận sinh của mã là:

$$G = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Do dim(C) = 2, mã C là một [3, 2]-code, nhận được từ việc mã hóa tập tin U = {00, 01, 10, 11} thành các từ mã 3 bit nhờ ma trận sinh G.

- Ta dùng các phép biến đổi tương đương để đưa G về ma trận sinh dạng chuẩn:

Từ $G = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$, dùng luật (C1): đổi chỗ 2 cột đầu, ta có : $G \sim \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$, dùng luật (R3): cộng hàng 1 vào hàng 2, ta có: $G \sim G' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$. Ma trận G' có 2 cột đầu là ma trận đơn vị I_2, vậy G' là ma trận sinh dạng chuẩn. Ta xác định C' bằng cách mã hóa tập tin U = {00, 01, 10, 11} nhờ ma trận G', ta nhận được mã C' = {000, 011, 100, 111} (SV tự kiểm tra việc tính các từ mã này). Mã C' khác với mã C nhưng C' ~ C và cùng là các mã tuyến tính C[3, 2].

- Bây giờ ta dùng các phép biến đổi tương đương chỉ trên các hàng để đưa G về ma trận sinh dạng chuẩn:

Từ $G = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$, dùng luật (R1): đổi chỗ hàng 1 với hàng 2, ta được $G \sim \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, dùng luật (R3): cộng hàng 2 vào hàng 1, ta có:

$G \sim G_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$. Ma trận G_1 có 2 cột đầu là ma trận đơn vị I_2, vậy $G_1 \sim G$.

Mặt khác, nếu mã hóa tập tin U = {00, 01, 10, 11} nhờ ma trận G_1 ta nhận được mã C_1 = {000, 010, 101, 111}, rõ ràng mã này trùng với mã C, vậy G và G_1 cùng sinh mã C, nên ma trận sinh dạng chuẩn G_1 có thể thay thế cho ma trận G trong các tính toán liên quan đến mã C.

Thí dụ 2.12 Cho mã tuyến tính C[7, 4, 3] có ma trận sinh:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Hãy tìm ma trận sinh dạng chuẩn tương đương với ma trận G

Cách 1: (nhanh) áp dụng các phép biến đổi tùy ý, theo cả hàng và côt, do G đã có sẵn một số cột là vector đơn vị, ta chỉ cần áp dụng các luật C1: đổi chỗ cột 1 cho cột 6, sau đó đổi chỗ cột 3 cho cột 4, ta nhận được ma trận G' tương đương với ma trận G:

$$G' = \begin{bmatrix} 1&0&0&0&1&0&1 \\ 0&1&0&0&1&1&1 \\ 0&0&1&0&1&1&0 \\ 0&0&0&1&0&1&1 \end{bmatrix}$$

G' là ma trận sinh dạng chuẩn của một mã tuyến tính C'[7, 4, 3] tương đương mã đã cho.

Cách 2: Chỉ áp dụng các luật biến đổi theo hàng R1 , R3, ta sẽ *đưa G về dạng chuẩn*.

Từ $G = \begin{bmatrix} 1&0&0&0&1&1&0 \\ 1&1&0&0&1&0&1 \\ 0&0&0&1&1&0&1 \\ 1&0&1&0&0&0&1 \end{bmatrix}$ áp dụng R3 2 lần: cộng hàng 1 vào hàng 2 và cộng hàng 1 vàohàng 4:

$G \sim \begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&0&1&1 \\ 0&0&0&1&1&0&1 \\ 0&0&1&0&1&1&1 \end{bmatrix}$ áp dụng R1: đổi chỗ hàng 3 và hàng 4, ta nhận được ma trận G' tương đương với G:

$G' = \begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&0&1&1 \\ 0&0&1&0&1&1&1 \\ 0&0&0&1&1&0&1 \end{bmatrix}$. G' là ma trận sinh dạng chuẩn của ma trận G, và G' sinh ra chính mã C.

Nhận xét: Để mã hóa với ma trận sinh dạng chuẩn, ta có *k* bit đầu tiên trong từ mã *w* chính là *k* bit của tin nguồn *u* , chỉ cần tính *n – k* bit chống nhiễu ứng với *n – k* cột cuối của ma trận sinh G. Việc giải mã cũng thuận lợi: *k* bit đầu tiên của từ mã *w* sẽ là tin nguồn *u*.

Thí dụ 2.13 Cho mã tuyến tính C[7, 4] có ma trận sinh $G_{4\times 7}$ như sau:

$$G = \begin{bmatrix} 1&0&0&0&1&1&0 \\ 0&1&0&0&0&1&1 \\ 0&0&1&0&1&1&1 \\ 0&0&0&1&1&0&1 \end{bmatrix}$$

Giả sử cần mã hóa tin $u = 1101$, ta có $w = (1101).G$, kết quả tính được $w = (1101000)$.

Để giải mã từ mã $w = 1101000$, lập hệ phương trình xác định các a_i từ ma trận G và các b_j, với $u = a_0\, a_1\, a_2\, a_3$, $w = b_0\, b_1\, b_2\, b_3\, b_4\, b_5\, b_6 = 1\ 1\ 0\ 1\ 0\ 0\ 0$. Thực tế chỉ cần hệ 4 phương trình:

$$\sum_{i=0}^{k-1} g_{i,j} a_i = b_j \ , \ (j = 0,1,...,n-1)$$

$$\begin{cases} 1a_0 + 0a_1 + 0a_2 + 0a_3 = 1 \\ 0a_0 + 1a_1 + 0a_2 + 0a_3 = 1 \\ 0a_0 + 0a_1 + 1a_2 + 0a_3 = 0 \\ 0a_0 + 0a_1 + 0a_2 + 1a_3 = 1 \end{cases} \rightarrow \text{Giải ra ta có:} \quad \begin{cases} a_0 = 1 \\ a_1 = 1 \\ a_2 = 0 \\ a_3 = 1 \end{cases}$$

Vậy giải mã từ mã $w = (1101000)$, ta nhận được tin ban đầu là $u = 1101$

2.4 MỘT SỐ GIỚI HẠN CỦA MÃ TUYẾN TÍNH TỐI ƯU

Chúng ta đã biết rằng, mỗi mã C có ba đặc trưng chủ yếu là:

1. Chiều dài của các từ mã trong C: *n*
2. Tổng số các từ mã trong C: *M*
3. Khoảng cách Hamming cực tiểu của mã C: *d = d(C).*

Ta ký hiệu (*n, M, d*)-code là mã có độ dài *n,* chứa M từ mã và có khoảng cách cực tiểu *d = d(C).*

Các tham số trên có ý nghĩa gì trong lý thuyết mã hóa?

Chiều dài từ mã n: từ mã càng dài thì việc truyền tin càng kém hiệu quả, tuy nhiên việc mã hóa chống nhiễu là việc thêm vào các thông điệp gốc một số bit, cần thêm vào một số bit tối thiểu để đạt được mục tiêu chống nhiễu. Nói chung chiều dài từ mã càng ngắn càng tốt.

Tổng số từ mã M: Mã càng có nhiều từ mã càng tốt, vì nó có thể truyền tải được nhiều thông điệp gốc, tức là chứa được nhiều thông tin.

Khoảng cách Hamming cực tiểu của mã d: là sự sai khác nhau tối thiểu giữa các từ mã, do đó, khoảng cách này càng lớn thì sự khác biệt giữa các từ mã càng lớn, do đó khả năng phát hiện lỗi và sửa lỗi càng cao.

Thông thường, một mã C(*n, M, d*) với d là khoảng cách Hamming cực tiểu của mã, *d = d(C*), được gọi là mã tối ưu nếu nó có *n* nhỏ (để có thể truyền nhanh các từ mã), M lớn (có thể truyền một số lớn các từ mã) và *d* lớn (có thể sửa được nhiều lỗi). Điều này liên quan đến mục tiêu của các bài toán tối ưu các mã, nhất các mã tuyến tính. Các mục tiêu đó gọi là các giới hạn của mã tuyến tính tối ưu. Trong các mã tuyến tính tối ưu ta có một số giới hạn quan trọng sau:

2.4.1. Giới hạn Griesmer

Đối với mã tuyến tính [*n, k, d*]-code, với *k* và $d = d_{\min}$ đã được xác định, phải xây dựng mã có độ dài các từ *n* nhỏ nhất.

Bài toán này tương ứng với giới hạn Griesmer:

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{2^i} \right\rceil$$

Trong đó $\lceil x \rceil$ là số nguyên trần của x: số nguyên nhỏ nhất *không bé hơn x*. Chẳng hạn $\lceil 3.14 \rceil = 4$, $\lceil 5 \rceil = 5$, $\lceil -3.14 \rceil = -3$.

2.4.2. Giới hạn Plotkin

Đối với mã tuyến tính [*n, k, d*], với n và k đã được xác định, phải xây dựng mã có khoảng cách cực tiểu *d* lớn nhất.

Bài toán này tương ứng với giới hạn Plotkin:

$$d_{\min} \leq \frac{n.2^{k-1}}{2^k - 1}$$

2.4.3. Giới hạn Hamming

Đối với mã tuyến tính [*n, k, d*], với *n* và số lỗi có thể sửa được tối đa là *t* đã xác định (do *t*

phụ thuộc vào *d*, nên ở đây coi như *n* và *d* đã xác định), phải xây dựng mã có số ký tự *k* của thông điệp cần truyền là lớn nhất (hay số (*n* - *k*) nhỏ nhất).

Bài toán này tương ứng với giới hạn Hamming:

$$2^{n-k} \geq \sum_{i=0}^{t} C(n,i)$$

trong đó giá trị t = $\lfloor \frac{d-1}{2} \rfloor$ (ở đây ký hiệu $\lfloor x \rfloor$ là số nguyên sàn của x: số nguyên lớn nhất không vượt quá x, chẳng hạn $\lfloor 3.14 \rfloor = 3$; $\lfloor 5.0 \rfloor = 5$; $\lfloor -3.14 \rfloor = -4$).

Ký hiệu *C(n, i)* là số tổ hợp chập *i* của *n* (còn được ký hiệu là C_n^i) và được tính bằng công thức:

$$C(n, i) = \frac{n!}{i!\,(n-i)!}, \text{ hoặc } C(n, i) = \frac{n.(n-1)...(n-i+1)}{i!}$$

Chú ý rằng *C(n, 0) = 1; C(n, n) =1; C(n, 1) = n; C(n, i) = C(n, n – i)*

Trong giáo trình này, để đơn giản, trong các phần sau ta chỉ trình bày các giới hạn trên cho các mã là những mã nhị phân tuyến tính và lập luận tương tự cho trường hợp tổng quát.

BÀI TẬP CHƯƠNG 2

1. Tính khoảng cách Hamming giữa các cặp xâu bit sau:
   a. 1111101; 1000011 — 5
   b. 00001110; 01110001 — 7
   c. 101000101; 010110010 — 8
   d. 0000000000; 1111111111 — 10

* Chú ý: Với các mã nhị phân, Có thể dùng công thức d(x,y) = w(x + y)

2. Cho mã tuyến tính C = {000 ; 110 ; 011 ; 101}
   a. Tìm ma trận sinh G của mã C
   b. Tìm mã đối ngẫu của C. ĐA:{000, 111}. Tìm ma trận kiểm tra H của mã C.
   c. Tính khoảng cách cực tiểu d(C) của mã ĐA: (d = 2).

* Chú ý: Với các mã nhị phân tuyến tính, có công thức d(C) = min {w(x) | ∀ x ∈ C, x ≠ O}

3. Cho mã tuyến tính C = {0000 ; 1100 ; 0011 ; 1111}
   a. Tìm ma trận sinh G của mã C. Tìm ma trận G' ~ G, từ G' hãy xác định mã C'. So sánh mã C' và mã C.
   b. Tìm mã đối ngẫu của C. (ĐA là C).Tìm ma trận kiểm tra H của mã C. (ĐA: H = G)
   c. Tính khoảng cách cực tiểu d(C) của mã. ĐA: (d = 2)

4. Giả sử rằng mã C có ma trận sinh:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

   a. Những từ mã nào trong mã C được tạo ra bởi ma trận sinh này?
   b. Hãy mã hóa thông điệp u = 0110 nhờ ma trận sinh G. Giải mã từ mã w = 1110100 nhờ ma trận G.

5. Giả sử rằng mã C có ma trận sinh:

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}$$

   a. Hãy *đưa ma trận G về dạng chuẩn* G' = (I_4 | A) (chú ý là chỉ biến đổi theo các hàng).
   b. Những từ mã nào trong mã C được tạo ra bởi ma trận sinh này?

c. Tìm ma trận kiểm tra H của mã C. (gợi ý: từ dạng chuẩn G = (I_4 | A) của mã C, tìm H theo công thức H = (A^T | I_3), xem định lý dưới đây.

Định lý: Nếu mã tuyến tính C[n , k] trên trường GF(q) có ma trận sinh dạng chuẩn G = (I_k | A) thì ma trận kiểm tra dạng chuẩn của mã C được xác định bởi: H = ($-A^T$ | I_{n-k}) ,
với A^T là ma trận chuyển vị của ma trận A trong ma trận sinh dạng chuẩn G.

Với các mã nhị phân tuyến tính, thì H = (A^T | I_{n-k})

6. Cho mã tuyến tính C có ma trận kiểm tra H = $\begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$

a/. Tìm ma trận sinh của mã C. ĐA : G = $\begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}$

b/. Viết tất cả các từ mã của C. ĐA : {00000, 10110, 01011, 11101}

7. Cho mã tuyến tính C trên trường $F_3 = \{0, 1, 2\}$ có ma trận sinh:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}$$

a/. Tìm tất cả các từ mã của mã C

b/. Đưa ma trận sinh G về dạng chuẩn bằng các phép biến đổi theo hàng (R1, R2, R3)

c/. Tìm mã đối ngẫu $C^{\perp}$ và ma trận kiểm tra H của mã C. (gợi ý: tìm ma trận kiểm tra H theo định lý trên: $H = (-A^T \mid I_{n-k})$, mã đối ngẫu của C là mã nhận H làm ma trận sinh)

8. Trên trường F_3 , cho mã C có ma trận sinh:

$$G = \begin{bmatrix} 1 & 1 & 2 & 0 & 1 \\ 2 & 0 & 1 & 2 & 1 \end{bmatrix}$$

a/. Viết tất cả các từ mã của C

b/. Biến đổi chỉ theo hàng để đưa ma trận sinh G về dạng chuẩn.

c/. Tìm ma trận kiểm tra của mã C có ma trận sinh trên đây

9. Phải xây dựng mã C với độ dài từ mã n là bao nhiêu để mã tuyến tính C[n, 4, 3] là mã tối ưu đạt giới hạn Griesmer

10. Phải xây dựng mã C với khoảng cách Hamming cực tiểu d là bao nhiêu để mã tuyến tính C[7, 4, d] là mã tối ưu đạt giới hạn Plotkin.

11. Cần xây dựng mã tuyến tính C[7, k, 3] để mã hóa các thông điệp gốc gồm k bit thành từ mã 7 bit, số bit của thông điệp gốc là bao nhiêu để C là mã tối ưu đạt giới hạn Hamming.

TÀI LIỆU THAM KHẢO

[1] Raymond Hill, (1993). *"A First Course in Coding Theory"*, Clarendon Press, Oxrord, USA. ISBN : 0-19-853803-0.

[2] Yehuda Lindell, *"Introduction to Coding Theory"*, Lecture Notes, Department of Computer Science Bar-Ilan University , Israel (2010).

[3] Tom Richardson, Rudiger Urbanke. *"Modern Coding Theory"*. Cambridge University Press (2008)

[4] Đặng Văn Chuyết, Nguyễn Tuấn Anh.(1998) *"Giáo trình Cơ sở lý thuyết truyền tin"*. NXB Giáo dục.