Cryptography and Coding Theory

Jaimes Joschko
University of Victoria

September 29, 2022

Contents

Chapter 1

Communication Systems

Communication is fundamentally a process of transferring information from a sender to a receiver. [Hail 08]

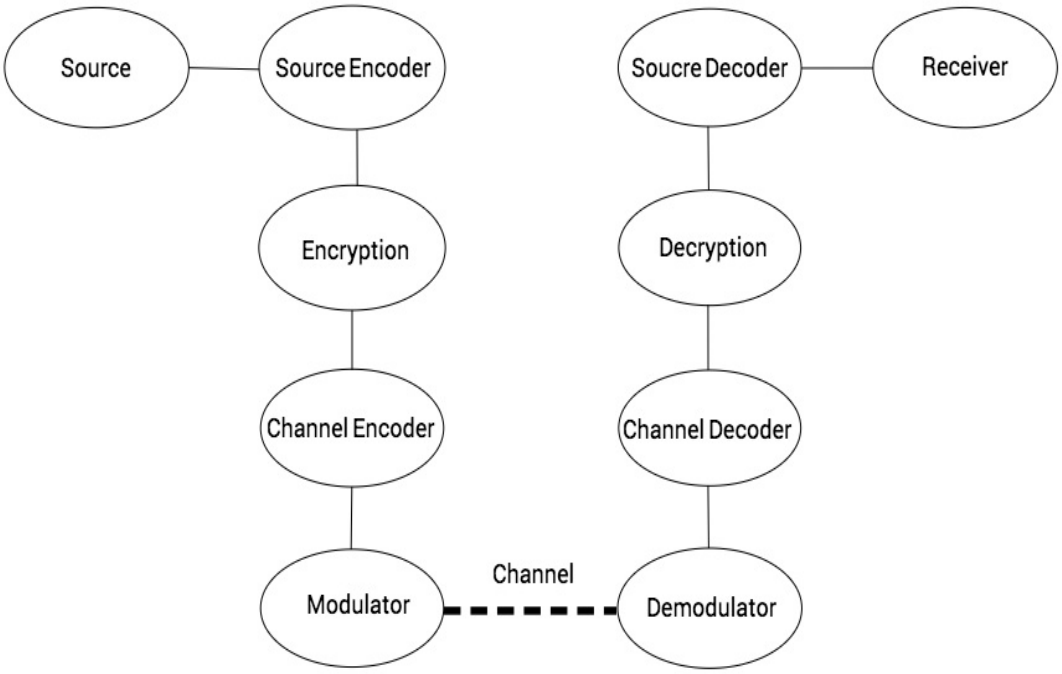


Figure 1.1: Standard Model of a Communication System

1.1 Source

i Also called the Data Source

ii It produces output (that is continuous or discrete)

iii voice or text

1.2 Source Encoder

i Transforms source data into binary/bits

ii Goal: minimize the number of bits required to represent source data well still allowing it to be reconstructed.

usually involves "compression" to remove unnecessary redundancy.

iii Ex. jpeg or mp3 (compression)

1.3 Channel encoder

1. Goal: to maximize the rate at which info can be reliably transmitted across the channel in the presence of disruptions that can introduce errors (noise).
2. Inside: error correction encoding, located inside the encoder. It adds redundancy to messages to allow transmission errors to be detected or corrected.
3. 21 questions for guessing a number, convert to binary, ask "is the first digit a one?", "second?", ... with lying, ask twice each digit, potentially have to three time if they lie.

1.4 Encryption

1. intended to make the data unintelligible to all but the intended receiver, (i.e., to preserve secrecy in the presence of unwelcome monitoring of the channel).
2. What does preserve secrecy mean ?

   -over the life-span of the data? -active intrusion?
3. Cryptography: Science of maintaining the secrecy of data from passive intrusion (eavesdropping) or active intrusion (disruption, or introduction of an altered messge).
4. Authentication: knowing who actually sent the message

   corroboration that the origin of the message is as claimed.
5. data integrity: property that the data has not been changed in an unauthorized way.

Chapter 2

Coding Theory

2.1 General Codes

2.1.1 Basic terminology

Definition 2.1.1. The **alphabet** of a code is any non-empty finite set of symbols.

Usually we use $K = \{0, 1\}$ for our alphabet. The elements of K in this case are (binary) digits, or bits.

Definition 2.1.2. A **word** is a finite sequence of elements from an alphabet.

Definition 2.1.3. The **length** of a word is the length of the sequence of the word.

For example, from the above alphabet K, 1111, 0000 and 1010 are all length 4 words, while 101010 is a length 6 word.

Definition 2.1.4. A **code** is a set C of words over an alphabet K. The words in C are called **codewords**.

Ex.

$$C_1 = \{00, 01, 10, 11\}$$
$$C_2 = \{000, 001, 01, 1\}$$

Definition 2.1.5. A **Block Code** is a code in which all codewords are of the same length. If they are all of length n, then n is the length of the code.

Ex. C_1 is a binary block code of length 2.

Definition 2.1.6. A **Prefix Code** is a code such that there does not exist different codewords w_i and w_j so that w_i is a prefix (initial segment) of w_j.

For example C_2 is a prefix code. Prefix codes are easy to decode since no codeword is a prefix of any other. Using C_2, suppose we have this correspondence.

$$000 \to N, 001 \to S, 01 \to E, 1 \to W$$

Then, $001011100000101011 = 001\ 01\ 1\ 1\ 000\ 001\ 01\ 01\ 1 = SEWWNSEEW$.

2.1.2 Assumptions about the Channel

To get started we need to make some assumptions about the channel. We will assume the following:

1. If n symbols are transmitted then n symbols are received (maybe not the same ones, error may occur).
2. There is no difficulty identifying the beginning of the 1st word transmitted.
3. The channel noise is scattered randomly rather than occurring in bursts.
4. The probability that one digit is altered in transmission is independent of the digit.

Definition 2.1.7. A binary channel is **symmetric** if 0 and 1 are transmitted with equal accuracy. The **reliability** of such a channel is the probability, p, $0 \leq p \leq 1$, that the digit sent is the digit received.

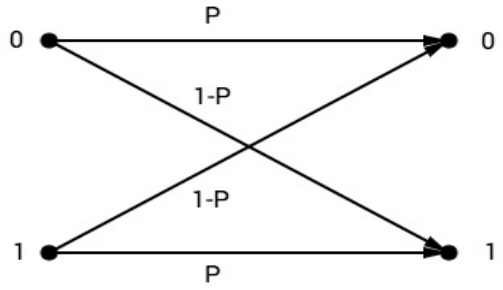


Figure 2.1: Binary Symmetric Channel

For example, if we are transmitting words of length 4 over over binary symmetric channel (BSC) with reliability p, then

$$\text{Probability of 0 errors is } \binom{4}{0} p^4 (1-p)^0$$

$$\text{Probability of 1 errors is } \binom{4}{1} p^3 (1-p)^1$$

$$\text{Probability of 2 errors is } \binom{4}{2} p^2(1-p)^2$$

$$\text{Probability of 3 errors is } \binom{4}{3} p^1(1-p)^3$$

$$\text{Probability of 4 errors is } \binom{4}{4} p^0(1-p)^4$$

2.1.3 Morse Code

The artist Samuel F. B. Morse (1971-1872) worked on developing the telegraph and created Morse Code. Morse code is a ternary (vs. binary) code, where the three characters are the **dot**, the **dash**, and the **space**.

For example, here are some standard correspondences.

A •−	J •−−−	S •••	1 •−−−−
B −•••	K −•−	T −	2 ••−−−
C −•−•	L •−••	U ••−	3 •••−−
D −••	M −−	V •••−	4 ••••−
E •	N −•	W •−−	5 •••••
F ••−•	O −−−	X −••−	6 −••••
G −−•	P •−−•	Y −•−−	7 −−•••
H ••••	Q −−•−	Z −−••	8 −−−••
I ••	R •−•	0 −−−−−	9 −−−−•

Punctuation Mark	Morse
Full-stop (period)	.-.-.-
Comma	--..--
Colon	---...
Question mark (query)	..--..
Apostrophe	.----.
Hyphen	-....-
Slash ("/")	-..-.
Brackets (parentheses)	-.--.-
Quotation marks	.-..-.
At sign	.--.-.
Equals sign	-...-

Prosign (?)	Morse
AA, New line	.-.-
AR, End of message	.-.-.
AS, Wait	.-...
BK, Break	-...-.-
BT, New paragraph	-...-
CL, Going off the air ("clear")	-.-..-..
CT, Start copying	-.-.-
DO, Change to wabun code	-..---
KN, Invite a specific station to transmit	-.--.
SK, End of transmission (also VA)	...-.-
SN, Understood (also VE)	...-.
SOS, Distress message	...---...

If we let the dot be one unit length, then the dash and the space between letter are three units long, while the space between words is seven units long. If a mistake occurs the sender waits, eight units, then restarts transmission from an appropriate place in the message.

One different between Morse code and the codes we will see in the following sections is that Morse Code is a variable length code, while the other codes we consider will be fixed length codes (see below). Furthermore, Morse code is not very good at preventing errors in transmission. For example, I and EE are very similar in Morse code.

2.1.4 Distance and Weight

If we transmit a word across a channel and the received word is not a codeword, then we can detect that errors have occurred. But, if the received word is a codeword, then there are two cases: (1) maybe no error occurred, or (2) maybe enough errors occurred to change the sent word into another codeword.

Ex. $C_1 = \{00, 01, 10, 11\}$ Every received word is a codeword. Therefore, no errors can be detected and no errors can be corrected.

Ex. Let $C_2 = \{000000, 010101, 101010, 111111\}$. Then, C_2 can detect 2 or more errors since there needs to be atleast 3 changes to transform a codeword into another codeword. Furthermore, C_2 can correct 1 error, but if 2 errors occur then there is to much ambiguity. If 010101 is sent and 110111 is received, then the received word is not a code word. Therefore errors have occurred. C_2 can correct 1 or fewer errors by majority rule (i.e., if 101010 is sent and 101110 is received, then decode by majority rule to 101010.) But, we can't correct 2 errors this way.

Ex. Let $C_3 = \{000, 011, 101, 110\}$. C_3 is obtained from C_1 by adding a bit to guarantee the number of 1's is even. In other words we are adding a parity check digit (i.e., sum mod 2= 0). The same technique is used in VISA and MASTER CARD. C_3 can detect 1 or fewer errors, but can't correct any errors since if 111 is received and 1 error has occurred then one of 011, 101, 110 could have been sent.

Definition 2.1.8. If u and v are binary words of the same length then $u+v$ is the binary word obtained by component wise addition modulo 2 (i.e., $0+0=0, 0+1=1, 1+0=1, 1+1=0$, or take the xor of the digits being combined). For example, $01101 + 11011 = 10110$.

Definition 2.1.9. Let v be a binary word of length n. The **hamming weight**, or just weight, of v, denoted $wt(v)$, is the number of occurrences of the digit "1" in v. For example, $wt(1010) = 2$.

Definition 2.1.10. Let u and v be binary words of length n, the **hamming distance** between u and v, denoted $d(u, v)$, is the number of position in which u and v differ (i.e., $wt(u + v)$). For example, $d(01011, 00111) = 2$.

Definition 2.1.11. For a code C with $|C| \geq 2$ the (minimum) **distance**, d, of C is the $min[d(u, v)] = min[wt(u + v)]$, $\forall u, v \in C$, where $u \neq v$.

For example, Let $C = \{0000, 1010, 0111\}$. Then,

$$0000 + 1010 = 1010 \quad wt = 2$$

$$0000 + 0111 = 0111 \quad wt = 3$$

$$1010 + 0111 = 1101 \quad wt = 3$$

Therefore, the minimum distance of C is 2.

Definition 2.1.12. Let C be a binary code of length n. If $v \in C$ is sent and w is received then the **error pattern** is $u = v + w$.

Ex. If 1111 is sent and 1100 is received, then the error pattern that occurred is 0011 since $1111 + 1100 = 0011$.

2.1.5 Detecting Errors

Definition 2.1.13. We say C **detects** the error pattern u if and only if $u+v \notin C,\ \forall v \in C$.

For example, let $C = \{001, 101, 110\}$. Then, C detects u=010, since for any $v \in C$, $u + v \notin C$. But, C does not detect $w = 100$ since if 001 is sent and w occurs then $001 + 100 = 101 \in C$ is received.

Theorem 2.1.14. A code C can detect all nonzero error patterns of weight at most $d-1$ if and only if the minimum distance of C is at least d.

Proof. ($\Leftarrow$): Suppose C has minimum distance at least d. let u be a nonzero error pattern with $wt(u) \leq d-1$. Let $v \in C$. Then, $d(v, v+u) = wt(v+v+u) \leq d-1 < d$. Since $u \neq 0$ and C has minimum distance atleast d, $v + u \notin C$. Therefore, u is detected.

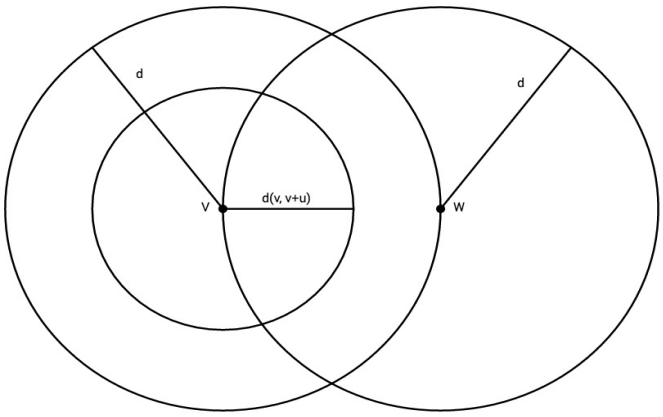


Figure 2.2: The distance between codewords, and the errors able to be detected.

($\Rightarrow$): Suppose C detects all nonzero error patterns of weight less than $d-1$. Let $v, w \in C$. Then $u = v + w$ can not be an error pattern as $v + u = v + (v + w) = w$. Therefore, $wt(u)$ is either 0 or greater than or equal to d. Therefore, C has min distance atleast d. □

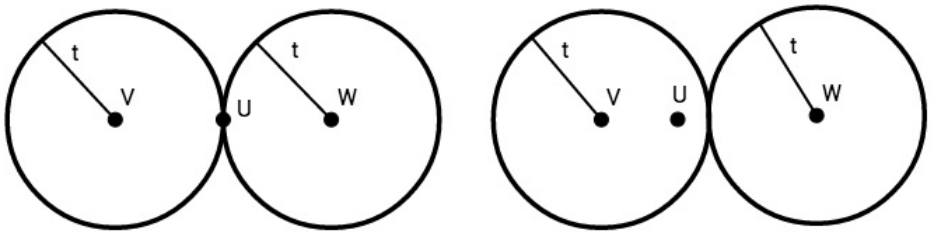
Definition 2.1.15. A code C is **d-error-detecting** if it detects all nonzero error patterns of weight less than or equal to d and does not detect at least one error pattern of weight $d+1$.

Theorem 2.1.16. C has minimum distance d if and only if C is $(d-1)$-error-detecting.

2.1.6 Correcting errors

Definition 2.1.17. A code C **corrects** the error pattern u if and only if $\forall v \in C$, $u + v$ is "closer" (in hamming distance) to v than to any other codeword. In other words,

$$d(v, u + v) < d(w, u + v) \quad \forall w \in C \backslash v.$$



detects but does not correct

detects and corrects

For example, let $C = \{0000, 1010, 0111\}$. Then, C corrects $u_1 = 0100$ since:

If 0000 is sent and u occurs then 0100 received. So,
- $d(0000, 0100) = 1$
- $d(1010, 0100) = 3$
- $d(0111, 0100) = 2$

If 1010 is sent and u occurs then 1110 received. So,
- $d(0000, 1110) = 3$
- $d(1010, 1110) = 1$
- $d(0111, 1110) = 2$

If 0111 is sent and u occurs then 0011 received. So,
- $d(0000, 0011) = 2$
- $d(1010, 0011) = 2$
- $d(0111, 0011) = 1$

But, C does not correct $u_2 = 1010$ since if 1010 is sent and the error pattern u_2 occurs then $0000 \in C$ is received. Thus, the received word is closer to 0000 (trivially) than to the sent word.

Theorem 2.1.18. A code C corrects all nonzero errors patterns of weight less than or equal to t if and only if the minimum distance, d, of C is at least $2t + 1$.

Proof. ($\Rightarrow$): Suppose C corrects all nonzero error patterns of weight less than or equal to t. Let $v, w \in C$. Suppose that $d(v, w) = d$ and $0 < d < 2t + 1$.

Case 1: If d is odd. Let $d = 2s + 1$. Notice $s < t$ since $d = 2s + 1 < 2t + 1$. So, $s + 1 \leq t$. Let u be any error pattern formed from $v + w$ by changing any s 1's to 0's. Then $wt(u) = s + 1$ since

$$wt(v + w) - s = d - s = 2s + 1 - s = s + 1 \leq t.$$

Suppose v is sent and $v + u$ is received. Then, $d(v, v + u) = wt(v + v + u) = s + 1$. But,

$$d(w, v + u) = wt(w + v + u) = wt((w + v) + u) = s$$

since $v + w$ and u differ by s 1's. Thus, $v + u$ is closer to w than to v. Thus, c does not correct u, which is a contradiction contradiction since $wt(u) \leq t$ and C corrects all nonzero error patterns of weight less than or equal to t.

Case 2: If d is even. Let $d = 2s$. Similar as in case 1 except that $d(v, v+u) = d(w, v+u) = s$, which is a contradiction since C can not tell if v or w was sent.

Therefore, the minimum distance of C is at least $2t + 1$.

($\Leftarrow$): Suppose the minimum distance of C is at least $2t+1$. Let u be an error pattern of weight less than t. Suppose $v \in C$ is sent and $v + u$ is received. Then, $d(v, v + u) = wt(v + v+u) \leq t$. Furthermore, for any $w \in C\backslash\{v\}$, $d(w, v+u) = wt(w+v+u) \geq 2t+1-t = t+1$. Hence, $v + u$ is closer to v than any other word w in C. Thus, C corrects u.

Therefore, C corrects all nonzero errors patterns of weight less than or equal to t. □

Definition 2.1.19. A code C is **t-error-correcting** if and only if it corrects all nonzero error patterns of weight less than or equal to t and does not correct some error pattern of weight $t + 1$.

2.1.7 Decoding

Definition 2.1.20. Maximum likelihood decoding is when a received word w is decoded to $v \in C$ if there exists unique $v \in C$ that achieves $min_{x \in C} d(x, w)$; if not, then ties are broken randomly.

Definition 2.1.21. Incomplete max likelihood decoding is when a received word w is decoded to $v \in C$ if there exists unique $v \in C$ that achieves $min_{x \in C} d(x, w)$; if not, then request retransmission.

We assume IMLD.

2.1.8 Main goals for a code

1. Fast encoding of info.
2. Easy transmission of encoded messages (i.e., binary).

3. Fast decoding of received messages.

4. Correction of errors introduced in the channel.

5. Maximum transfer of information per time unit.

2.2 Linear Codes

2.2.1 introduction to linear codes

Definition 2.2.1. A code C is **linear** if $u + v \in C$ whenever $u, v \in C$.

Theorem 2.2.2. The minimum distance of a linear code is the smallest weight of a nonzero codeword.

Proof: see exercises

This allows us to tell how many errors are detected/corrected by looking at $|C|-1$ words, rather than $\binom{|C|}{2}$ sums of pairs of codewords. Some other advantages of linear codes are: easy encoding/decoding and easy to describe patterns detected or corrected.

We will be focusing on binary linear codes. So, our alphabet is $K = \{0, 1\}$. Following from this, K^n is the set of words of length n over K. (Note: K^n with scalar multiplication over K is a vector space.) Let $S \subseteq K^n$, and say $S = \{v_1, v_2, \ldots, v_l\}$. Then,

$$< S >= \{w \in K^n \mid w = a_1v_1 + a_2v_2 + \ldots + a_lv_l, \text{ where } a_i \in K, v_i \in S\}.$$

If $S = \emptyset$, then $< S >= \{0000\ldots00\}$.

Definition 2.2.3. A word $v \in K^n$ is **orthogonal** to S if $v \cdot w = 0$ for all $w \in S$. Note that if v is orthogonal to S then v is orthogonal to all $w \in < S >$.

Definition 2.2.4. The **orthogonal complement** of S, $S^\perp$, is the set of all $v \in K^n$ which are orthogonal to S.

Theorem 2.2.5. If V is a vector space and $S \subseteq V$ then $S^\perp$ is a subspace of V.

Definition 2.2.6. If $S \subseteq K^n$ and $C =< S >$ we still write $C^\perp = S^\perp$ and call $C^\perp$ the **dual code** of C.

For example, if $n = 4$ and $S = \{0100, 0101\}$, then $C =< S >= \{0000, 0100, 0101, 0001\}$. So, to find $C^\perp$. We must find all words $v = wxyz$ such that $v \cdot 0100$ and $v \cdot 0101$. Thus, we have

$$0w + 1x + 0y + 0z = 0 \text{ and } 0w + 1x + 0y + 1z = 0.$$

Thus, $x = z = 0$. Therefore, w and y are arbitrary. So, $S^\perp = C^\perp = \{0000, 0010, 1000, 1010\}$

Theorem 2.2.7. Any nonempty set of vectors $S \subseteq K^n$ has a maximal linearly independent subset.

Definition 2.2.8. If $C =< S >$ is a linear code then the **dimension** of C is the dimension of $< S >$, (i.e., the number of vectors in a basis for $< S >$).

Note: If V is a finite vector space, and S is a subspace of V, then $dim(S) + dim(S^{\perp}) = dim(V)$. So, If C is a linear code of length n then $dim(C) + dim(C^{\perp}) = n$.

Theorem 2.2.9. A linear code $C =< S >$ of dimension k has exactly 2^k codewords.

Proof. Let $B = \{v_1, v_2, \ldots, v_k\}$ be a basis for C. Then, every vector, w, can be uniquely written as $w = a_1v_1 + a_2v_2 + \ldots + a_kv_k$, where $a_i \in K = \{0, 1\}$. There are 2 choices for each a_i. Therefore, there are 2^k choices for $(a_i, a_2, \ldots, a_k)$, and each of these gives a different codeword. □

2.2.2 Generator matrices for linear codes

Recall, A linear code $C =< S >$ of dimension k has exactly 2^k codewords. Let $\{v_1, v_2, \ldots, v_k\}$ be a basis for C. Then, every code word is uniquely represented as $w = \alpha_1v_1 + \ldots + \alpha_kv_k$, $\alpha_i \in \{0, 1\}$. Thus, we can think of the codeword, w, as $(\alpha_1, \ldots, \alpha_k)G$, where G is the $k \times n$ matrix

$$\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix}.$$

G is called the generator matrix.

Definition 2.2.10. A **generator matrix** for a linear code $C =< S >$ is a matrix whose rows are a basis for S.

For example, let $S = \{11101, 10110, 01011, 11010\}$ and $C =< S >$. First, we need to find a basis for C.

$$\begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 01011 \\ 00111 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix}$$

So, $\{11101, 01011, 00111\}$ is a basis for C. Thus, $G = \begin{bmatrix} 11101 \\ 01011 \\ 00111 \end{bmatrix}$ is a generator matrix for C.

Suppose we have the following correspondence:

$$\begin{aligned} 000 &= A \\ 001 &= E \\ 010 &= H \\ 011 &= K \\ 100 &= L \\ 101 &= M \\ 110 &= P \\ 111 &= X \end{aligned}$$

To encode the message $HELP$ we compute $(010)G = 01011, (001)G = 00111, (100)G = 11101$, and $(110)G = 10110$. Thus, we have

$$HELP = 01011\ 00111\ 11101\ 10110.$$

Facts about generator matrices.

1. Codes can have multiple generator matrices.
2. The rows of a generator matrix are linearly independent
3. The $dim(C)$ is the number of rows and the length of C is the number of columns.
4. C always has a generating matrix in RREF.
5. Any matrix G whose rows are linearly independent is a generating matrix for some linear code.

2.2.3 Parity check matrix for linear codes

Definition 2.2.11. A **parity check matrix** for a linear code C is a matrix H whose columns are a basis for $C^{\perp}$.

Facts:

1. $wH = 0$ for any codeword w in C.
2. H^T is a generating matrix for $C^{\perp}$.
3. If C has length n and dimension k then $C^{\perp}$ has length n and dimension $n - k$.
4. If G is a $k \times n$ matrix then H is a $n \times (n - k)$ matrix.
5. $GH = 0$.

Theorem 2.2.12. Let C be a linear code of length n, and let H be a parity check matrix for C. Then C consists of the words $w \in K^n$ such that $wH = 0$.

Proof. Let H have columns $[x_1|x_2|\ldots|x_{n-k}]$ where $k = dim(c)$. Then, wH is the vector $(w \cdot x_1, \ldots, w \cdot x_{n-k})$. Since $\{x_1, \ldots, x_{n-k}\}$ is a basis for $C^{\perp}$, we have $w \in C \iff w \cdot x_i = 0$, $1 \leq i \leq n - k$. □

2.2.4 Deriving the generator and parity check matrices for a linear code

Let $C =< S >$. If G is a generator matrix for C and H is a parity check matrix for C, then we have the following situation:

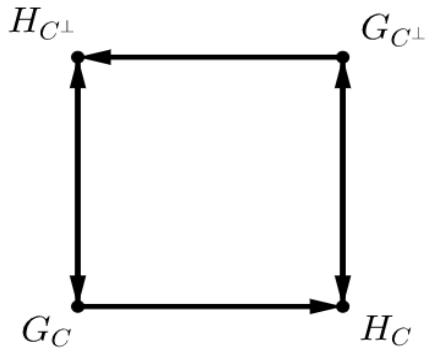


Figure 2.3: All four matrices can be derived from one of them.

1. First, $H_{C^\perp} \leftrightarrow G_C$ and $G_{C^\perp} \leftrightarrow H_C$ are achieved by the transposition.

2. To find a parity check matrix for $C =< S >$ (of length n and dim k). Recall, if we put the words of S in a matrix A then the rows of A span C (i.e., $\begin{bmatrix} I_k & X \\ 0 & 0 \end{bmatrix}$). Therefore $G = \begin{bmatrix} I_k | X \end{bmatrix}$ is a generator matrix for C.

3. Now, C has length n. Therefore, X has $n-k$ columns. Let $H = \begin{bmatrix} X \\ I_{n-k} \end{bmatrix}$, then H has $n-k$ linearly independent columns, and

$$GH = \begin{bmatrix} I_k | X \end{bmatrix} \begin{bmatrix} X \\ I_{n-k} \end{bmatrix} = I_k X + X I_{n-k} = X + X = 2X = 0.$$

   Therefore, the columns of H are in $C^\perp$. Since H has $n - k = dim(C^\perp)$ linearly independent columns, they form a basis for $C^\perp$. Therefore, H is a parity check matrix for C.

For example, let $S = \{11101, 10110, 01011, 11010\}$, $C =< S >$.

$$A = \begin{bmatrix} 11101 \\ 10110 \\ 01011 \\ 11010 \end{bmatrix} \rightarrow \begin{bmatrix} 11101 \\ 01011 \\ 00111 \\ 00000 \end{bmatrix} \rightarrow \begin{bmatrix} 100|01 \\ 010|11 \\ 001|11 \\ - - - \\ 000|00 \end{bmatrix} \rightarrow \begin{bmatrix} I_3 & X \\ 0 & 0 \end{bmatrix},$$

where $X = \begin{bmatrix} 01 \\ 11 \\ 11 \end{bmatrix}$. Therefore,

$$G = \begin{bmatrix} 100|01 \\ 010|11 \\ 001|11 \end{bmatrix}$$

is a generator matrix for C, and

$$H = \begin{bmatrix} 01 \\ 11 \\ 11 \\ 10 \\ 01 \end{bmatrix}$$

is a parity check matrix for C.

$$GH = \begin{bmatrix} 100|01 \\ 010|11 \\ 001|11 \end{bmatrix} \begin{bmatrix} 01 \\ 11 \\ 11 \\ 10 \\ 01 \end{bmatrix} = \begin{bmatrix} 00 \\ 00 \\ 00 \end{bmatrix} = 0.$$

2.2.5 equivalence of linear codes

Definition 2.2.13. Let C_1 and C_2 be codes of length n. Then C_1 and C_2 are **equivalent** if there exists a permutation $\pi \in S_n$ which when applied to the codewords on C_1 gives C_2.

For example, Let C_1 be the linear code with generator matrix $G_{C_1} = \begin{bmatrix} 100 \\ 001 \end{bmatrix}$. Then, $C_1 = \{000, 100, 001, 101\}$. Let C_2 be obtained from C_1 by exchanging the last 2 positions in every word. Then, $C_2 = \{000, 100, 010, 110\}$. Furthermore, $G_{C_2} = \begin{bmatrix} 100 \\ 010 \end{bmatrix}$. In otherwords, C_1 and C_2 are equivalent codes.

Theorem 2.2.14. Any linear code C_1 is equivalent to linear code C_2 having a generator matrix in standard form $G = [I_k|X]$.

Definition 2.2.15. A linear code with a generator matrix in standard, $G = [I_k|X]$, form is called **systematic**.

2.2.6 Encoding for linear codes

To encode a word $v = (\alpha_1, \ldots, \alpha_k)$, just $(\alpha_1, \ldots, \alpha_k)G = vG = [vI_k|vX]$, where vI_k are the information bits and vX are the parity check digits. If C is not systematic then the information bits and the parity check bits are intermixed. Therefore, if C is systematic then the first k bits of the encoded word are the bits in v (i.e., they are the data) and the remaining $(n-k)$ bits are there to help handle errors.

2.2.7 Decoding for linear codes

Recall, if $K = \{0, 1\}$ then $(K^n, +)$ is Abelian group, and a linear code $C \subseteq K^n$ is a subgroup of K^n. So, if $u \in K^n$, then $C + u = \{v + u \mid v \in C\}$ is the (right) coset determined by u. (Abelian, so right and left are the same).

For example, let $C = \{0000, 1011, 0101, 1110\}$. Then, the cosets of C are:

$$C + 0000 = \{0000, 1011, 0101, 1110\}$$

$$C + 1000 = \{1000, 0011, 1101, 0110\}$$

$$C + 0100 = \{0100, 1111, 0001, 1010\}$$

$$C + 0010 = \{0010, 1001, 0111, 1100\}$$

Theorem 2.2.16. Let $C \subseteq K^n$ be a linear code, and $u, v \in K^n$. Then,
1) $u \in C + v \implies C + u = C + v$
2)$u \in C + u$
3)$u + v \in C \implies C + u = C + v$
4)$u + v \notin C \implies C + u \neq C + v$
5) Either $C + u = C + v$ or $C + u \cap C + v = \emptyset$
6)$|C + u| = |C|$
7)$dim(C) = k \implies |C| = 2^k = |C + u| \forall u \in K^n \implies 2^{n-k}$ different cosets
8) $C = C + 0$ is a coset.

Recall: for Maximum likelihood decoding, if w is received, then if there exists a unique $u \in C$ such that $d(u, w)$ is minimum then decode w to u, otherwise: (1) Complete MLD breaks ties randomly, or (2)Incomplete MCD-request retransmission.

Let C be a linear code. Suppose $v \in C$ is transmitted and w received. The error pattern is $u = v + w$. Now, $w + u = w + v + w = v \in C$. Therefore, the error pattern and the received word are in the same coset. So, if you have w, then you can identify the coset which contains u. Note: error patterns of low weight are more likely than error patterns of high weight. Thus, we assume error patterns of small weight are more likely than errors of high weight. Therefore, choose as u a word of least weight in $C + w$, and decode to $w + u$.

For example, from above we have: $C = \{0000, 1011, 0101, 1110\}$, and the cosets of C are:

$$C + 0000 = \{0000, 1011, 0101, 1110\}$$

$$C + 1000 = \{1000, 0011, 1101, 0110\}$$

$$C + 0100 = \{0100, 1111, 0001, 1010\}$$

$$C + 0010 = \{0010, 1001, 0111, 1100\}$$

Suppose $w = 1101$ is received. Then, the error pattern is in $C + w$ and there is a unique choice for $u \in C + w$ of least weight, namely $u = 1000$. Therefore, decode as $u + w = 1000 + 1101 = 0101$.
Suppose $w = 1111$ received. The Coset $C + w$ has 2 words of least weight. So, if we use CMLD just choose one of 0100, 0001 as u and decode; if not, (i.e., IMLD) then request retransmission.

To decode efficiently we need to quickly:

1. Find the coset containing the received word.
2. Find a element of least weight in that coset.

Definition 2.2.17. Let C be a linear code of length n and dimension k, with a parity check matrix H. For any $w \in K^n$, the **syndrome** of w is the word wH. Note: $wH \in K^{n-k}$.

For the example above we have a Parity check matrix check matrix for C, $H = \begin{bmatrix} 11 \\ 01 \\ 10 \\ 01 \end{bmatrix}$.
The syndrome of $w = 1101$ is $wH = 11$. Let's compute xH for all $x \in C + w$.

$$C + w = \{1000, 0011, 1101, 0110\},$$

and all of them have syndrome 11.

Theorem 2.2.18. Let $C \subseteq K^n$ be a linear code with parity check matrix H. Then, for $u, w \in K^n$
a)$wH = 0 \iff w \in C$
b)$wH = uH \iff u, w$ are in the same coset of C.

Proof. a) Proved in theorem 2.2.12
b) We have

$$wH = uH \iff (w + u)H = wH + uH = 0 \iff w + u \in C \iff C + w = C + u.$$

□

Corollary 2.2.19. We can identify each coset by its syndrome. Note: if $dim(C) = k$ then there are 2^{n-k} cosets and 2^{n-k} syndromes.

Suppose u is the error pattern in a received word. Then uH is the sum of the rows of H that corresponds to the positions where errors occurred in transmission.

Definition 2.2.20. A **standard decoding array**, (SDA), is a table that matches each syndrome with a word of least weight in the corresponding coset. (If there is a tie then select randomly between them.)

For example, we have the following SDA for our example above

Syndrome	Coset leader
00	0000
01	0100 or 0001
10	0010
11	1000

To make a SDA:

1. Find the cosets.
2. Choose the coset leaders.
3. Match the coset leaders to the syndromes (or, equivalently just compute the syndrome for each coset leader).
4. (optional) sort the list.

2.3 Bounds for Codes

2.3.1 Bounds

Note: The following codes need not be linear unless stated.

Theorem 2.3.1. If $0 \leq t \leq n$ and $v \in K^n$, then the number of words u such that $d(u, v) \leq t$ is

$$[\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \binom{n}{3} + \ldots + \binom{n}{t}].$$

Theorem 2.3.2. *Hamming Bound:* If $C \subseteq K^n$ is a code with minimum distance $d \leq 2t + 1$ or $2t + 2$ then

$$|C| \cdot [\binom{n}{0} + \binom{n}{1} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}] \leq 2^n.$$

Proof. Since C has minimum distance d, no word $w \in K^n$ is distance at most t from 2 different codewords. For $v \in K^n$ define $B_t(v) = \{w \in K^n \mid d(v, w) \leq t\}$. By the argument above, every $w \in K^n$ belongs to at most one set $B_t(v), v \in C$. Therefore, $\sum_{v \in C} |B_t(v)| \leq |K^n| = 2^n$. By theorem 2.3.1, $|B_t(v)| = \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}$. Therefore,

$$|C| \cdot [\binom{n}{0} + \binom{n}{1} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}] = \sum_{v \in C} |B_t(v)| \leq 2^n.$$

□

1. The Hamming bound can be written as

$$|C| \leq \frac{2^n}{\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{t}}.$$

2. For equality to hold in the Hamming bound each word in the space needs to be contained in a ball of size t around some codeword. (i.e. the balls of size t partition the space).

For example, suppose C has length 6 and minimum distance $3 = 2 \cdot 1 + 1$. By the Hamming Bound

$$|C| \leq \frac{2^6}{\binom{6}{0} + \binom{6}{1}} = \frac{64}{7} = 9\frac{1}{7}.$$

Since $|C| \in \mathbb{Z}$, $|C| \leq 9$. If C is linear then $|C| = 2^k$. Therefore, $|C| \leq 2^3 = 8$ and $dim(C) \leq 3$.

Theorem 2.3.3. *Gilbert-Varshamov bound:* If $\binom{n-1}{0} + \binom{n-1}{1} + \binom{n-1}{2} + \ldots + \binom{n-1}{d-1} \leq 2^{n-k}$ then there exists a linear code of length n, dimension k and minimum distance d.

Proof. Suppose the inequality holds. We construct a parity check matrix H. We need to do this such that any $d-1$ row of H are linearly independent.

$$H = \begin{bmatrix} \\ \_ \_ \_ \_ \\ I_{n-k} \end{bmatrix}.$$

In the last $n-k$ rows of H put I_{n-k}. Suppose the last $t \leq n-1$ rows of H have been selected. Among the 2^{n-k} candidates for rows of H we can not select the zero row, any row already present, or any row that's a some of $2, 3, \ldots, d-2$ of the existing rows, as doing so would create a linearly dependent set of size $d-1$ row of H. Therefore, the number of possible rows we can't select is

$$\begin{aligned} &1 + \binom{t}{1} + \binom{t}{2} + \ldots + \binom{t}{d-2} \\ &= \binom{t}{0} + \binom{t}{1} + \binom{t}{2} + \ldots + \binom{t}{d-2} \\ &\leq \binom{n-1}{0} + \binom{n-1}{1} + \binom{n-1}{2} + \ldots + \binom{n-1}{d-2} \end{aligned}$$

where we've used: $\binom{r}{s} = 0$ if $s > r$, and if $p < q$ then $\binom{p}{s} \leq \binom{q}{s}$ for any S.

By hypothesis, the RHS $< 2^{n-k}$. Therefore, we can add another row to H. Therefore, By induction H can be completed. If at step n, H does not contain a set of d linearly independent rows, the last row can be chosen to be a sum of some $d-1$ rows of H. The columns of H are linearly independent by construction. Therefore, $C^{\perp}$ has dimension $n-k$ (columns of H are a basis for $C^{\perp}$. Therefore, C has $dim = k$ □

Corollary 2.3.4. If $1 \leq d \leq n$ then there exist a linear code of length n and minimum distance at least d with $|C| \geq \frac{2^{n-1}}{\binom{n-1}{0}+\binom{n-1}{1}+\binom{n-1}{2}+\ldots+\binom{n-1}{d-2}}$.

Proof. Let k be the largest integer such that $\binom{n-1}{0} + \binom{n-1}{1} + \binom{n-1}{2} + \ldots + \binom{n-1}{d-2} < 2^{n-k}$ Since the inequality holds for $k = 0$, $\binom{n-1}{d-1} + \binom{n-1}{d} + \binom{d+1}{2} + \ldots + \binom{n-1}{n-1}$ are missing from the LHS. Therefore, such a k exists. For the k there is a linear code with $|C| = 2^k$ and $2^k[\binom{n-1}{0} + \binom{n-1}{1} + \binom{n-1}{2} + \ldots + \binom{n-1}{d-2}] < 2^{n-k}2^k = 2^n$, or $2^k < \frac{2^n}{\binom{n-1}{0}+\binom{n-1}{1}+\binom{n-1}{2}+\ldots+\binom{n-1}{d-2}}$. Since k is largest, we have $\frac{2^n}{\binom{n-1}{0}+\binom{n-1}{1}+\binom{n-1}{2}+\ldots+\binom{n-1}{d-2}} \leq 2^{k-1}$ or $\frac{2^{n-1}}{\binom{n-1}{0}+\binom{n-1}{1}+\binom{n-1}{2}+\ldots+\binom{n-1}{d-2}} \leq 2^k = |C|$. □

Definition 2.3.5. An (n, k, d)**-code** is a linear code with length n, dimension k, and distance d.

Examples

1. Is there a (9,2,5)-code? By the Gilbert-Varshamov theorem, such a code exists if $\binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3} < 2^{9-2}$. $93 < 128$. Therefore, yes.

2. Is there a (15,7,5)-code. By Gilbert-Varshamov, such a code exists if $\binom{14}{0}+\binom{14}{1}+\binom{14}{2}+\binom{14}{3} < 2^{15-7}$. $470 < 256$. Therefore inconclusive by G-v. (Turns out, code exists).

3. Give bounds on the size and dimension of a linear code with length $n = 9$ and minimum distance $d = 5$. The Hamming bound says $|C| < \frac{2^9}{\binom{9}{0}+\binom{9}{1}+\binom{9}{2}} = \frac{512}{46} \approx 11.1$
   note: $d = 2 \cdot 2 + 1$ is where the 2 comes from.

   C is linear $\implies$ $|C| = 2^k$, $\implies$ $|C| \leq 8$ ($k \leq 3$) The Corollary says $|C| \geq \frac{2^8}{\binom{8}{0}+\binom{8}{1}+\binom{8}{2}+\binom{8}{3}} = \frac{256}{93}$ Therefore, $|C| \geq 4$. Therefore, $Dim(C)$ is 2 or 3. By the corollary such a code with $k = 2$ exists. What about $k = 3$? By G-v it exists if $93 = \binom{8}{0} + \binom{8}{1} + \binom{8}{2} + \binom{8}{3} < 2^{9-3} = 64$ Can't tell using G-v, But in the proof we used t is maximum thus can't be 3.

   Hamming Bound, length $d = 2t + 1$ or $2t + 2$. $|C| \leq \frac{2^n}{\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{t}}$

   Why can't d be even ??
   $2^4 = \binom{4}{0} + \binom{4}{1} + \binom{4}{2}$ $+ \binom{4}{3} + \binom{4}{4}$ first part is more than half!

2.3.2 Perfect Codes

Definition 2.3.6. A code C of length n and odd minimum distance $d = 2t + 1$ is called **perfect** if $|C| = \frac{2^n}{\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{t}}$. (i.e., equality holds in the hamming bound).

For example, the following are the trivial perfect codes.

1. k^n, $d = 2 \cdot 0 + 1$ $|K^n| = 2^n = \frac{2^n}{\binom{n}{0}} = \frac{2^n}{1}$.

2. $C = \{00\cdots0, 11\cdots\}$ of length $n = 2t + 1$.
   $|C| = 2 = \frac{2^n}{\binom{n}{0}+\binom{n}{1}+\ldots+\binom{n}{t}} = \frac{2^n}{2^{n-1}}$ since $\binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{t} = \binom{n}{n} + \binom{n}{n-1} + \ldots + \binom{n}{t+1}$.

Theorem 2.3.7. *Tietavairen and Van Lint, 1973:* If C is a non-trivial perfect code of length n and minimum distance $d = 2t + 1$, then either:
(i) $n = 2^r - 1$ for some $r \geq 2$ and $d = 3$ (Hamming codes)
(ii) $n = 23$ and $d = 7$ (Golay code)

Note:

1. If C is perfect and $d = 2t + 1$ then every $w \in K^n$ is within distance t of a codeword.

2. C corrects exactly the error patterns of weight $\leq t$.

3. A non-trivial perfect code is 1 or 3 error correcting.

2.4 Hamming Codes

Definition 2.4.1. Let $r \geq 2$ and H be the $(2^r - 1) \times r$ matrix whose rows are the non-zero binary words of length r. Then H has linearly independent columns. The linear code with parity check matrix H is the **Hamming code** of length $2^r - 1$.

For example, $(r = 3)$, $\left[\begin{array}{c} 111 \\ 110 \\ 101 \\ 011 \\ \hline 100 \\ 010 \\ 001 \end{array}\right]$.

From before, if H is in the form $H = \left[\begin{array}{c} X \\ \hline I_r \end{array}\right]$, then we can find a generator matrix $G = \left[I_{n-r}|X\right]$. Thus, the dimension of the Hamming code of length $n = 2^r - 1$ is $n - r = 2^r - 1 - r$. Therefore, the number of codewords is 2^{2^r-1-r}. Furthermore, no 2 rows of H are identical. Therefore, any set of 2 different rows of H are linearly independent. The rows of H are all non-zero binary words of length r. Thus, some 3 rows are linearly independent. Therefore, the minimum distance of a hamming code is 3.

Consider the Hamming Bound with $n = 2^r - 1$ and $d = 2 \cdot 1 + 1 = 3$.

$$2^{2^r-1-r} = |C| \leq \frac{2^{2^r-1}}{\binom{2^r-1}{0} + \binom{2^r-1}{1}} = \frac{2^{2^r-1}}{2^r} = 2^{2^r-1-r}.$$

Thus, equality holds. Therefore, Hamming codes are perfect! Hence, Hamming codes are perfect 1-error correcting codes.

It is Easy to make a SDA because the coset leaders are words of weight ≤ 1. Just compute the syndrome for each one.

For our example above, we have $(r = 3)$ and $H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}$ So, the SDA is

Syndrome	Coset Leader
000	00000000
111	01000000
110	00100000
101	00010000
011	00001000
100	00000100
010	00000010
001	00000001

.

Note: There is a better form of H. Consider the equivalent code whose parity check matrix is the set of non-zero binary words of length r in increasing numerical order.

For example, $H' = \begin{bmatrix} 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{bmatrix}$. Notice that:

Row of H	1	2	3	4	5	6	7
Row of H'	7	6	5	3	4	2	1

.

So, for $G = \begin{bmatrix} 1000|111 \\ 0100|110 \\ 0010|101 \\ 0001|011 \end{bmatrix}$. Apply the permutation above to the columns of G to get:

$$G' = \begin{bmatrix} 1101001 \\ 0101010 \\ 1001100 \\ 1110000 \end{bmatrix}$$

. and the SDA: $(r = 3)$

Syndrome	Coset Leader
000	00000000
001	01000000
010	00100000
011	00010000
100	00001000
101	00000100
110	00000010
111	00000001

.

So, the advantage of using H' and G' is that the syndrome corresponding to a coset leader (word in k^7 of weight 0 or 1) is either 0 or the row of H corresponding to the position where the leader has a 1. But the i th row of H' is the binary rep of i. Therefore, the syndrome is the binary rep of the position where the coset leader is a 1 (if it is not 0). This is the index of the bit to "flip" to decode the received word.

For example, suppose 0110010 received, then the syndrome is 111. Thus, we decode by flipping bit 7 (i.e., decode to to 0110011).

2.5 Extended Codes

2.5.1 Basics

Suppose C is a linear code of length n with generator matrix $G = \begin{bmatrix} I_k & | & X \end{bmatrix}$. The first k bits of the encoded word are data bits and the last $n-k$ bit are parity check bits.

For example, let

$$G = \begin{bmatrix} 1000|111 \\ 0100|110 \\ 0010|101 \\ 0001|011 \end{bmatrix}.$$

The 5th encoded bit is a 1 if and only if the first 3 data bits have an odd number of 1's.

Definition 2.5.1. Let C^* be obtained from C by adding an extra digit so that the number of 1's in each codeword is even. Then, C^* is the **extended code** of C. In other words, C^* is obtained from C by adding an extra digit so each codeword has even weight.

Suppose H is a parity check matrix for C. Then, $H^* = \begin{bmatrix} H & | & j \\ -- & & -- \\ 0000 & | & 1 \end{bmatrix}$, where j is all 1's, is a parity check matrix for C^*.

A generator matrix for C^* is obtained by applying the definition to the rows of the generator matrix for C. Therefore, $G^* = \begin{bmatrix} G|b \end{bmatrix}$, where b is chosen so that the number of 1's in each row is even.

For example, consider the Hamming code of length 7.

$$H^* = \begin{bmatrix} 111|1 \\ 110|1 \\ 101|1 \\ 011|1 \\ 100|1 \\ 010|1 \\ 001|1 \\ - - - - - \\ 000|1 \end{bmatrix} \quad \text{and} \quad G^* = \begin{bmatrix} 1000111|0 \\ 0100110|1 \\ 0010101|1 \\ 0001011|1 \end{bmatrix}.$$

2.5.2 Benefits

Suppose v is a word in C and v^* is the corresponding word in C^*, then $wt(v^*)$ equals:

1. $wt(v)$ if $wt(v)$ is even, or
2. $wt(v) + 1$ if $wt(v)$ is odd.

Therefore, if C has even minimum distance d (i.e., the smallest weight of a nonzero codeword), then C^* also has minimum distance d. If C has odd minimum distance d then C^* has minimum distance $d + 1$. Hence, if the minimum distance of C is odd, then C^* detects one more error than C (it corrects no more). Therefore, extending a code C is only useful if the minimum distance of C is odd.

For example, the Extended Hamming code in the example above becomes length 8 code with minimum distance 4 (it was 3 in Hamming code).

Curiosity, If C^* is the extended Hamming code of length 8, then $C^* = (C^*)^{\perp}$. Look at the dimemsions of the matrices, $G^*(G^*)^T = 0$ Therefore $(G^*)^T$ is a parity check matrix for C^*.

2.6 The Golay Code

2.6.1 The Extended Golay Code

Facts about the Extended Golay Code:

1. The extended Golay Code is a $(24, 12, 8)$-code.
2. The following is Golay's original construction from 1949.
3. It was used in the Voyager space program.
4. The construction is done in three steps.

Definition 2.6.1. A **circulent matrix** is a matrix in which each row is obtained from the first by cyclically rotating the first row.

For example,

$$G = \begin{bmatrix} 011000 \\ 00110 \\ 00011 \\ 10001 \\ 11000 \end{bmatrix}$$

is a five cycle, C_5, circulent matrix.

The construction:

1. Let B_1 be the 11x11 matrix with 1st row 11011100010 and each subsequent row a cyclic shift (left) of the one above. (i.e., B_1 is a circulent matrix with 1st row 11011100010.)

$$B_1 = \begin{bmatrix} 11011100010 \\ 10111000101 \\ 01110001011 \\ 11100010110 \\ 11000101101 \\ 10001011011 \\ 00010110111 \\ 00101101110 \\ 01011011100 \\ 10110111000 \\ 01101110001 \end{bmatrix}.$$

2. Let B be the matrix $B = \begin{bmatrix} B_1 & | & j \\ -- & & -- \\ j^T & | & 0 \end{bmatrix}$

   Observe:

   (a) $B = B^T$ (i.e. B is symmetric).

   (b) $BB^T = BB = I$ (i.e., the rows of B are $\perp$.)

Definition 2.6.2. The **extended Golay code** C_{24} is the linear code with generator matrix

$$\left[I_{12}|B_{12\times 12}\right].$$

Note: the length of C_{24} is 24 and the dimension is 12. Thus, there are $2^{12} = 4096$ codewords.

The parity check matrix for C_{24} is $H = \begin{bmatrix} B \\ I \end{bmatrix}$ since $GH = \left[B|I\right] \begin{bmatrix} B \\ I \end{bmatrix} = B + B = 0$.

Another parity check matrix for C_{24} is $H' = \begin{bmatrix} I \\ B \end{bmatrix}$ since $GH' = I^2 + B^2 = I + I = 0$. Thus, another Generator matrix is $G' = \begin{bmatrix} B & I \end{bmatrix}$ since $G'H' = 0$ (i.e., C_24 is a self-dual code).

Theorem 2.6.3. Supposed C is a linear code of even length n and dimension $\frac{n}{2}$. If C has a basis in which any two 2 vectors are orthogonal then $C = C^{\perp}$.

Theorem 2.6.4. C_{24} has minimum distance 8.

Proof. The following three claims prove the theorem.

1. Claim: The weight of every codeword is a multiple of 4.

   Proof. This is true for words that are rows of G since each one has 8 ones.

   Suppose, for some $k \geq 1$, that the weight of any linear combination of k or fewer rows of G has weight a multiple of 4. Let $v \in C_{24}$ be a linear combination of $k+1$ rows of G. Then, $v = r+w$, where r is a row of G and w is a linear combination of k rows of G. We know $4 \mid wt(r)$ since r is a row of G and $4 \mid wt(w)$ by the induction hypothesis. Since any 2 rows of G are orthogonal, any 2 rows have an even number of 1's in common. (i.e., the dot product needs to be even, (0 mod2), and this only happens if they have an even number of 1's in common). Now, if $w = v_1 + v_2 + \ldots v_k$ is odd then one of the summons has to be odd if w is odd. Therefore, r and w have an even number of 1's in common, say $2x$. Thus, $wt(v) = wt(r + w) = wt(r) + wt(w) - 2(2x)$. Each summon is a multiple of 4. Therefore $4|wt(v)$. □

2. Claim: If every codeword is a multiple of 4 then the minimum distance is 4 or 8.

Proof. Since there are code words of weight 8, the weight of every codeword is a multiple of 4, and there are nonzero codewords; the word with least weight has either weight 4 or 8. Therefore, the minimum distance is 4 or 8 since the minimum distance is equal to the least weight of a nonzero codeword. □

3. Claim: There are no codewords of weight 4.

Proof. Suppose $v \in C_{24}$ and $wt(v) = 4$. Since $G = \left[I_{12}|B\right]$ and $G' = \left[B|I_{12}\right]$ are both generator matices for C_{24}, there exist w_1 and w_2 such that $v = w_1 \cdot G = w_2 \cdot G'$. (Note: $wt(v) = 4 \implies w_1, w_2 \neq 0$). Since at least one half of v must contain at most 2 ones, either $wt(w_1) \leq 2$ or $wt(w_2) \leq 2$. But no sum of one or two rows of B has weight less than or equal to 3. Therefore, $wt(v) = wt(w_i) + wt(w_i B) > 2 + 4 > 4$. Therefore, there is no $v \in C_{24}$ has weight 4. □

Therefore, C_{24} has minimum distance 8. □

2.6.2 The Golay Code

Definition 2.6.5. Puncturing a code means removing the same digit from each codeword.

Definition 2.6.6. The **Golay code** is obtained from the extended Golay code by puncturing (deleting) the last digit from each word in C_{24}.

Therefore, a generator matrix for this code is $G = \left[I_{12}|B'\right]$, where $B' = \begin{bmatrix} B_1 \\ j^T \end{bmatrix}$. G is a 12×23 matrix and has linearly independent rows. Therefore C_{23} has dimension 12 and $2^{12} = 4096$ codewords. Now, clearly, C_{24} is the extended code of C_{23}. Furthermore, since all nonzero codewords in C_{24} have weight greater than or equal to 8, C_{23} has minimum distance 7. Thus, C_{23} is a $(23, 12, 7)$-code. It is perfect and it corrects all error patterns of weight 3 and no others. (i.e. Every word in K^{23} is within distance 3 of exactly one word in C_{23}).

2.7 Reed-Muller Codes

Definition 2.7.1. The rth **Reed-Muller** code of length 2^m, denoted $RM(r, m)$, $0 \leq r \leq m$, is defined recursively as:
(1) $RM(0, m) = \{00\ldots0, 11\ldots1\}$ and $RM(m, m) = K^{2^m}$.
(2) $RM(r, m) = \{x \oplus (x+y) \mid x \in RM(r, m-1), y \in RM(r-1, m-1)\}$ where $\oplus$ is concatenation.

For example, we have:

1. $RM(0, 0) = \{0, 1\}$
2. $RM(0, 1) = \{00, 11\}$
3. $RM(1, 1) = \{00, 01, 10, 11\} = K^2$
4. $RM(0, 2) = \{0000, 1111\}$
5. $RM(1, 2) = \{x \oplus (x + y) | x \in RM(1, 1), y \in RM(0, 1)\}$
   $= \{00|00, 00|11, 01|01, 01|10, 10|10, 10|01, 11|11, 11|00\}$
   $= \{0000, 0011, 0101, 0110, 1010, 1001, 1111, 1100\}$
6. $RM(2, 2) = K^{2^2} = K^4$

Definition 2.7.2. Let $G(r, m)$ be the generator matrix for $RM(r, m)$. It is defined recursively as:
(1) $G(0, m) = \begin{bmatrix} 11\ldots1 \end{bmatrix}$.
(2) For $0 < r < m$, $G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}$.
(3) $G(m, m) = \begin{bmatrix} G(m-1, m) \\ 00\ldots01 \end{bmatrix}$.

For example, we have:

1. $G(0, 1) = \begin{bmatrix} 11 \end{bmatrix}$
2. $G(1, 1) = \begin{bmatrix} 11 \\ 01 \end{bmatrix}$
3. $G(1, 2) = \begin{bmatrix} G(1, 1) & G(1, 1) \\ 0 & G(0, 1) \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \end{bmatrix}$
4. $G(2, 2) = \begin{bmatrix} G(1, 2) \\ 0001 \end{bmatrix} = \begin{bmatrix} 1111 \\ 0101 \\ 0011 \\ 0001 \end{bmatrix}$

5. $G(1,3) = \begin{bmatrix} G(1,2) & G(1,2) \\ 0 & G(0,2) \end{bmatrix} = \begin{bmatrix} 1111|1111 \\ 0101|0101 \\ 0011|0011 \\ 0000|1111 \end{bmatrix}$

Theorem 2.7.3. $RM(r,m)$ has the following properties.
(1) length 2^m
(2) minimum distance 2^{m-r}
(3) dimension $\sum_{i=0}^{r} \binom{m}{i}$
(4) for $r > 0$, $RM(r-1,m) \subseteq RM(r,m)$
(5) for $r < m$, $RM(r,m)^{\perp} = RM(m-1-r,m)$.

Recall:

$0 \leq r \leq m$. $RM(0,m) = \{000\ldots, 111\ldots 1\}$ where the are 2^m 0's and 1's respectively. $RM(m,m) = K^{2m}$ $RM(r,m) = \{x \oplus (x+y), x \in RM(r,m-1), y \in RM(r-1,m-1)\}$, where $0 < r < m$.

Theorem: $RM(r,m)$ has the following properties. (1) length 2^m
(2) minimum distance 2^{m-r}
(3) dimension $\sum_{i=0}^{r} \binom{m}{i}$
(4) for $r > 0$, $RM(r-1,m) \subseteq RM(r,m)$
(5) for $r < m$, $RM(r,m)^{\perp} = RM(m-1-r,m)$.

end recall

Proof: (1) Easy by induction

(4) (Induction on $r+m$.)
Since $RM(0,1) \subseteq RM(1,1)$, the statement is true if $r+m \leq 2$. Assume that if $r+m < t$ (and $r > 0$) the $RM(r-1,m) \subseteq RM(r,m)$. Suppose $r+m = t$. Then, there are several cases.
Case 1: $(r = 1)$ (want $RM(0,m) \subseteq RM(1,m)$). From the recursive definition of $G(r,m)$, the first row of $G(r,m)$ is all 1's, since (come up with a reason!!!!!). Since $G(0,m) = [11\ldots 1]$ we have that $G(0,m)$ is submatrix of $G(1,m)$. Therefore, $RM(0,m) \subseteq RM(1,m)$.
Case 2: $(r = m)$. We know $RM(m,n) = k^{2^m}$. Therefore, $RM(m-1,m) \subseteq RM(m,m)$.
Case 3: $(0 < r < m)$. We have $G(r-1,m) = \begin{bmatrix} G(r-1,m-1) & G(r-1,m-1) \\ 0 & G(r-2,m-1) \end{bmatrix}$, and $G(r,m) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix}$. By hypothesis every row of $G(r-1,m-1)$ is a row of $G(r,m-1)$, and every row of $G(r-2,m-1)$ is a row of $G(r-1,m-1)$. Therefore, from the construction, every row of $G(r-1,m)$ is a row of $G(r,m)$. Therefore, $RM(r-1,m) \subseteq RM(r,m)$.

(2) (induction on $m+r$). It is clear if $m+r = 0$ or if $m+r = 1$. Suppose that if $s+t < l$, then the minimum distance of $RM(s,t) = 2^{t-s}$. Consider the case where $r+m = l$. If $r = 0$ we are good. If $r = m$ we are good tooSuppose $0 < r < m$. Then,

$RM(r,m) = \{x \oplus (x+y) \mid x \in RM(r,m-1), y \in RM(r-1,m-1)\}$, and we know $RM(r-1,m-1) \subseteq RM(r,m-1)$. Therefore, $x+y \in RM(r,m-1)$. If $x \neq y$ then, by IH, $wt(x+y) \geq 2^{m-1-r}$ and $wt(x) \leq 2^{m-1-r}$. Therefore, $wt(x \oplus (x+y)) = wt(x) + wt(x+y) \geq 2^{m-1-r} + 2^{m-1-r} = 2^{m-r}$. If $x = y$ then $x + y = 0$ and $x \in RM(r-1,m-1)$. Therefore, $wt(x \oplus (x+y)) = wt(x) \leq 2^{(m-1)-(r-1)} = 2^{m-r}$. Therefore minimum distance $d \geq 2^{m-r}$. Equality can be seen to hod by choosing $x = y \in RM(r-1,m-1)$ so that equality holds in the above argument.

(3) and (5) are homework.

By the Theorem, $RM(1,m)$ is a $(2^m, m+1, 2^{m-1})$-code. So it corrects a lot of errors.

Ex. $RM(1,3)$ is an (8,4,4)-code. the generator matrix is $G(1,3) = \begin{bmatrix} 1111\ 1111 \\ 0101\ 0101 \\ 0011\ 0011 \\ 0000\ 1111 \end{bmatrix}$. Suppose w is received. If we find $v \in RM(1,3)$ such that $d(v,w) < 2$, then (since d=4) we decode w to v. Suppose $d(v,w) > 6$, then $d(v + 11\ldots1, w) \leq 1$, $11\ldots1 \in RM(1,3)$. Therefore, $v + 11\ldots1$ is a codeword, since $11\ldots1 \in RM(1,3)$ Notice, $d(v,w) \leq 1 \iff d(v+11\ldots1, w) \geq 7$. Therefore, $RM(1,m)$ can be decoded by examining at most half the codewords.

2.8 Decimal Codes

2.8.1 ISBN

ISBN stands for "International Standard Book Number. " It is a ten digit number, $x_1x_2, \ldots x_{10}$, with several parts. The first group of numbers is a *Group Identifier* (i.e., it identifies a country or group of countries where the book was published). For example, $x_1 = 0$ means the book was published in USA, canada, UK, or Australia; $x_1 = 3$ indicates publication in German; and $x_1x_2 = 87$ indicates publication in Denmark. The second group of number is the *Publisher Prefix*. It is 2 to 7 digits long and identifies the publisher. For example, 13 is prentice-hall. The third is the *Tittle Prefix* or Title number. It is 1 to 6 digits long and is assigned by publisher.

The Group Identifier, Publisher Prefix, and Tittle Prefix is always a total of 9 digits. The last digit, x_{10}, is a Parity check digit chosen such that :

$$1 \cdot x_1 + 2 \cdot x_2 + \ldots + 10 \cdot x_{10} \equiv 0 \pmod{11}$$

or equivalently

$$1 \cdot x_1 + 2 \cdot x_2 + \ldots + 9 \cdot x_9 \equiv -10x_{10} \equiv x_{10} \pmod{11}.$$

Note: If x_{10} is 10 then we write x (i.e., we need a symbol for ten in (mod 11)).

For example, is 0-201-34292-8 a valid ISBN? We compute

$$(1 \cdot 0) + (2 \cdot 2) + (3 \cdot 0) + (4 \cdot 1) + (5 \cdot 3) + (6 \cdot 4) + (7 \cdot 2) + (8 \cdot 9) + (9 \cdot 2) \equiv 30 \equiv 8 \pmod{11}.$$

Therefore, yes!

2.8.2 Properties of ISBN

1. The ISBN detects all single errors. Suppose a single error occurs and x_i is received as $x_i + e$ where $e \not\equiv 0 \pmod{11}$. The sum becomes

   $$1 \cdot x_1 + 2 \cdot x_2 + \ldots + i(x_i + e) + \ldots + 10 \cdot x_{10} \equiv 0 + ie \pmod{11}.$$

   Now, we have $i \cdot e \equiv 0 \pmod{11}$ if and only if $11|i \cdot e$. Hence, $11|i$ or $11|e$, which is false. Therefore, it detects all single errors.

2. The ISBN can correct a single error if it is known which digit is in error. For example, find x if $01383x0943$ is an ISBN.

   $$(1{\cdot}0)+(2{\cdot}1)+(3{\cdot}3)+(4{\cdot}8)+(5{\cdot}3)+(6{\cdot}x)+(7{\cdot}0)+(8{\cdot}9)+(9{\cdot}4)+(10{\cdot}3) \equiv (6{\cdot}x)+9 \pmod{11}.$$

   To get a valid ISBN we need $(6 \cdot x) + 9 \equiv 0 \pmod{11}$. Thus, $(6 \cdot x) \equiv -9 \equiv 2 \pmod{11}$. Hence, $x \equiv 4 \pmod{11}$ since 2 is the unique reciprocal of 6 (mod 11).

3. The ISBN detects any error arising from transposing 2 digits. Suppose $x_1x_2\ldots x_{10}$ is a valid ISBN. Assume x_i and x_j get exchanged where $i < j$ and $x_i \neq x_j$. The check sum equation becomes

   $1{\cdot}x_1{+}2{\cdot}x_2{+}\ldots{+}i{\cdot}x_i{+}\ldots{+}j{\cdot}x_j{+}\ldots 10{\cdot}x_{10} \equiv [1{\cdot}x_1{+}2{\cdot}x_2{+}\ldots{+}10{\cdot}x_{10}]{+}i(x_j{-}x_i){+}j(x_i{-}x_j) \pmod{11}$.

   $\equiv 0 + i(x_j - x_i) - j(x_i - x_j) \equiv (i-j)(x_j - x_i) \not\equiv 0(mod 11)$ since $(i-j) \neq 0$ and $(x_j - x_i) \neq 0$.

2.8.3 Single Error Correcting Decimal Code of length 10

We want a single error correcting decimal code. Recall: If u and v are words in any code then the distance between u and v, denoted $d(u, v)$, is the number of places where u and v differ.

Theorem 2.8.1. Let C be the set of 10 decimal digits numbers (i.e., $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$), where $x_1 \ldots x_8$ are information digits and x_9 and x_{10} are parity check digits chosen so that

$$S_1 = 1 \cdot x_1 + 2 \cdot x_2 + \ldots + 10 \cdot x_{10} \equiv 0 \pmod{11}$$

and

$$S_2 = x_1 + x_2 + \ldots + x_{10} \equiv 0 \pmod{11},$$

then C is a single error correcting decimal code with 10^8 possible code words.

For example, let $x_1x_2x_3x_4x_5x_6x_7x_8 = 02013429$ and choose x_9 and x_{10} such that S_1 and S_2 are satisfied. Notice, $10 \equiv -1 \pmod{11}$ so adding S_1 and S_2, x_{10} drops out. Thus, we can find x_9 uniquely then x_{10} uniquely . So,

$$1 + 9x_9 + 10x_{10} \equiv 0 \pmod{11}$$

and

$$10 + x_9 + x_{10} \equiv 0 \pmod{11}.$$

Therefore, $x_9 = 0$ and $x_{10} = 1$.

Finding parity check digits

In general, if

$$9x_9 + 10x_{10} \equiv a \pmod{11} \quad \text{and} \quad x_9 + x_{10} \equiv b \pmod{11}$$

then $10x_9 \equiv a + b \pmod{11}$. Therefore,

$$(1) \quad x_9 \equiv 10(a+b) \pmod{11}$$

and x_{10} is determined by $10(a+b) + x_{10} \equiv b \pmod{11}$, which is equivalent to

$$(2) \quad x_{10} \equiv b - 10a - 10b \equiv a + 2b \pmod{11}.$$

Single error correcting

Suppose some codeword $x_1x_2\ldots x_{10}$ is sent and an error $e \not\equiv 0$ occurs in the ith digit. The received word is $x_1x_2\ldots x_{i-1}(x_i+e)x_{i+1}\ldots x_{10}$ Therefore, $S_1 = 1\cdot x_1 + 2\cdot x_2 + \ldots + 10\cdot x_{10} + i\cdot e \pmod{11}$ and $S_2 = x_1 + x_2 + \ldots + x_{10} + e \pmod{11}$, $= 0 + e \pmod{11}$

2nd equation gives the magnitude of e. Use 1st equation t find the position i

$S_1 \equiv 0 + ie \pmod{11} \equiv iS_2 \pmod{11}$

Therefore $i \equiv S_1S_2^{-1} \pmod{11}$.

Decoding

Suppose received word is r compute $r\cdot H = \begin{bmatrix}S_1S_2\end{bmatrix}$ where $H = \begin{bmatrix}1 & 1\\ 2 & 1\\ 3 & 1\\ 4 & 1\\ 5 & 1\\ 6 & 1\\ 7 & 1\\ 8 & 1\\ 9 & 1\\ 10 & 1\end{bmatrix}$.

If $S_1 = S_2 = 0$, r is a codeword.

If $S_1 \neq 0$ and $S_2 \neq 0$, then assume an error of magnitude $\equiv S_2$ in position $S_1S_2^{-1} \equiv i(mod 11)$ and decode r as $x_1x_2\ldots x_{i-1}(x_i - e)x_{i+1}\ldots x_{10}$

If $S_1 \equiv 0$ or $S_2 \equiv 0$ (not both) then at least 2 errors have occurred so repeat transmission.

What happens if x_i and x_j are transposed? Get $S_2 \equiv 0$ and $S_1 \equiv 0$ (as before) Therefore can detect all errors arising from transposing 2 digits.

2.8.4 2-Error Correcting Decimal Code of length 10

Theorem 2.8.2. Let C be the set of 10 decimal digits numbers (i.e., $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$), where $x_1x_2x_3x_4x_5x_6$ are information digits, and $x_7x_8x_9x_{10}$ are parity check digits chosen to satisfy

$$S_1 = \sum_{i=1}^{10} ix_i \equiv 0(mod 11),$$

$$S_2 = \sum_{i=1}^{10} x_i \equiv 0(mod 11),$$

$$S_3 = \sum_{i=1}^{10} i^2x_i \equiv 0(mod 11),$$

and

$$S_4 = \sum_{i=1}^{10} i^3x_i \equiv 0(mod 11).$$

Then, C is a 2-Error Correcting Decimal Code of length 10.

Proof. Suppose $x_1x_2x_3x_4x_5x_6x_7x_8x_9x_{10}$ is sent and errors $e_i \neq 0$ and $e_j \neq 0$ occur in positions i and j, respectively. Then, from the check equations we have:

$$\begin{array}{cc} S_1 \equiv ie_i + je_j \pmod{11}, & S_2 \equiv e_i + e_j \pmod{11}, \\ S_3 \equiv i^2e_i + j^2e_j \pmod{11}, & S_4 \equiv i^3e_i + j^3e_j \pmod{11}. \end{array}$$

Solving these four equations for the four unknowns i, j, e_i, e_j we get that i and j are the roots of $ax^2 + bx + c = 0$ (in $\mathbb{Z}_{11}$) where

$$a = S_1^2 - S_2S_3 \qquad b = S_2S_4 - S_1S_3 \qquad c = S_3^2 - S_1S_4.$$

Thus, using the quadratic formula in Z_{11} to find i and j, then finding e_i and e_j by solving

$$S_1 \equiv ie_i + je_j \pmod{11} \quad \text{and} \quad S_2 \equiv e_i + e_j \pmod{11}.$$

So, we can conclude that the parity check matrix is

$$H = \begin{bmatrix} 1 & 1 & 1^2 & 1^3 \\ 2 & 1 & 2^2 & 2^3 \\ \cdots & & & \\ 10 & 1 & 10^2 & 10^3 \end{bmatrix}$$

and decoding is done as follows. Suppose r is the received word. Compute $rH = \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} = S$ (i.e., the syndrome). Then:

1. If $S = 0$, r is a codeword.
2. If $S \neq 0$ and $a = b = c = 0$. Then, $e_j = 0$, $S_1 = ie_i$, $S_2 = e_i$, $S_3 = i^2e_i$, $S_4 = i^3e_i$. compute a, b, c..... Thus, assume a single error has occurred in digit $i = S_1S_2^{-1}(mod11)$ and has size $e_i = S_2$.
3. If $S \neq 0$ and $a \neq 0$, and $b^2 - 4ac$ is a square in Z_11, then find i, j, e_i, e_j as before and correct.
4. If $b^2 - 4ac$ is not a square, then more than two errors occurred, request retransmission.

□

2.8.5 *UPC*

The universal product code, UPC, is a decimal code of length 12, (i.e., words $x_1x_2 \dots x_{12}$), where x_1 is the product type, $x_2x_3 \dots x_6$ signifies the manufacturer, $x_7 \dots x_11$ is the product number, and x_{12} is the check digit such that

$$3(x_1 + x_3 + x_5 + x_7 + x_9 + x_{11}) + (x_2 + x_4 + x_6 + x_8 + x_{12}) \equiv 0 \text{ (mod10)}.$$

For example, the UPC $53800051150x_{12}$ is "50 cents off kellogg's rice-crispies". Thus, $x_{12} = 3(5+8+0+5+1+0) + (3+0+0+1+5) \equiv 6 \text{ (mod10)}$. So, $x_{12} = -6 \equiv 4 \text{ (mod10)}$.

Suppose an error of size $e \neq 0$ occurs in digit i. If i is even then the sum changes by e. If i is odd the the sum changes by $3e$. Now, $gcd(3, 10) = 1$, $10|3e \iff 10|e$, and $0 \leq e \leq 9$. So, $10 \not| e$. Therefore, if an error occurs then the sum $\neq 0$ (mod 10). Therefore, e is detected.

If adjacent digits x_i and x_j transposed, then the change to sum is

$$-3x_i + x_i - x_j + 3x_j = 2(x_j - x_1),$$

which is not detect if $2(x_j - x_1) \equiv 0 \text{ (mod10)}$.

2.8.6 US money orders

In the code for US money orders codewords have length 11 (i.e., $x_1x_2x_3 \dots x_{11}$), where x_{11} is a check digit satisfying $x_1x_2 \dots x_{10} \equiv x_{11} \text{ (mod 9)}$ and $0 \leq x_{11} \leq 8$. (Note: $x_1x_2 \dots x_{10} \equiv x_{11} \text{ (mod 9)}$ if and only if $x_1 + x_2 + \dots + x_{10} \equiv x_{11} \text{ (mod 9)}$.)

If an error in x_{11} occurs then it is detected. If an error in $x_1 \dots x_{10}$ occurs then it is detected unless it is changed from a $0 \to 9$ or $9 \to 0$.

The probability that the ith digit is 0 (or 9) is 0.1 Suppose ith digit is 0. Then, the probability that it becomes a 9 is 1/9. So, the probability digit i get changed from a $0 \to 9$ is 1/90. Similarly, the probability digit i get changed from a $9 \to 0$ in digit i is 1/90. Now, the probability an error occurs in the 1st 10 digits (given that an error occurs) is 10/11. Therefore, the probability an error is undetected is $10/11 \times 2/90$.

2.8.7 US postal code

The codewords of the US postal code are $x_1x_2\ldots x_{10}$ bar codes where each decimal digit is represented by a group of 5 vertical bars. Each group of 5 vertical bars has 2 long bars and 3 short ones. x_{10} is such that $x_1+x_2+\ldots+x_{10}\equiv 0\ (\text{mod}10)$ (Note $\binom{5}{2}=10$ arrangements.)

Write 1 for a long bar and 0 for a short bar. Then, each group of 5 is a binary sequence $abcde$ and represents the number $7a+4b+2c+d\ (\text{mod } 11)$. Thus, choosing the following correspondence:
$$\begin{array}{lllll} 00011\to 1 & 00101\to 2 & 00110\to 3 & 01001\to 4 & 01010\to 5 \\ 01100\to 6 & 10001\to 7 & 10010\to 8 & 10100\to 9 & 11000\to 0 \end{array}$$

For example, $|ii|i$ represent 8.

errors

The scanner could read | as i or i as |, but this is always detected because then there would be the wrong number of |'s. $0\to 1$ or $1\to 0$ are the only single errors that can occur. If the ith block has an error, then the number is undefined. But, that number can be recovered from the check equation.

2.9 Cyclic Codes

Definition 2.9.1. A k-dimensional linear subspace C of F_q^n is called a *linear code* or a $[n,k]$ code over the field F_q, where q is a prime.

Most of the material we have covered so far holds for $[n,k]$-codes with slight changes. Suppose for now on that $\gcd(n,q)=1$.

Definition 2.9.2. An $[n,k]$ code C over F_q is *cyclic* if

$$\forall_{c_0,c_1,\ldots,c_{n-1}\in C}[(c_{n-1},c_0,c_1,\ldots,c_{n-2})\in C].$$

Now, from algebra we know $R/S=\frac{F_q[x]}{<(x^n-1)>}$ is isomorphic to F_q^n. The natural isomorphism is given by

$$(a_0,a_1,\ldots,a_{n-1})\to a_0+a_1x+\ldots+a_{n-1}x^{n-1}.$$

So, there is a natural correspondence between words and polynomials in R/S, where multiplication by x corresponds to the cyclic shift

$$(a_0,a_1,\ldots,a_{n-1})\to(a_{n-1},a_0,\ldots,a_{n-2}).$$

Note: the powers of x are modulo n. So, from algebra again, every cyclic code in F_q^n corresponds to a principle ideal generated by some polynomial $f(x)$ that divides x^n-1.

Definition 2.9.3. $f(x)$ is called the *generator polynomial* of the cyclic code.

Theorem 2.9.4. Let $f(x) = a_0 + a_1x + \ldots + a_{n-k}x^{n-k}$ be the generator polynomial of the cyclic code C and let $g(x) = \frac{x^n-1}{f(x)} = g_0 + g_1x + \ldots + h_kx^k$. Then, $f(x), xf(x), \ldots, x^{k-1}f(x)$ form a basis of the code C, that is

$$G = \begin{bmatrix} a_0 & a_1 & \ldots & a_{n-k} & 0 & 0 & \ldots & 0 \\ 0 & a_0 & a_1 & \ldots & a_{n-k} & 0 & \ldots & 0 \\ \ddots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ddots \\ 0 & 0 & \ldots & 0 & a_0 & a_1 & \ldots & a_{n-k} \end{bmatrix}$$

Furthermore, since $f(x)g(x) = 0$ in the ring R/S we have that:

$$H = \begin{bmatrix} 0 & 0 & \ldots & 0 & g_k & g_{k-1} & \ldots & g_0 \\ 0 & \ldots & 0 & g_k & g_{k-1} & \ldots & g_0 & 0 \\ \ddots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ddots \\ g_k & g_{k-1} & \ldots & g_0 & 0 & \ldots & 0 & 0 \end{bmatrix}$$

is a parity check matrix for C.

Theorem 2.9.5. The binary cyclic code of length $n = 2^r - 1$ for which the generator polynomial is the minimal polynomial of a primitive element of F_{2^r} is equivalent to the $[n, n-m]$ binary Hamming code.

For example, if $q = 2$, $n = 7$, and $g(x) = 1 + x + x^3$. Then, the cyclic code C generated by $g(x)$ has dimension 4 and is equivalent to the $[7, 4]$ Hamming code.

2.10 Assignment 1

1. Let C be a binary code of length n. We define the information rate of C to be the number $i(C) = \frac{1}{n}\log_2(|C|)$. This quantity is a measure of the proportion of each codeword that is carrying the message, as opposed to redundancy that has been added to help deal with errors.

   The *reliability* of a binary symmetric channel (BSC) is the probability, p, that the digit sent is the digit received. Suppose you are using a BSC with reliability $p = 1 - 10^{-8}$ and that digits are transmitted at the rate 10^7 digits per second.

   (a) Suppose the code C contains all of the binary words of length 11.

      i. What is the information rate of C?
      Solution:
      $$\frac{1}{11}\log_2(2^{11}) = \frac{11}{11}\log_2(2) = 1.$$

      ii. What is the probability that a word is changed during transmission and the errors are not detected?
      Solution: The probability that a word is changed during transmission and the errors are not detected, call it P, is just the probability that a word is changed since the information rate is 1. Thus,
      $$P = 1 - \binom{11}{0}(1 - 10^{-8})^{11}(10^{-8})^0 = 1 - (1 - 10^{-8})^{11} \approx 1.0\bar{9} \times 10^{-7}.$$

      iii. If the channel is in constant use, about how many words are (probably) transmitted incorrectly each day?
      Solution: If the channel is in constant use, then approximately
      $$\frac{(60^2 \cdot 24 \cdot 10^7) \cdot 1.0\bar{9} \times 10^{-7}}{11} \approx 8640 \text{ words each day are transmitted incorrectly.}$$

   (b) Now suppose the code C' is obtained form C by adding an extra (parity check) digit to the words in C, so that the number of 1's in each codeword is even.

      i. What is the information rate of C'?
      Solution:
      $$\frac{1}{12}\log_2(2^{11}) = \frac{11}{12}\log_2(2) = \frac{11}{12}.$$

      ii. What is the probability that a word is transmitted incorrectly and the transmission errors go undetected?
      Solution: The probability that a word is transmitted incorrectly and the transmission errors go undetected, P, is the probability that an even number of errors occur since the parity check bit only works if an odd amount of errors occur in a word. Thus,
      $$P = \sum_{2,4,6,8,10,12} \binom{12}{i}(1 - 10^{-8})^{12-i}(10^{-8})^i \approx 6.59 \times 10^{-15}.$$

iii. If the channel is in constant use, about how long do you expect must pass between undetected incorrectly transmitted words? Express your answer as a number of days.
Solution: If the channel is in constant use, then approximatly

$$\frac{(60^2 \cdot 24 \cdot 10^7)}{12} \times 6.59 \times 10^{-15} \approx 0.00047448.$$

words per day are undetected incorrectly transmitted words.

2. Establish the following three properties of the Hamming distance (for binary codes C):

   (a) $d(u, w) = 0$ if and only if $u = w$.
   Solution: Suppose $d(u, w) = 0$. Then, by definition $u = w$ since $d(u, w)$ is defined to be the number of positions in which u and v differ.
   Similarly, suppose $u = w$. Then, by definition, $d(u, w) = 0$ since $d(u, w)$ is defined to be them number of positions in which u and v differ, and since they are equal they differ in 0 positions.

   (b) $d(v, w) = d(w, v)$.
   Solution: $d(v, w) = wt(v + w) = wt(w + v) = d(w, v)$. The second equality holds since component-wise addition for binary words is commutative. Note: $d(v, w) = d(w, v)$ can be seen directly form the definition as well since the number of positions in which v differs from w is the same as the number of positions in which w differs from v.

   (c) $d(v, w) \leq d(v, u) + d(u, w), \forall u \in C$.
   Solution: Let $v = v_1 v_2 \dots v_n$, $w = w_1 w_2 \dots w_n$, and $u = u_1 u_2 \dots u_n$. By definition, $d(v, w) = |\{k \ : \ v_k \neq w_k\}| = |\{k \ : \ v_k \neq u_k\} \oplus \{k \ : \ u_k \neq w_k\}|$ since $d(v, w) = wt(v + w) = (v + u + u + w)$. But, $|A \oplus B| \leq |A| + |B|$ for all sets A, B. Thus, $|\{k \ : \ v_k \neq u_k\} \oplus \{k \ : \ u_k \neq w_k\}| \leq |\{k \ : \ v_k \neq u_k\}| + |\{k \ : \ u_k \neq w_k\}| = d(v, u) + d(u, w)$.

3. Prove that the minimum distance of a linear code is the smallest weight of a non-zero codeword.

   Solution: Suppose C has minimum distance d, and let $u \in C$ be a non-zero codeword with minimum weight. First, $\emptyset = 00 \dots 0 \in C$ since C is a linear code. Thus, $d \leq d(\emptyset, u) = wt(\emptyset + u) = wt(u)$. Now, suppose $wt(u) > d$. Then, there are codewords $v \neq w$ such that $d = d(v, w) = wt(v + w) < wt(u)$. But, C is a linear code. So, $v + w = x \in C$. Hence, there is a non-zero codeword $x \in C$ such that $wt(x) < wt(u)$, which is a contradiction since u is a codeword of minimum weight. Thus, $wt(u) \leq d$. Therefore, $d = wt(u)$.

4. Prove that a code can *simultaneously* correct all error patterns of weight at most t, and unambiguously detect all non-zero error patterns of weight $t + 1$ to d (where $t \leq d$) if and only if it has minimum distance at least $t + d + 1$.

   For example, consider $C = \{000, 111\}$. This single error correcting code detects all non-zero error patterns of weight at most 2. But, if 000 is sent and 110 is received, then only one error is detected and the received word is incorrectly decoded as 111. The ambiguity here is that it is not clear whether the error pattern is 110 or 001.

Solution: Let C be a code.

Suppose C corrects all error patterns of weight less than or equal to t and unambiguously detects all non-zero error patterns of weight $t+1$ to d. Let $D = d(u, v)$ for some $u, v \in C$. Suppose $0 < D < t + d + 1$ and let w be any error pattern formed from $u + v$ by changing d 1s to 0s. Then $wt(w) = d + 1$. Now, $d(u, u + w) = d + 1$ and $d(v, u + w) = d$. Thus, $u + w$ is closer to v than to u, which is a contradiction. Hence, $D \geq t + d + 1$.

Suppose C has minimum distance $d + t + 1$. Since $t \leq d$, $2t + 1 \leq d + t + 1$. Thus, by the theorem from class C corrects all error patterns of weight $\leq t$. Furthermore, if w is sent and an error pattern u with $\leq d$. Then, $d(w, w + u) = t + 1$. So, C does not correct u, but it does detect u, since $(d+t+1)-d = t+1$. Therefore, C unambiguously detects all nonzero error patterns of weight $t + 1$ to d.

5. Let H be a parity matrix for a linear code C. Prove that C has minimum distance d if and only if any set of $d - 1$ rows of H are linearly independent and some set of d rows of H are linearly dependent.

   Solution: Suppose C has minimum distance d. Then, by question 3 $d = wt(u)$ where $u = u_1 u_2 \dots u_n$ is a nonzero codeword of minimum weight. Consider uH. Since $u \in C$ and H is a parity check matrix for C, $uH = 0$. Thus, some $d = wt(u)$ rows of H sum to zero. In other words, there is a set of d linearly dependent rows! Now, suppose there is a set of $d - 1$ linearly dependent rows. Then, there is some word $x \in 2^n$ with $wt(x) = d - 1$ such that $xH = 0$. But, this means $x \in C$ since H is a parity check matrix for C. But, this is a contradiction since $wt(x) = d - 1 < d = wt(u)$ and u was a nonzero codeword of minimum degree.

   Suppose any set of $d - 1$ rows of H are linearly independent and some set of d rows of H are linearly dependednt. Since there is some set of d rows of H that are linearly dependent, there exists a $u \in C$ with $wt(u) = d$ such that $uH = 0$. Furthermore, since every set of $d - 1$ rows of H are linearly independent, there is no $x \in 2^n$ with $wt(x) \leq d - 1$ such that $xH = 0$. Thus, there is no $x \in C$ with $wt(x) \leq d - 1$ since H is a parity check matrix for C. Hence, $u \in C$ is a codeword of minimum weight (wt(u)=d). Therefore, by question 3, C has minimum distance d.

6. Let $S = \{11000, 01111, 11110, 01010\}$, and $C =< S >$.

   (a) Find a generator matrix, and parity check matrix for C.

   Solution:

$$\begin{bmatrix} 11000 \\ 01111 \\ 11110 \\ 01010 \end{bmatrix} \rightarrow \begin{bmatrix} 11000 \\ 01111 \\ 00110 \\ 01010 \end{bmatrix} \rightarrow \begin{bmatrix} 10001 \\ 01001 \\ 00101 \\ 00011 \end{bmatrix}$$

   So, the elements of S are linearly independent. Thus, $G = \begin{bmatrix} 10001 \\ 01001 \\ 00101 \\ 00011 \end{bmatrix}$ is a generator

matrix for C. Furthermore, $G = \begin{bmatrix} I_4 \mid X \end{bmatrix}$ where $X = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$. Thus, $H == \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ is a parity check matrix for C.

(b) What is the dimension of C, and of $C^{\perp}$?

Solution: The dimension of C is the number of codewords in a basis for C, which is the same as the number of linearly independent rows of G. Thus, $dim(C) = 4$, and $dim(C^{\perp}) = 1$.

(c) An (n, k, d)-code is a linear code of length n, dimension k, and minimum distance d. Find the parameters (n, k, d) for C and $C^{\perp}$. (You may want to use the result of the previous question.)

Solution: C is a $(5, 4, 3)$-code by question 5 since H has at most 1 linearly independent row (i.e., they're all 1). $C^{\perp}$ is a $(5, 1, 5)$-code by question 5 since $G^T = H_{C^{\perp}} = \begin{bmatrix} 1000 \\ 0100 \\ 0010 \\ 0001 \\ 1111 \end{bmatrix}$, which has 4 linearly independent rows but not 5.

7. Let C be the linear code with generator matrix

$$G = \begin{bmatrix} 1000111 \\ 0100110 \\ 0010101 \\ 0001011 \end{bmatrix}.$$

Assign data to the words in K^4 by letting the letters $A, B, \ldots, P$ correspond to $0000, 0001, \ldots, 1111$, respectively.

(a) Encode the message $CALL\ HOME$ (ignore the space).

Solution: From G and the correspondence we get:

$$CG = 0010G = 0010101$$

$$AG = 0000G = 0000000$$

$$LG = 1011G = 1011001$$

$$HG = 0111G = 0111000$$

$$OG = 1110G = 1110100$$

$$MG = 1100G = 1100001$$

$$EG = 0100G = 0100110.$$

So, $CALL\ HOME$ is encoded as

$$0010101\ 0000000\ 1011001\ 1011001\ 0111000\ 1110100\ 1100001\ 0100110.$$

(b) Suppose the message $0111000, 0110110, 1011001, 1011111, 1100101, 0100110$ is received. decode this message using $CMLD$ procedure involving syndromes and cosets (syndrome decoding). How many errors occurred in transmission?

Solution:

$$H = \begin{bmatrix} 111 \\ 110 \\ 101 \\ 011 \\ 100 \\ 010 \\ 001 \end{bmatrix}.$$

So,

$$0111000H = 000 \quad 0110110H = 101 \quad 1011001H = 000$$

$$1011111H = 110 \quad 1100101H = 100 \quad 0100110H = 000.$$

By question 5, $d = 3$. So, C is 1-error correcting. Thus, coset representatives will have weight 1. So, we can find a partial SDR by looking at the 7 weight 1 words for their syndromes to get

Syndrome	Coset Leader
000	0000000
100	0000100
101	0010000
110	0100000

So, $0111000, 0110110, 1011001, 1011111, 1100101, 0100110$ decodes to $HELPME$, where single errors occurred in words 2, 4, and 5. Thus, the message is $HELP\ ME$ and 3 errors occurred in transmission.

2.11 Assignment 2

1. (a) Prove the *Singleton bound*: For an (n, k, d)-code, $d - 1 \leq n - k$. (Hint: consider a parity check matrix.)

   Solution: Let C be a (n, k, d)-code. By definition C is a linear code. Now, suppose $d - 1 > n - k$, and consider a parity check matrix H for C. Since C has length n and dimension k, H is some permutation of a systematic parity check matrix $H' = \begin{bmatrix} X \\ I_{n-k} \end{bmatrix}$. But, by question 5 on assignment 1, C has minimum distance d if and only if any set of $d - 1$ rows of H are linearly independent and some set of d rows of H are linearly dependent. But, this is a contradiction since the rows of H corresponding to I_{n-k} form a maximally linearly independent set of rows and $d - 1 > n - k$. Therefore, $d - 1 \leq n - k$ for any (n, k, d)-code.

   (b) An (n, k, d)-code is called *maximum distance separable* (MDS) if equality holds in the Singleton bound, that is, if $d = n - k + 1$. Prove that the following statements are equivalent.

   i. C is MDS.

   ii. Every $n - k$ rows of the generator matrix are linearly independent.

   iii. Every k columns of the generator matrix are linearly independent.

   Solution: Let C be a (n, k, d)-code.

   Suppose C is MDS. Then, $d = n - k + 1$, or equivalently $d - 1 = n - k$. Thus, by question 5 on assignment 1 every $d - 1 = n - k$ rows of the parity check matrix H are linearly independent.

   Suppose every $n - k$ rows of the parity check matrix H are linearly independent. Then, $d - 1 \geq n - k$ by question 5 on assignment 1. Furthermore, $d - 1 \leq n - k$ by part (a). Therefore, $d - 1 = n - k$. Hence, C is MDS

   Still can't figure this out!!!!!!!!!

   (c) Show that the dual of an $(n, k, n - k + 1)$ MDS code is an $(n, n - k, k + 1)$ MDS code.

   Solution: Suppose C is a $(n, k, n - k + 1)$-MDS-code. Then, the dual, $C^{\perp}$, of C is a $(n, n - k, d)$-code. But, by part (b) every k columns of the generator matrix, G, for C are linearly independent. Thus, every k rows of $G^T = H_{C^{\perp}}$, are linearly independent. But, $k = n - (n - k)$. So, by part (b) $C^{\perp}$ is an MDS and $d = n - (n - k) + 1 = k + 1$. Thus, $C^{\perp}$ is an $(n, n - k, k + 1)$ MDS code.

2. Let C be a Hamming code of length 15. Find the number of error patterns the extended code C^* will detect and the number of error patterns that C^* will correct.

   Solution: Let C be a Hamming code of length 15. Then, C is a $(15, 11, 3)$-code. So, C^* is a $(16, 11, 4)$-code.

   Suppose $u \in K^{16}$ is an error pattern. Now, u is detected if and only if $u + v \notin C^*$ for all $v \in C^*$. Hence, u is detected if and only if $u \notin C^*$ since $w + v \in C^*$ for all $w, v \in C^*$ because C^* is linear. Therefore, C^* detects $|K^{16}| - |C^*| = 63488$ error patterns.

   From class we know that an extended code C^* does not correct anymore errors that the base code C. So, since C is a Hamming code and Hamming codes are perfect, C^*

only corrects error patterns of distance $t = 1$ from codewords. Therefore, C^* corrects $\binom{16}{1}$ error patterns.

3. Count the number of codewords of weight 7 in the Golay code C_{23}. (Hint: Start by proving that every word of weight 4 in K^{23} is distance 3 from exactly one codeword.)

   Solution: Let C be the Golay code C_{23}. Then, from class we know C is perfect and is 3-error correcting (i.e., the minimum distance is 7). Therefore, K^{23} is partitioned by balls of radius 3 around codewords of C. Hence, each word in K^{23} is in a ball of radius radius 3 around exactly one codeword. Thus, every word of weight 4 in K_{23} is in a ball of radius 3 around exactly on codeword. But, this means if u is a word in K^{23} of weight 4 then $d(u, v) \leq 3$ for exactly one codeword $v \in C$. Furthermore, for any nonzero codeword v of C, $wt(v) \geq 7$, since C has minimum distance 7. But, this implies that $d(u, v) \geq 3$ for all $u \in K^{23}$ of weight 4 and $v \in C$ since $d(u, v) = wt(u + v) \geq 7 - 4 = 3$ (i.e., u can share at most 4 1s with v). Therefore, every word u of weight 4 in K^{23} is distance 3 from exactly one codeword v. Moreover, the codeword v is of weight 7 and u must share all its 1s with v.

   Now, there are $\binom{23}{4}$ words of weight 4 in K^{23} and each word of weight 7 has exactly $\binom{7}{4}$ words of weight 4 around it. Therefore, there are $\binom{23}{4}/\binom{7}{4} = 253$ codewords of weight 7.

4. Let $G(1, 3)$ be the generator matrix for $RM(1, 3)$. Decode the following received words:

   $$01011110\ 01100111\ 00010100\ 11001110.$$

   Solution:

   $$G(1,3) = \begin{bmatrix} 1111 & 1111 \\ 0101 & 0101 \\ 0011 & 0011 \\ 0000 & 1111 \end{bmatrix}.$$

   $d(0101|1110, 0101|1010) = 1$, thus we decode to 0101
   $d(0110|0111, 0110|0110) = 1$, thus we decode to 0110
   $d(1100|1110, 1100|1100) = 1$, thus we decode to 1010

   00010100 cannot be decoded since there is no codeword distance 1 away (i.e., 2 or more errors have occurred). Thus, request retransmission.

5. If possible, devise a 6-ary single error correcting code (i.e., one that uses 0,1,2,3,4,5) with length 6, and 4 information digits. Describe the code, an encoding procedure, and decoding procedure. Prove that your code corrects all single errors. What is its information rate? If not possible, say why not.

   Solution: Let C be the set of 6-ary strings of length 6 such that: $x_1x_2x_3x_4$ are information digits and x_5x_6 are parity check digits chosen so that:

   $$S_1 = 1x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 \equiv 0 \pmod 7$$

   and

   $$S_1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \equiv 0 \pmod 7$$

are satisfied. (Note: x_5 or x_6 could be 6, but ingnore these cases.) So, there are 6^4 possible codewords.

To encode, if $1x_1+2x_2+3x_3+4x_4 \equiv -a \pmod 7$ and $x_1+x_2+x_3+x_4 \equiv -b \pmod 7$. Then, $x_5 \equiv 6(a+b) \pmod 7$ and $x_6 \equiv 2b+a \pmod 7$.

To decode a sent word $r = x_1x_2x_3x_4x_5x_6$, compute $r \cdot [S_1, S_2] = r \cdot \begin{bmatrix} 11 \\ 21 \\ 31 \\ 41 \\ 51 \\ 61 \end{bmatrix}$.

If $S_1 \equiv S_2 \equiv 0$ then r is a codeword (i.e., decode to $x_1x_2x_3x_4$).
If $S_1 \not\equiv 0$ and $S_2 \not\equiv 0$, then assume an error e of magnitude $e \equiv S_2 \pmod 7$ occurred in position $i \equiv S_1S_2^{-1} \pmod 7$ and decode r as $x_1 \dots (x_i - e) \dots x_6$. If $S_1 \equiv 0$ or $S_2 \equiv 0$ but not both then atleast 2 errors have occurred so repeat transmission.

Suppose some codeword $x_1x_2x_3x_4x_5x_6$ is sent and a error $e \not\equiv 0 \pmod 7$ occurs in the ith digit. Then, the received word is $x_1 \dots (x_i + e) \dots x_6$. So,

$$S_1 = 1x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 + i \cdot e \equiv i \cdot e \pmod 7$$

and

$$S_1 = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + e \equiv e \pmod 7$$

since $x_1x_2x_3x_4x_5x_6$ is a codeword. Therefore, S_2 tells us the magnitude of e and $S_1S_2^{-1} \equiv i \pmod 7$ tells us the position it occurs in. Therefore, it corrects all single errors using the decoding procedure above.

Finally, our code has $6^4 = 1296$ codewords and is of length 6. Therefore, the information rate is $\frac{1}{6}\log_6(6^4) = \frac{4}{6} = \frac{2}{3}$.

6. Codewords $x_1x_2x_3x_4x_5$ with decimal digits are defined by $x_1x_2x_3x_4 \equiv x_5 \pmod 9$, where x_5 is a check digit.

   (a) Show that the parity check equation is equivalent to $x_1+x_2+x_3+x_4 \equiv x_5 \pmod 9$.
   Solution: Since $x_1 + x_2 + x_3 + x_4$ is a decimal number, $x_1 + x_2 + x_3 + x_4 = x_1(10)^3 + x_2(10)^2 + x_3(10) + x_4$. So,

   $$x_1 + x_2 + x_3 + x_4 \equiv x_5 \pmod 9$$

   if and only if

   $$x_1(10)^3 + x_2(10)^2 + x_3(10) + x_4 \equiv x_5 \pmod 9$$

   if and only if

   $$x_1 + x_2 + x_3 + x_4 \equiv x_5 \pmod 9$$

   since $10^x \equiv 1 \pmod 9$.

   (b) Assuming that each single error is equally likely, what percentage of single errors are undetected?
   Solution: Suppose each single error is equally likely. Now, $x_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. But, $0 \equiv 9 \pmod 9$. So, for $1 \leq i \leq 4$ if x_i is 0 or 9 and e occurs such that x_i changes from a 0 to a 9, or visa versa, then e will not be detected. This does not hold for x_5 since x_5 can't be 9. Thus, if w is sent and an error e occurs then there is a $\frac{2}{10} \cdot \frac{1}{9} \cdot \frac{4}{5} = \frac{4}{225}$ chance that e is not detected.

(c) Which errors involving transposition of digits are detected?

Solution: By part (a) any transposition of the digits $x_1x_2x_3x_4$ will not be detected since $x_1 + x_2 + x_3 + x_4 \equiv x_5 \pmod 9$ no matter what order $x_1x_2x_3x_4$ are in. So, the only transpositions that can be detected are transpositions involving x_5 and x_i, $1 \leq i \leq 4$, such that $x_5 \not\equiv x_i \pmod 9$.

7. Consider the code with 10 decimal digits in which the check digit x_{10} is the least residue of $x_1x_2 \cdots x_9 \pmod 7$ (that is, $0 \leq x_10 \leq 6$). (As of my last information, this code is used by UPS and Federal Express.)

(a) Under what conditions are single errors undetected?

Solution: First of all, in codewords $0 \leq x_{10} \leq 6$, so any nonzero error in x_{10} is always detected. Now, since $10 \equiv 3 \pmod 7$, by similar reasoning to question 6, our check equation is equivalent to

$$2x_1 + 3x_2 + x_3 + 5x_4 + 4x_5 + 6x_6 + 2x_7 + 3x_8 + x_9 \equiv x_{10} \pmod 7.$$

Thus, if an error e occurs in x_i where $1 \leq i \leq 9$ and goes undetected, then

$$c_i(x_i + e) \equiv c_ix_i \pmod 7.$$

This is equivalent to $(x_i + e) \equiv x_i \pmod 7$ since $gcd(c_i, 7) = 1$. Therefore, e goes undetected if and only if e change a 0 to a 7, 1 to an 8, or an 2 to an 9, and vice versa.

(b) Assuming that each single error is equally likely, what percentage of single errors are undetected?

Solution: Suppose each single error is equally likely. Then, the total number of errors is $(9 \times 9 \times 10) + (7 \times 9)$. So, $\frac{54}{873} \times 100 = 6.19$ percent.

(c) Repeat (b) for errors involving transposition of digits.

Solution: From the check equation in part (a) we know that any transposition between the digits (x_1, x_7), (x_2, x_8), and (x_3, x_9) will go undetected. Furthermore, transposition between digits that are congruent

8. Is it true that all words in a self-dual code have even weight?

Solution: Yes, all words in a self-dual binary code C have even weight. By definition every word in a self dual code is orthogonal to itself. Thus, if $v = v_1v_2 \ldots v_n \in C$ then

$$v \cdot v = v_1v_1 + v_2v_2 + \ldots + v_nv_n \equiv 0 \pmod 2.$$

But, if $v_i = 1$ then $v_i \cdot v_i = 1$. Therefore, there are an even number of terms in the sum $v \cdot v$ that equal 1, which implies there are an even number of 1s in v. Therefore, $wt(v)$ is even.

Chapter 3

Cryptography

3.1 Intoduction

Definition 3.1.1. Cryptography is the study of communication in the presence of adversaries.

Definition 3.1.2. Cryptanalysis is the study of methods for defeating information security objectives.

Definition 3.1.3. Cryptology is both the study of communication in the presence of adversaries and the study of methods for defeating information security objectives.

Definition 3.1.4. An **Alphabet** is a finite set $\mathbb{A}$. The set $\mathbb{A}^n$ is the set of all strings of length n over $\mathbb{A}$. The set $\mathbb{A}^*$ is the set of all finite strings over $\mathbb{A}$.

Definition 3.1.5. An **Encryption scheme** is a tuple (M, C, K, E_k, D_k), where M is the plaintext message space, C is the ciphertext space, K is the key space (often a key pair (e,d)), $E_k : M \to e$ is an encryption function, and $D_k : e \to M$ is a decryption function $D_k : e \to M$ such that $D_k(E_k(m)) = m, \forall m \in M$.

There are two main types of **Adversaries**. The first is passive, they eavesdropping only. The second is active, they may insert messages or block transmissions. The goals of encryption are to provide secrecy and detect altered or forged messages. By "breaking" a system we normally mean discovering or figuring out the plaintext from the ciphertext.

Definition 3.1.6. Types of Security:

1. **unconditionally security**: the adversary can gain no knowledge about the plaintext no matter how much ciphertext or whatever computing power is available.
2. **Computationally secure**: it is not feasible to discover the plaintext from the ciphertext using known methods and a given amount of computing power.

3. **provably secure**: breaking the system is at least as hard as solving some "hard" (i.e. np-complete) mathematical problem.

Kerckhoff's Principle (1883): The security of the cipher should rest with the key alone (i.e., security is maintained even if the adversary knows the encryption scheme).

Definition 3.1.7. Types of Attacks on a encryption scheme:

1. **ciphertext only attack**: the adversary tries to recover the plaintext or even better the key by observing some amount of ciphertext.
2. **known plaintext attack**: the adversary has some amount of plaintext and the corresponding ciphertext.
3. **chosen plaintext attack**: same as known plaintext except the adversary chooses the plaintext.
4. **chosen ciphertext** pick ciphertext, and get to know the plaintext. the plaintext corresponding to ciphertext chosen by the adversary is known.

Definition 3.1.8. A **Symmetric key encryption scheme** is an encryption scheme where the same amount of computational resources are needed to find the decryption function given the encryption key as to find the encryption function given the decryption key.

Definition 3.1.9. A **Block cipher** is a cipher that encrypts blocks of a fixed number of characters.

Definition 3.1.10. A **Stream** cipher or (**State** cipher) is a cipher where the mapping of a block may depend on its location in the message.

3.2 Non-secure encryption schemes

3.2.1 simple substitution ciphers

Definition 3.2.1. A simple substitution cipher is a cipher where the keys are permutations of $\mathbb{A}$ and the cipher text is produced by substitution of letters.

For example, a shift cipher on the numbers corresponding to the capital letters (i.e., $A, B, \ldots, Z \to 1, 2, \ldots, 26$) uses an encryption key e which is a fixed shift (mod 26), $\alpha \to \alpha + e$ (mod 26). The Decryption key is $d = 26 - e$.

Any substitution cipher is completely insecure against chosen plaintext attacks (i.e., just use plaintext AB...Z) Also there are only 26 keys in the shift cipher so a ciphertext only attacks is not hard.

Any simple substitution cipher is completely insecure against a ciphertext only attack by frequency analysis.

Definition 3.2.2. A **Affine** cipher with the correspondence $A, B, \ldots, Z \to 1, 2, \ldots, 26$ uses the encryption function $E(x) = ax + b$ (mod26), where the $\gcd(a, 26) = 1$ (i.e., needs to be one-to-one). The decryption function $D(x) == a^{-1}(y - b)$ (mod 26). The keys is (a^{-1}, b)

Note: if $a = 1$ then it is a shift cipher. For decryption, to find a^{-1} use the euclidean algorithm to find r and s such that $ar + ms = 1$, where $m \in \mathbb{Z}_m$. So, $ar = 1 + m(-s)$ so $ar \equiv 1$. Therefore, $a^{-1} \equiv r \pmod{m}$. Hence, the key is (a^{-1}, b).

Definition 3.2.3. A **poly-alphabetic** cipher is a cipher where each source symbol can map to one of several possible ciphertext symbols.

For example, the Vigenere cipher. The symbols in $\mathbb{A}$ we correspond to $0, 1, \ldots, |\mathbb{A}| - 1$ and create a keyword $k = k_1 k_2 \ldots k_n \in A^n$. The encryption is performed on n characters blocks and so is decryption. Suppose $\mathbb{A} = \{A, B, \ldots Z\}$, $k = golf$, and we want to send the plaintext: "follow through". To encrypt add character wise in blocks of 4. More specifically, line "follow through" up with "golfgolfgolfg" and add f with g, o with o, l with l, l with f, etc. To decrypt just subtract "golfgolfgolfg" from the ciphertext.

3.2.2 Breaking some simple substitution ciphers

In the Vigenere cipher the block length is the same as the key length. Therefore, the same mapping is applied to each block. So, we can apply frequency analysis to try to figure out the block length, then the key. This is similar but a bit more complicated then the following example, where we break the following Affine cipher.

Suppose $A, B, \ldots, Z \to 0, 1, \ldots, 25$ The Affine cipher has the encryption function $x \to ax + b \pmod{26}$, where $a, b \in \mathbb{Z}_{26}$ and $gcd(a, 26) = 1$. Suppose the following is observed

from an affine cipher:

$$FMXVEDKAPHFENDRB$$

$$NDKRXRSREFMORUDS$$

$$DKDVSHVUFEDKAPRK$$

$$DLYEVLRHHRH$$

The frequencies of the letters are:

$A:2$	$B:1$	$C:0$	$D:6$	$E:5$	$F:4$
$G:0$	$H:5$	$I:0$	$J:0$	$K:5$	$L:2$
$M:2$	$N:1$	$O:1$	$P:3$	$Q:0$	$R:8.$
$S:3$	$T:0$	$U:2$	$V:4$	$W:0$	$X:2$
$Y:1$	$Z:0$				

Now, we figure out the relative frequencies. The most frequent character are: $R(8)$, $D(6)$, E, H, K (5 each), and F, S, V (4 each). So, a good guess would be that $E \to R$ $T \to D$. Therefore, $E_k(4) = 17$ and $E_k(19) = 3$. Since $E_k(x) = ax + b$, this leads to $4a + b \equiv 17 \pmod{26}$ and $19a + b \equiv 3 \pmod{26}$. Unfortunately, the gcd is not one than so not it. Guess $E \to R$ $T \to E$ also fails. Guess $E \to R$ $T \to H$ also fails.

Try $E \to R$ $T \to K$. This leads to $4a + b \equiv 17 \pmod{26}$ and $19a + b \equiv 10 \pmod{26}$. Therefore, $a = 3$ and $b = 5$, which is a legal key. Now see what happens. The decryption key is $(3^{-1}, 5) = (9, 5)$. Therefore the decryption function $D_k(y) = 9(y - 5) = 9y - 19 \pmod{26}$ If this is applied to the cypher text, we get:

$$ALGORITHMSAREQUITEGENERAL$$

$$DEFINITIONSOFARITHMETICPROCESSES$$

Therefore, we probably have the right key (potentially not if the text was shorter since you could still get something that made sense but was not correct).

3.2.3 Affine decryption Java Code

```
public static void DecodeGivenAInverseandB (String s1, int a, int b){

    int i, j, k, l;
    char letter, replacementLetter;
    String s2;
    s2="";

    for(l=0; l< s1.length(); l++){

        i=0;

        for(letter ='A'; letter<='Z'; letter++){
            if(s1.charAt(l)==letter){
                break;
            }
            i++;
        }

        replacementLetter = 'A';
        j = (a*(i-b))%26;

        if(j<0){
            j=j+26;
        }

        for(k=0; k<j; k++){
        replacementLetter++;
        }

        s2=s2 + replacementLetter;

    }

    System.out.println(s2);

}
```

3.2.4 Frequency Analysis Java Code

```
public static void FrequencyOfLetters (String s1){

    int i, numberOfOccurrencesOfLetter;
    char letter;

    for(letter ='A'; letter<='Z'; letter++){

        numberOfOccurrencesOfLetter = 0;

        for(i=0; i< s1.length(); i++){

            if(letter == s1.charAt(i)){
                numberOfOccurrencesOfLetter++;
            }
        }

    System.out.println(letter + ":" +   numberOfOccurrencesOfLetter);

    }

}
```

3.3 Public Key Systems

In a private key system the decryption key D_k is either the same as the encryption key E_k or can be easily obtained from it. Exposure of the encryption key E_k renders the system insecure. Due to this, private key systems require prior communication of the keys somehow.

In a public key system, even if E_k is known, then it is (believed to be) computationally infeasible to determine D_k. So, only someone who has D_k can decode it. Therefore, E_k can

be made public. This makes public key systems very useful. To send a message using a public key system you first look up the public key of you intended recipient, then encrypt using their key. There is no need for a key exchange. Generally the idea of public key systems is attributed to Diffie and Hellman, 1976. The first public key system: RSA (Rivest, Shamir, Adleman, 1977) A public key system cannot be unconditionally secure since if the adversary has the ciphertext, then they can use the public key to encrypt every possible plaintext until a match is found.

Definition 3.3.1. A function $f : M \to C$ is called **one-way** if $f(m)$ is "easy" to compute $\forall m \in M$ and for most $c \in C$ it is "hard" to find m such that $f(m) = c$.

For example, the passwords in the unix system, any user/attacker can see the password file, but it is believed to be encrypted with a 1-way function. A 1-way function f is a "trapdoor" if it has the property that with some extra information it becomes possible to find m such that $f(m) = c$. For public key systems, the extra information is the decryption key.

3.3.1 RSA cryptosystem

RSA is a public key system. The message space and the cipher space are both $\mathbb{Z}_n^*$ (i.e., $M = C = \mathbb{Z}_n^*$) such that $n = pq$ where p and q are "large" primes. The key pair is $k = (a, b)$, where $ab \equiv 1 \pmod{\phi(n)}$, where $\phi(n) = (p-1)(q-1)$. To encrypt a message x you compute $E_k(x) = x^b \pmod{\text{n}}$. To decrypt a ciphertext y you compute $D_k(y) = y^a \pmod{\text{n}}$. n and b are public, while p, q, and a are all secret.

Note the following facts:

1. $ab \equiv 1 \pmod{\phi(n)}$ if and only if $ab = 1 + t \cdot \phi(n)$, $t \in \mathbb{Z}^+$.
2. Euler's theorem: If $\gcd(x, n) = 1$, then $x^{\phi(n)} \equiv 1 \pmod{n}$.
3. If $n = pq$, where p, q are prime, then $x_1 \equiv x_2 \pmod{pq}$ if and only if $x_1 \equiv x_2 \pmod{p}$ and $x_1 \equiv x_2 \pmod{q}$.

Theorem 3.3.2. $D_k(E_k(x)) = x$.

Proof. Suppose $x \in Z_n$ and $\gcd(x, n) = 1$. Then,

$$D_k(E_k(x)) = (x^b)^a = x^{ab} = x^{1+t\phi(n)} \pmod{n}.$$

Therefore, by Euler's theorem

$$D_k(E_k(x)) = x^{1+t\phi(n)} = x(x^{\phi(n)})^t = x \pmod{n}.$$

Now, suppose $\gcd(x, n) = d > 1$. Then, $D_k(E_k(x)) = x^{1+t\phi(n)}$. Since $n = pq$, where p, q are prime, we have $x_1 \equiv x_2 \pmod{pq}$ if and only if $x_1 \equiv x_2 \pmod{p}$ and $x_1 \equiv x_2 \pmod{q}$.

Therefore,

$$\begin{aligned} D_k(E_k(x)) &= x^{1+t\phi(n)} \\ &= x(x^{t(p-1)(q-1)}) \\ &\equiv xx^{t\phi(p)(q-1)} \equiv x \pmod{p} \\ &\equiv xx^{t\phi(q)(p-1)} \equiv x \pmod{q} \end{aligned}$$

□

Example of Using RSA

Suppose Bob chooses two primes $p = 101$ and $q = 113$. Then, $n = pq = 11413$. So, $\phi(n) = (p-1)(q-1) = 11200 = 2^6 5^2 7$. Now, b must be relatively prime to $\phi(n)$ in order to have $ab \equiv 1 \pmod{\phi(n)}$. Therefore, b not divisible by 2,5, or 7. Suppose, Bob chooses $b = 3533$. Then, Bob needs a such that $ab \equiv 1 \pmod{\phi(n)}$. (i.e., use the Euclidean Algorithm to find a and y such that $ba + \phi(n)y = 1$). Now, $a \equiv -4603 \equiv 6597 \pmod{11200}$. So, Bob publicizes b and n, but keeps p, q, and a secret.

To send the message 9276 to Bob, compute $9276^{3533} \equiv 5761 \pmod{11413}$ and send it over the channel. Bob will receives 5761and compute $5761^{6597} \equiv 9276 \pmod{11413}$.

How to Exponentiation Quickly

"square and multiply"

$$b = (b_l b_{l-1} \dots b_0)_2 = 2^l b_l + 2^{l-1} b_{l-1} \dots + b_0.$$

So,

$$x^b = x^{2^l b_l + 2^{l-1} b_{l-1} \dots + b_0} = x^{b_0} (x^2)^{b_1} \dots (x^{2^l})^{b_l}.$$

Thus, we can compute x^1, x^2, ...x^{2^l}, where $l = 1 + \lfloor (\log_2(x)) \rfloor$. Then, multiply together the ones corresponding to the binary digits that equal b.

For example, suppose you want to compute $57^{26} \pmod{91}$.

$$26 = (11010)_2 \text{ and } 1 + \lfloor (\log_2(26)) \rfloor = 5.$$

So, we have

$$57, \quad 57^2 \equiv 3249 \equiv 64, \quad 57^4 \equiv 64^2 \equiv 4096 \equiv 1, \quad 57^8 \equiv 1^2, \quad 57^{16} \equiv 1^2 \pmod{91}.$$

Therefore, $57^{26} \equiv 57^{16} \times 57^8 \times 57^2 \equiv 1 \times 1 \times 64 \equiv 64 \pmod{91}$

Setting-up RSA

1. Generate 2 large primes p and q.

2. Compute $n = pq$, and $\phi(n) = (p-1)(q-1)$.

3. Choose a random number b, where $0 < b < \phi(n)$ such that $\gcd(b, \phi(n)) = 1$.

4. Find $a = b^1 \pmod{\phi(n)}$.

5. Publish b and n in the public directory.

Note: The probability that a randomly chosen b such that $0 < b < \phi(n)$ is relatively prime to $\phi(n)$ is

$$\frac{\phi(\phi(n))}{\phi(n) - 1}.$$

So for 3, just repeatedly choose b and use the Euclidean Algorithm to test whether $\gcd(b, \phi(n)) = 1$. Then, 4 is a by product of the work in for 3.

3.3.2 Probabilistic Primality Testing

Definition 3.3.3. A **decision problem** is a problem with a yes or no answer.

For example, the question "is n compusite?" is a decision problem. Note: there exists a short proof of n being prime if n is prime.

Definition 3.3.4. A **probabilistic algorithm** is an algorithm that uses random numbers.

For example, the Monte-Carlo algorithm is a yes-based probabilistic algorithm. Namely, the Monte-Carlo algorithm is a probabilistic algorithm for a decision problem. It always answers "yes" correctly, and answers "no" incorrectly with some probability. The error probability of a yes-based Monte-Carlo algorithm is ϵ if, when the answer is "yes" the algorithm answers "no" with probability less than ϵ. (i.e., where the probability is computed over all possible random choices made by the algorithm when it is run with a given input).

Main Idea

1. Generate large odd random numbers.

2. Run a yes-based Monte-Carlo algorithm for the decision problem "is n composite?".

3. If the answer is yes, then the number is definitely not prime, try again.

4. If the answer is no, then it is wrong with probability less than ϵ. So re-run the algorithm repeatedly. If it ever answers yes, then start over with a new number. Otherwise, after k runs, n is composite with probability ϵ^k, which can be made as small as desired.

How many runs are expected before a prime is found? By the Prime Number Theorem, PNT (1898): The limit as n goes to infinity of $\frac{\pi(n)}{(n/\ln(n))} = 1$ where $\pi(n)$ is the number of primes less than n. Therefore, the probability that a randomly chosen integer less than k is prime is about $\frac{(k/\ln(k))}{k} = \frac{1}{\ln(k)}$. If k is about 10^{100} (i.e., 100 digit), then the probability is about $\frac{1}{100\ln(10)}$, which is about 1/230. So you'd expect to try about 230 random integers before finding a prime. This is reduced to 115 if we generate only odd numbers.

3.3.3 Miller-Rabin Algorithm

The Miller-Rabin Algorithm is a yes-based Monte-Carlo algorithm for the decision problem "is n composite?" and has error probability less $\frac{1}{4}$.

Basic Idea

The basic idea comes from Fermat's Little Theorem (i.e., If n is prime and the $\gcd(a, n) = 1$, then $a^{n-1} \equiv 1 \pmod{n}$.

Suppose $n - 1 = 2^k m$, where m is odd (i.e., we can write any odd number this way). Then, $a^{2^{k-1}m} \equiv \pm 1 \pmod{n}$. Hence $a^{2^{k-2}m} \equiv \pm 1 \pmod{n}$, so on to $a^m \equiv \pm 1 \pmod{n}$. (i.e., $a^{2^i m} \equiv 1 \pmod{n}$ for all i).

The Algorithm

1. Write $n - 1 = 2^k m$, m odd.
2. Choose a random integer a, $1 < a < n - 1$.
3. Compute $b = a^m \pmod{n}$.
4. If $b \equiv 1 \pmod{n}$, then answer "prime" and "stop".
5. For $i = 0, 1, ..., k-1$. If $b \equiv -1 \pmod{n}$ then answer "prime" and "stop". Else replace b by b^2 and carry on with step 5.
6. Answer "composite".

Theorem 3.3.5. The Miller-Rabin algorithm is a yes-based Monte-Carlo algorithm for deciding if n is composite.

Proof. suppose the algorithm answers composite for some prime n. since the answer is composite we know $a^m \not\equiv 1 \pmod{n}$. In step 5, the sequence of values tested is $b = a^m$, $b = (a^n)^2 = a^{2m}$, $b = (a^{2m})^2 = a^{4m}$, ... $b = a^{2^{k-1}m}$ since the algorithm answers composite, we know $a^{2^i m} \not\equiv -1 \pmod{n}$, $0 \leq i \leq k - 1$ starsince n is prime, we know by fermat's theorem that $a^{2^i m} \equiv 1 \pmod{n}$ $a^{2^i m} \equiv (a^{2^{k-1}m})^2$ therefore $a^{2^{k-1}m} \equiv \pm 1 \pmod{n}$ by star $a^{2^{k-1}m} \not\equiv -1 \pmod{n}$ therefore, $a^{2^{k-1}m} \not\equiv 1 \pmod{n}$ repeating the same argument, we eventually obtain $a^m \equiv 1 \pmod{n}$ a contradiction. □

Note: the error probability in the Miller-Rabin algorithm is at most 0.25.

3.3.4 Attacking RSA

If you can factor $n = pq$ and determine p and q, this gives you $\phi(n) = (p-1)(q-1)$, from which a can be determined as $ab \equiv 1 \pmod{\phi(n)}$. In particular, it is enough to determine $\phi(n)$. If $\phi(n)$ is known, then we have $n = pq$ and $\phi(n) = (p-1)(q-1)$. Two equation in 2 unknowns (p and $4q$). Plug $q = n/p$ into the 2nd equation: $\phi(n) = (p-1)(n/p-1) = n - p - n/p + 1$ or $p^2 - (n - \phi(n) + 1)p + n = 0$. The root are p and q Since they are factors of constant terms.

For example, suppose you have learned $n = 84773093$ and $\phi(n) = 84755668$, then the quadratic is $x^2 - 18426x + 84773093 = 0$. So, the quadratic formula gives the roots 9539 and 8887 and these are the factors of n.

3.3.5 Pollard's $p-1$ Factoring Algorithm

We want to factor n (i.e., discover a divisor).

Theorem 3.3.6. If for every prime power q such that $q|p-1$, where $p|n$, we have $q \leq B$, then $(p-1)|B!$.

Let $a = 2^{B!} \pmod{n}$. Since $p|n$, we have $a \equiv 2^{B!} \pmod{p}$. By Fermat's theorem $2^{p-1} \equiv 1 \pmod{p}$. Since $p-1|B!$ we know $2^{B!} \equiv (2^{p-1})^t \equiv 1^t \pmod{p}$. Therefore, $a \equiv 1 \pmod{p}$ or $p|a-1$. Furthermore, $p|n$ so we have $p|\ \gcd(a-1, n)$. Therefore, we can compute $a-1$ and get a factor of n using the euclidean algorithm.

Suppose p-1 has the prime factorization $p - 1 = p_1^{\alpha_1} \dots p_k^{\alpha_k} = q_1 \dots q_k (q_i = p_i^{\alpha_i})$ note that $\gcd(q_1, q_j) = 1$ if $i \neq j$ by hypothesis, each $q_i \leq B$ therefore, $q_1, \dots, q_k$ all are different terms in 1.2...B=B! therefore, each $q_i|B!$ and since $\gcd(q_1, q_j) = 1$ if $i \neq j$, $p - 1 = q_1 \dots q_k|B!$

Algorithm

Input: two integers n, B.
step 1: compute $a \equiv 2^{B!} \pmod{n}$
step 2: set $d = \gcd(a-1, n)$
step 3: If $1 < d < n$ then success; else no factor found.

For example, $n = 36259$.

Try $B = 5$.

$$2^{1} \equiv 2, \quad 2^{2!} \equiv 4, \quad 2^{3!} \equiv 64, \quad 2^{4!} \equiv 25558, \quad 2^{5!} \equiv 22719 \pmod{n}.$$

Therefore, $a = 22719$, and $\gcd(a-1, n) = \gcd(22718, 36259) = 1$ so it fails.
Try $B = 10$. $2^{10!} \equiv 25251 \pmod{n}$. Therefore, $a = 25251$ and $\gcd(a-1, n) = \gcd(25250, 36259) = 101$, so we have success.
In fact, $n = 36259 = 101x359$. Note: $p - 1 = 100 = 2^2 5^2$ but $5^2 \not| 10 = B$.

3.3.6 Pollard's $p-1$ Java Code

```
public class PollardPMinusOne {
    public static void main (String [] args){
    FactoringUsingPMinusOne(262063);
    FactoringUsingPMinusOne(9420457);
    }
    public static void FactoringUsingPMinusOne (int n){
        boolean nNotFactored = true;
        int i=1;
        double j=1;
        while(nNotFactored) {
            j=gcd(TwoToPowerBFactorialMod(i, n)-1, n);
            if(j!=1){
                System.out.println(n + " factors into: " + (int)j + " and " + (int)(n/j) + " where B=" + i + " was used.");
                nNotFactored = false;
            }
            i++;
        }
    }
    public static double TwoToPowerBFactorialMod (int b, int n){
        double TwoToPowBFact = 2;
        for(int i=1;i<=b;i++){
            TwoToPowBFact = bigPowerMod(TwoToPowBFact, i, n)%n;
        }
        return TwoToPowBFact;
    }
    public static double bigPowerMod (double x, int y, int n) {
        double powerOfXToY = x;
        for(int i=1; i<y;i++){
            powerOfXToY = (powerOfXToY*x)%n;
        }
        return powerOfXToY;
    }
    public static double gcd(double n, double a) {
        if (a == 0){
            return n;
        } else{
        return gcd(a, n % a);
        }
    }
```

3.3.7 The Rabin Cryptosystem

The Rabin Cryptosystem is a public key cryptosystem. It is secure against chosen plaintext attacks unless $n = pq$ can be factored, but is insecure agains chosen ciphertext attacks.

The Rabin Cryptosystem requires two distinct primes p and q each congruent to 3 modulo 4. The message space and ciphertext space are $M = C = \mathbb{Z}_n$. The public key is some number

$B \in \mathbb{Z}_n$. To encrypt you use

$$E_k(x) = x(x + B) \pmod{n}$$

and to decrypt you use

$$D_k(y) = \sqrt{4^{-1}B^2 + y} - 2^{-1}B.$$

Thus, n and B are public, while p and q are private.

Note: It turns out encryption is not one-to-one! Therefore, decryption not unambiguous.

Chinese Remainder Theorem

Consider the system of congruences

$$\begin{aligned} x &\equiv a_1 \pmod{n_1} \\ x &\equiv a_2 \pmod{n_2} \\ x &\equiv a_3 \pmod{n_3} \\ x &\equiv a_4 \pmod{n_4} \\ &\cdots \\ x &\equiv a_k \pmod{n_k} \end{aligned}$$

If $\gcd(n_i, n_j) = 1$ when $c \neq j$, then the system has a unique solution modulo $n_1, n_2, \ldots, n_k$. The solution is $x = \sum_{i=1}^{k} a_i M_i y_i \pmod{n_1 n_2 \ldots n_k}$ where $M_i = n_1 n_2 \ldots n_k / n_i$ and $y = M_i^{-1} \pmod{n_i}$.

Theorem 3.3.7. If p and q are different odd primes, then the congruence $x^2 \equiv 1 \pmod{pq}$ has exactly 4 solutions modulo pq.

Proof. $x^2 \equiv 1(modpq)$ if and only if $x^2-1 \equiv (x-1)(x+1)(modpq)$ if and only if $(x-1)(x+1) = kpq$ therefore, $p|(x-1)(x+1)$ and $q|(x-1)(x+1)$ therefore, $x \equiv \pm 1(modp)$ and $x \equiv \pm 1(modq)$ consider the 4 systems
(1) $x \equiv 1(modp)$ and $x \equiv 1(modq)$
(2) $x \equiv 1(modp)$ and $x \equiv -1(modq)$
(3) $x \equiv -1(modp)$ and $x \equiv 1(modq)$
(4) $x \equiv -1(modp)$ and $x \equiv -1(modq)$
by the CRT each system has a unique solution (mod pq). Since p and q are odd, $1 \not\equiv -1(modp)$ and $1 \not\equiv -1(modq)$. Therefore, there are 4 unique solutions. □

Definition 3.3.8. If $t \in \mathbb{Z}$, then a number s is a square root of t if $s^2 \equiv t \pmod{n}$.

Note: not all numbers in $\mathbb{Z}_n$ have square roots (i.e., at most half of them do).

Suppose n is an odd prime then every number has 0 or 2 square roots:

$$s^2 \equiv t \iff (-s)^2 \equiv t \text{ and } s \neq -s.$$

If n is composite then numbers can have more that 2 square roots (i.e. modulo 15, 1 has four square roots).

Let $n = pq$, where p and q are different odd primes. Let ω be one of the four square roots of 1modulo n. Then,

$$\begin{aligned} E_k(\omega(x + 2^{-1}B) - 2^{-1}B &= [\omega(x + 2^{-1}B) - 2^{-1}B][\omega(x + 2^{-1}B) - 2^{-1}B + B] \\ &= \omega^2(x + 2^{-1}B)^2 - 2^{-1}B\omega(x + 2^{-1}B) + B\omega(x + 2^{-1}B) \\ &\quad - 2^{-1}B\omega(x + 2^{-1}B) + 4^{-1}B^2 - 2^{-1}B^2 \\ &= (x + 2^{-1}B)^2 - B\omega(x + 2^{-1}B) + B\omega(x + 2^{-1}B) + 4^{-1}B^2 - 2^{-1}B^2 \end{aligned}$$

Thus, it is independent of ω, so there are four plaintext that give the same ciphertext.

Suppose the ciphertext y is received. How do we find x such that $x(x+B) \equiv y \pmod{n}$? First, make the substitution $x_1 = x + 2^{-1}B (x = x_1 - 2^{-1}B)$. Then the congruence becomes $(x_1 - 2^{-1}B)^2 + B(x_1 - 2^{-1}B) \equiv y \pmod{n}$. This holds, if and only if $x_1^2 - 2x_1 2^{-1}B + (2^{-1}B)^2 + x_1 B - 2^{-1}B^2 \equiv y \pmod{n}$ if and only if $x_1^2 \equiv 2^{-1}B^2 - (2^{-1}B)^2 + y \pmod{n}$. So, let $c = 2^{-1}B^2 - (2^{-1}B)^2 + y$. Then, $x_1^2 \equiv c \pmod{n}$, which is equivalent to $x_1^2 \equiv c \pmod{p}$ and $x_1^2 \equiv c \pmod{q}$. The first congruence has 0 or 2 solutions and same for the second. So, using the CRT as before, we get up to 4 solutions to the original congruence.

Let m be a prime number $a \neq 0 \pmod{m}$ is a quadratic residue modulo m. If $x^2 \equiv a \pmod{M}$ has a solution, then is a quadratic non-residue modulo m; Otherwise there are exactly $m - \frac{1}{2}$ quadratic residues modulo m if m if prime.

Recall: Euler's Criterion. Let m be prime, then a is a quadratic residue modulo m if and only if $a^{m-\frac{1}{2}} \equiv 1 \pmod{m}$.

Now, suppose a is a quadratic residue modulo m. How do we find x such that $x^2 \equiv a \pmod{m}$. If $m \equiv 1 \pmod{4}$ then there is no known efficient deterministic method to find it. But, when $m \equiv 3 \pmod{4}$ then the square roots are easy to find.

Suppose a is a quadratic residue modulo m then

$$(\pm a^{(m+1)/4})^2 \equiv a^{(m+1)/2} \equiv a a^{(m-1)/2} \equiv a \pmod{m}$$

by Euler's criterion. Therefore, the square roots of a are $\pm a^{(m+1)/4}$.

Therefore, if $n = pq$, then $x_1^2 \equiv c \pmod{m}$ if and only if $x_1^2 \equiv c \pmod{p}$ and $x_1^2 \equiv c \pmod{q}$. So, we can use the previous method to get 2 square roots of c modulo p and 2 square roots of c modulo q. Hence, using the CRT as before we get up to four solutions to the original congruence. Once, the four values of x_1 are know the possible values of x can be determined.

Example

Let $n = 7 \times 11$ and $B = 9$. Then, $E_k(x) = x(x + 9) \pmod{77}$ and $D_k(x) = \sqrt{1 + y} - 43 \pmod{77}$. Suppose the ciphertext 22 is received. Then,

$$D_k(22) = \sqrt{(23)} - 43 \pmod{77}.$$

So, we need

$$\sqrt{(23)} \pmod{77}.$$

Since 7 and 11 are both congruent to 3 modulo 4, we have:

$$23^{(7+1)/4} \equiv 2^2 \equiv 4 \pmod{7}$$

$$23^{(11+1)/4} \equiv 1^3 \equiv 1 \pmod{11}.$$

Hence, the square roots of 23 modulo 7 are ± 4 and the square roots of 23 modulo 11 are ± 1. So, we have $4 \equiv \sqrt{23} \pmod{7}$ and $1 \equiv \sqrt{23} \pmod{11}$. So, by the chinese remainder theorem, $x_1 \equiv \sqrt{23} \pmod{77}$, where $x_1 = 4 \cdot 11 \cdot y_1 + 1 \cdot 7 \cdot y_2$ with $y_1 \equiv 11^{-1} \equiv 2 \pmod{7}$ and $y_2 \equiv 7^{-1} \equiv 8 \pmod{11}$. Therefore, $x_1 \equiv 144 \equiv -10 \pmod{77}$.

Similarly the other square roots of 23 are 10, and $\pm 32 \pmod{77}$. The 4 possible plaintexts are $10 - 43 \equiv 44 \pmod{77}$, $67 - 43 \equiv 24 \pmod{77}$, $32 - 43 \equiv 66 \pmod{77}$, and $45 - 43 \equiv 2 \pmod{77}$.

Security

Suppose the attacker has a decryption algorithm A. Then:
(1) choose a random $r \in \mathbb{Z}_n$
(2) Compute $y = r^2 - B^2/4 \pmod{n}$
(3) Use A to obtain a decryption x.
(4) Let $x_1 = x + B2^{-1}$
(5) If $x_1 \equiv \pm r \pmod{n}$ then failure else $\gcd(x_1 + r, n) = p$ or$q$.

This factors n with probability less than or equal to $1/2$

Step 2, the y we are after is $E_k(r - 2^{-1}B)$.

$$E_k(r - 2^{-1}B) = \ldots = r^2 - 4^{-1}B^2$$

Step 4, $x_1^2 \equiv (x + 2^{-1}B)^2 \equiv x^2 + xB + 4^{-1}B^2 \equiv y + 4^{-1}B^2 \equiv r^2$. Therefore $x_1 = \pm r$ or $\pm \omega r$ modulo n, where ω is a square root of 1 modulo n. So, $x_1^2 - r^2 \equiv o \pmod{n}$. Hence,

$$n | (x_1 + r)(x_1 - r).$$

If $x_1 = pm\omega r \pmod{n}$ where $\omega \neq \pm 1$ then n does not divide either factor on the RSH. Therefore, computing the $\gcd(x_1 + r, n)$ or $\gcd(x_1 - r, n)$ must give p or q.

3.3.8 El Gammal Cryptosystem

Discrete Logs (indices)

Suppose p is prime. Then, $\mathbb{Z}_p$ is a field and $\mathbb{Z}_p^*$ is cyclic. If a generates $\mathbb{Z}_p^*$, then a is a *primitive* element of $\mathbb{Z}_p^*$ (i.e., primitive root of p). That is, the numbers $a, a^2, \ldots, a^{p-1}$ modulo p are

$1, 2, \ldots p-1$ in some order. For each element $x \in \mathbb{Z}_p^*$ there is a number $e \in \{1, 2, \ldots, p-1\}$ such that $a^e \equiv x \pmod{p}$. We call e the discrete logarithm (or index) of x with respect to a and denote it by $\log_a(x)$.

For example, Suppose $p = 7$. Is 2 a primitive root? No, since $2^3 \equiv 1$. How about 3? yes. Solve $9x^4 \equiv 5 \pmod{7}$

$$\log_3(9x^4) \equiv \log_3(5) \equiv 5 \pmod{\phi(7)}.$$

So, $\log_3(9) + 4log_3(x) \equiv 5 \pmod{6}$. Hence, $4\log_3(x) \equiv 3 \pmod{6}$. Thus, there is no solution.

El Gammal Cryptosystem

The El Gammal cryptosystem based on the idea that finding discrete logs is a hard problem. It is a public key cryptosystem and non-deterministic (i.e., it uses a random number). Choose a prime p such that finding discrete logs in $\mathbb{Z}_p$ is assumed to be hard. Then, choose a generator in $\mathbb{Z}_p^*$, say α. The message space is $M = \mathbb{Z}_p^*$ and the ciphertext space is $C = \mathbb{Z}_p \times \mathbb{Z}_p$. The keys are tuples (p, α, e, β), where $\beta = \alpha^e \pmod{p}$ with $e = \log_\alpha(\beta)$. p, α, β are public, while e is secret.

The encryption function is $E_k(x) = (y_1, y_2)$ where $y_1 \equiv \alpha^k \pmod{p}$ and $y_2 \equiv x\beta^k \pmod{p}$, with $k \in \mathbb{Z}_{p-1}$ chosen at random. The decryption function is $D_k((y_1, y_2)) = y_2(y_1^e)^{-1} \pmod{p}$, where $y_1^e \equiv (\alpha^k)^e \equiv (\alpha^e)^k \equiv \beta^k \pmod{p}$. Therefore, $(y_1^e)^{-1} = (\beta^k)^{-1}$. Hence, $y_2(y_1^e)^{-1} \equiv y_2(\beta^k)^{-1} = x\beta^k(\beta^k)^{-1} \equiv x \pmod{p}$.

For example, suppose $p = 23$, $\alpha = 5$, and $e = 9$. Then, $\beta = \alpha^e = 5^9 = 80 = 11 \pmod{23}$. Suppose we want to encrypt $x = 7$. Then, first choose a random $k \in \mathbb{Z}_{22}$; say $k = 13$. Compute $y_1 = 5^{13} = 90 = 21 \pmod{23}$. So, $y_2 = 7 \cdot 11^{13} = 8 \cdot 13^3 = 4 \pmod{23}$. Thus, we send the pair $(21, 4)$. When $(21, 4)$ is received. We compute $4 \cdot (21^9)^{-1} = 16 \cdot 14 = 17 \pmod{23}$, and get $17^{-1} = 19 \pmod{23}$, which is found using the Euclidean algorithm. Finally, $4 \cdot 19 = 76 = 7 \pmod{23}$.

3.3.9 Shanks' Algorithm for Computing Discrete Logarithms

Let p be a prime, and suppose α is a generator of $\mathbb{Z}_p^*$. Given β we want to find $e = \log_\alpha(\beta)$. Let $m = \lceil(\sqrt{p-1})\rceil$. Then, we can write $e = mj + i$, where $0 \leq i \leq m-1$. (Turns out $0 \leq j \leq m-1$). Now, $\beta = \alpha^e = \alpha^{mj+i} \pmod{p}$ if and only if $\beta\alpha^{-i} = \alpha^{mj} \pmod{p}$.

Algorithm

1. Compute $\alpha^{mj} \pmod{p}$, where $0 \leq j \leq m-1$, save the pairs $(j, \alpha^{mj} \pmod{p})$, and sort by increasing $2nd$ co-ordinate.

2. Compute $\beta\alpha^{-i} \pmod{p}$, where $0 \leq i \leq m-1$, save the pairs $(i, \beta\alpha^{-i} \pmod{p})$, and sort by increasing 2nd co-ordinate.

3. Find (j, y) in first list such that (i, y) in second list and the second coordinates match.

4. $e = \log_\alpha(\beta) = mj + i \pmod{p-1}$. Note: it is $\pmod{p-1}$ since the first $p-1$ powers repeat in a cycle of length $p-1$ by Fermat's theorem.

For example, suppose $p = 23$ and $\alpha = 5$. We want $\log_5(11)$ (Recall it is 9). $m = \lceil(\sqrt{22})\rceil = 5$. Therefore, $m - 1 = 4$. Compute

$$5^{5\cdot 0},\ \ 5^{5\cdot 1},\ \ 5^{5\cdot 2},\ \ 5^{5\cdot 3},\ \ 5^{5\cdot 4} \pmod{23}.$$

So, we have 1, 20, 9, 19, 12 (mod 23).

$$L_1 = (0,1), (2,9), (4,12), (3,19), (1,20).$$

Compute $11 \cdot 5^{-i} = 11 \cdot 5^{22-i} (\text{mod } 23)$ by Fermat's theorem $11 \cdot 5^2 2$, $11 \cdot 5^2 1$, $11 \cdot 5^2 0$, $11 \cdot 5^1 9$, $11 \cdot 5^1 8$. So, we have 11, $11 \cdot 14 = 16$, $11 \cdot 14^2 = 16 \cdot 14 = 17$, $11 \cdot 14^3 = 17 \cdot 14 = 8$, $11 \cdot 14^4 = 8 \cdot 14 = 20$ (mod 23). So,

$$L_2 = (3,8), (0,11), (1,16), (2,17), (4,20).$$

Hence, $(1, 20)$ matchs $(4, 20)$. Therefore, $j = 1, i = 4$. Therefore, $e = mj + i = 5 \cdot 1 + 1 = 9 \pmod{22}$.

3.3.10 Generalized El Gammal

Elliptic Curves, arises in connection with elliptic integrals, which arise in computing the arc length of an ellipse. Let $p > 3$ be a prime, the elliptic curve $y^2 \equiv x^3 + ax + b \pmod{p}$ over $\mathbb{Z}_p$ is the set of all pairs $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ that make the congruence true, where $a, b \in \mathbb{Z}_p$ are such that $4a^3 + 27b^2 = 0 \pmod{p}$ together with a special point θ called the point at infinity.

Let $(G, \cdot)$ be a finite group and $\alpha \in G$. Let $H =< \alpha >$ such that computing logs with respect to α is "hard". The message space is $M = G$ and the cipher text space is $C = G \times G$. The keys are (G, α, e, β), where $\beta = \alpha^e$ with α, β public and e is secret.

For $x \in G$, $E_{key}(x) = (y_1, y_2)$, such that $y_1 = \alpha^k$, $y_2 = x \cdot \beta^k$ where k is random $0 \le k \le |G| - 1$. $D_{key}((y_1, y_2)) = y_2 \cdot (y_1^e)^{-1}$.

Let $p > 3$ be prime. The elliptic curve: $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{Z}_p$ and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$, is the set of solutions $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ to $y^2 = x^3 + ax + b \pmod{p}$, together with a special point θ called the point at infinity.

We can make an elliptic curve E into an abelian group using the following operation: (arithmetic in $\mathbb{Z}_p$) let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are points on E. If $x_1 = x_2$ and $y_1 = -y_2$ then $P + Q = \theta$ otherwise $P + Q = (x_3, y_3)$ where $x_3 = \lambda^2 - x_1 - x_2$; $y_3 = \lambda(x_1 - x_3) - y_1$ and $\lambda = (y_2 - y_1)(x_2 - x_1)^{-1}$ if $P \neq Q$ $\lambda = (3x_1^2 + a)(2y_1)^{-1}$ if $P = Q$ Finally, $P + \theta = \theta P = P$ $\theta + \theta = \theta$.

The inverse of (x, y) is $(x, -y)$.

For example, let $p = 11$, $E : y^2 = x^3 + x + 6$. Find points on E. For $x \in \mathbb{Z}_{11}$, compute $x^3 + x + 6 \pmod{11}$, and try to solve for y (as $11 \equiv 3 \pmod 4$, the square roots of z are $\pm z^{(11+1)/4} \equiv \pm z^3 \pmod{11}$.

Thus, there are 13 points on E.

suppose $\alpha = (2, 7)$ Need "powers" (multiples) of α compute $2\alpha = (2, 7) + (2, 7)$ $\lambda = (3x_1^2 + a)(2y_1)^{-1} = (3 \cdot 2^3 + 1)(2 \cdot 7)^{-1} = 2 \cdot 3^{-1} = 2 \cdot 4 = 8(mod 11)$ therefore, $x_3 \equiv 8^2 - 2 - 2 \equiv 5(mod 11)$ and $y_3 \equiv 8(2 - 5) \equiv 2(mod 11)$ therefore $2\alpha = (5, 2)$

powers of α are $(2, 7), (5, 2), (8, 3), (10, 2), (3, 6), (7, 9), (7, 2), (3, 5), (10, 9), (8, 8), (5, 9), (2, 4)$

Theorem 3.3.9. Hasse's theorem: If $p > 3$ is prime and E is an elliptic curve over $\mathbb{Z}_p$, then the number $N(E)$ of points on E satisfies $p + 1 - 2\sqrt{p} \leq N(E) \leq p + 1 + 2\sqrt{p}$.

(alg for N(E)-schoof)

theorem If $p > 3$ is prime and E is a elliptic curve over $\mathbb{Z}_p$, then there exists $n_1, n_2 \in \mathbb{Z}$ such that $(E, +)$ isomorphic $\mathbb{Z}_{n_1} \times \mathbb{Z}_{n_2}$, further $n_2 | n_1$ and $n_2 | p - 1$.

Set up ElGammal with $\alpha = (2, 7)$ and $e = 7$. Therefore $\beta = 7\alpha = (7, 2)$ (from the table above)

$E((v, w)) = (k(2, 7), (v, w) + k(2, 7))$ where k is random $0 \leq k \leq 12$. $D((y_1, y_1'), (y_2, y_2')) = (y_2, y_2') - 7(y_1, y_1')$ Encrypt $(10, 9) \in E$-choose random k, $0 \leq k \leq 12$, $k = 3$ compute $k\alpha = k(2, 7) = 2(2, 7) = (8, 3)$ $(10, 9) + 3 \cdot 7(2, 7) = (10, 9) + (3, 5) = 9\alpha + 8\alpha = 17\alpha = 4\alpha = (10, 2)$ send $((8, 3), (10, 2)$

really all you are sending is a number between 1-12, but you are sending four times as much info

decrypt $((8, 3), (10, 2)$ compute $(10, 2) - 7(8, 3) = 4\alpha - 7 \cdot 3\alpha = -17\alpha = 9\alpha = (10, 9)$

3.3.11 Menzies-Vanstone Cryptosystem

E is an elliptic curve over $\mathbb{Z}_p$, $p > 3$ is prime such that $(E, +)$ has a cyclic subgroup $H =< \alpha >$ in which computing logarithms is "hard". The message space $M = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ and the cipher space is $C = E \times \mathbb{Z}_p^* \times \mathbb{Z}_p^*$. The keys are $k = (E, \alpha, e, \beta)$, where $\beta = e\alpha$, $e = log_\alpha(\beta)$, α, β are public and E,e are secret.

To encrypt: $E_k(x_1, x_2) = (y_0, y_1, y_2)$ where $y_0 = l\alpha$, ($l \in \mathbb{Z}_p^*$ is random), $y_1 = c_1 x_1 \pmod p$, $y_2 = c_2 x_2 \pmod p$, $((c_1, c_2) = l\beta \in E)$.

To decrypt: $D_k(y_0, y_1, y_2) = (x', x'')$ where $x' = y_1 c_1^{-1} \pmod p$, $x'' = y_2 c_2^{-1} \pmod p$, and $ey_0 = (c_1 c_2)$.

For example, using the same elliptic curve as before. $\alpha = (2, 7)$ $p = 11$

l 1 2 3 4 5 6 7 8 9 10 11 12
$l\alpha$ $(2,7),(5,2),(8,3),(10,2),(3,6),(7,9),(7,2),(3,5),(10,9),(8,8),(5,9),(2,4)$

Say $e = 7$ therefore $\beta = e\alpha = 7\alpha = (7,2)$ Encrypt (9,1) note not on E but still in message space choose $l \in \mathbb{Z}_p^*$. say $l = 6$ $y_0 = 6\alpha = (7,9)$; $l\beta = 6(7,2) = 6 \cdot 7\alpha = 42\alpha = 3\alpha = (8,3)$. Therefore, $(c_1, c_2) = (8,3)$ $y_1 = 8 \cdot 9 = 6 mod 11$, $y_2 = 3 \cdot 1 \equiv 3 mod 11$. So, send $((7,9),6,3)$. Decrypt $((7,9),6,3)$. $ey_0 = 7(7,9) = 7 \cdot 6\alpha = 3\alpha = (8,3)$, $x' = 6 \cdot 8^{-1} = 6 \cdot 7 \equiv 9 (mod 11)$, $x'' = 3 \cdot 3^{-1} = 3 \cdot 4 \equiv 1 mod 11$.

want to show decryption is the inverse of encryption. Suppose $(x_1, x_2) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ and $E_k(x_1, x_2) = (y_0, y_1, y_2)$. $D_k(y_0, y_1, y_2)$: compute $ey_0 = e(l\alpha) = l(e\alpha) = l\beta = (c_1, c_2)$ $x' = y_1 c_1^{-1} = c_1 x_1 c_1^{-1} = x_1 (mod p)$, $x'' = y_2 c_2^{-1} = c_2 x_2 c_2^{-1} = x_2 mod p$.

3.4 Cryptosystem Design Ideas

The first step is to take a hard problem, find a special case of the problem that is easy to solve. Then, find a reversible transformation of the special case that makes it look like the general case. Then, use the transformed special case for encryption and use the reverse transformation to return to the special case for decryption.

For example, a hard problem is the Subset Sum problem. Given positive integers $s_1, s_2, \ldots, s_n$ and T (i.e., subset sizes and target). Question: Is there a $0-1$ vector $x = (x_1, x_2, \ldots, x_n)$ such that $x_1 s_1 + x_2 s_2 + \ldots + x_n s_n = T$? Easy/special case: list of sizes $s_1, s_2, \ldots, s_n$ such that $s_j > s_1 + s_2 + \ldots + s_{j-1}, 2 \leq j \leq n$ (called super-increasing). Then, it is easy to solve Subset Sum and there is a unique solution.

Algorithm

For $i = n, n-1, \ldots, 1$
If $T \geq s_i$ then $x_i = 1$ and $T = T - s_i$; else $x_i = 0$
If $x_1 s_1 + x_2 s_2 + \ldots + x_n s_n = T$ then yes; else no

Merkle-Hellman

Encryption rule: $E_k(x_1, x_2, \ldots, x_n) = \sum x_i s_i$. Suppose $S = (s_1, s_2, \ldots, s_n)$ is a super-increasing sequence $p > \sum_{i=1}^n s_i$ prime, $a \in \mathbb{Z}_p^*$. Then, $t = (t_1, t_2, \ldots, t_n)$ is defined by $t_i = a \cdot s_i (\text{mod } p)$. The message space is $M = \{0,1\}^n$ and the cipher space is $C = \{0, 1, \ldots, n(p-1)\}$. The key is $k = (s, p, a, t)$ where t is public and s, p, a are private. $E_k(x_1, x_2, \ldots, x_n) = \sum x_i t_i$. $D_k(y) =$ solu $(x_1, x_2, \ldots, x_n)$ to Subset Sum with sizes S and target $T = a^{-1} y \ (\text{mod } p)$.

For example, suppose $S = (2, 5, 10, 25)$, $p = 53$, and $a = 10$. The Public list $t = (t_1, t_2, t_3, t_4)$ is $t_1 = 10 \cdot 2 = 20$, $t_2 = 10 \cdot 5 = 50$, $t_3 = 10 \cdot 10 = 100 = 47$, $t_4 = 10 \cdot 25 = 250 = 38$ all modulo 53. So,

$$t = (20, 50, 47, 38)$$

Suppose we want to Encrypt (1,0,1,1). Compute $20 + 47 + 38 = 105$ and send 105. To decrypt 105. Compute $T = 10^{-1} \cdot 105 = 16 \cdot 105 = 16(-1) = 37 \pmod{53}$. Then,

$$37 > 23 \quad x_4 = 1$$

$$12 > 10 \quad x_3 = 1$$

$$2 < 5 \quad x_2 = 0$$

$$2 = 2 \quad x_1 = 1$$

Hence, we get (1,1,0,1).

Why does decryption reverse encryption: Suppose $y = \sum x_i t_i$. We need to show $\sum x_i s_i = T = a^{-1}y \pmod{p}$. Now, $y = x_1t_1 + x_2t_2 + \ldots + x_nt_n = x_1as_1 + x_2as_2 + \ldots + x_nas_n \pmod{p}$. Therefore, $T = a^{-1}y = a^{-1}x_1as_1 + a^{-1}x_2as_2 + \ldots + a^{-1}x_nas_n = x_1s_1 + x + 2s_2 + \ldots + x_ns_n \pmod{p}$. Since $(s_1, s_2, \ldots s_n)$ is super increasing, the solution is unique.

3.5 Assignment 3

1. The following text was intercepted from an Affine cipher:

   KQEREJEBCPPCJCRKIEACUZBKRVPKRBCIBQCARBJCVFCUP
   KRIOFKPACUZQEPBKRXPEIIEABDKPBCPFCDCCAFIEABDKP
   BCPFEQPKAZBKRHAIBKAPCCIBURCCDKDCCJCIDFUIXPAFF
   ERBICZDFKABICBBENEFCUPJCVKABPCYDCCDPKBCOCPERK
   IVKSCPICBRKIJPKABI
   Determine the plaintext. Give a clear description of the steps you followed.

   Solution: Since we know it is from an Affine cipher, we can break the cipher by frequency analysis of the plain text. Using the following java code with $s1$ as the text above we get the following frequency table:

```
public static void FrequencyOfLetters (String s1){

    int i, numberOfOccurrencesOfLetter;
    char letter;

    for(letter ='A'; letter<='Z'; letter++){

        numberOfOccurrencesOfLetter = 0;

        for(i=0; i< s1.length(); i++){

            if(letter == s1.charAt(i)){
                numberOfOccurrencesOfLetter++;
            }
        }

    System.out.println(letter + ":" +   numberOfOccurrencesOfLetter);

    }


}
```

A:13	N:1
B:21	O:2
C:32	P:20
D:9	Q:4
E:13	R:12
F:10	S:1
G:0	T:0
H:1	U:6
I:16	V:4
J:6	W:0
K:20	X:2
L:0	Y:1
M:0	Z:4

Figure 3.1: Frequency of letters in ciphertext

So, a reasonable guess of correspondences would be $E \to C$ and $T \to B$ since C occurs the most and B the second most. Solving the corresponding concurrencies (i.e., $4a+b \equiv 2 \pmod{26}$ and $19a+b \equiv 1 \pmod{26}$) we see that $a = 19$ (Note: $\gcd(a, 26) = 1$) and $b = 4$. This implies $a^{-1} = 11$. Running the following java code to decrypt using $a^{-1} = 11$ and $b = 4$ we get:

```
public static void DecodeGivenAInverseandB (String s1, int a, int b){

    int i, j, k, l;
    char letter, replacementLetter;
    String s2;
    s2="";

    for(l=0; l< s1.length(); l++){

        i=0;

        for(letter ='A'; letter<='Z'; letter++){
            if(s1.charAt(l)==letter){
                break;
            }
            i++;
        }

        replacementLetter = 'A';
        j = (a*(i-b))%26;

        if(j<0){
            j=j+26;
        }

        for(k=0; k<j; k++){
        replacementLetter++;
        }

        s2=s2 + replacementLetter;

    }

    System.out.println(s2);

}
```

O CANADA TERRE DE NOS AIEUX TON FRONT EST CEINT DE FLEURIONS GLORIEUX CAR TON BRAS SAIT PORTER LEPEE IL SAIT PORITER LA CROIX TON HISTOIRE EST UNE EPOPEE DES PLUS BRILLIANTS EXPLOITS ET TA VALEUR DE FOI TREMPEE PROTEGERA NOIS FOYERS ET NOS DROITS

Thus, we have discovered the plaintext since this is legible and the probability that a different plaintext was encrypted is negligible.

2. Suppose Bob has an RSA cryptosystem with a large modulus n which can not be factored easily. Alice sends a message to Bob by representing the alphabetic characters A, B, ..., Z as 0, 1, ..., 25, respectively, and then encrypting each character (i.e., number) separately.

   (a) Explain how a message encrypted in this way can easily be decrypted.

   (b) The following ciphertext was encrypted using the scheme described above with $n = 18721$ and $b = 25$:

   $$365; 0; 4845; 14930; 2608; 2608; 0$$

   Illustrate your method from (a) by decrypting this ciphertext without factoring n.

   This example illustrates a protocol failure in RSA. It demonstrates that a ciphertext can sometimes be decrypted by an adversary if the system is used in a careless way. A secure cryptosystem is not enough to assure secure communication, it must also be used properly.

Solution: (a) This encryption technique can be easily decrypted since RSA encryptions allow attackers to encrypt chosen plaintext (i.e., The encryption function $E_k(x)$ is known). Thus, an attacker can just individually encrypt A, B, ..., Z represented as 0, 1, ..., 25, and see what ciphertexts corresponds to the 26 plaintext letters then just read off the message. Essentially, Alice just used the encryption function to come up with a correspondence between A, B, ..., Z and numbers in $\mathbb{Z}_n$ then used them for a very simple substitution cipher.

(b) Encrypting A, B, ..., Z represented as 0, 1, ..., 25 individually we get the following values for A, B, ..., Z

$A \to 0$	$B \to 1$	$C \to 6400$	$D \to 18718$	$E \to 17173$	$F \to 1759$	$G \to 18242$
$H \to 12359$	$I \to 14930$	$J \to 9$	$K \to 6279$	$L \to 2608$	$M \to 4644$	$N \to 4845$
$O \to 1375$	$P \to 13444$	$Q \to 16$	$R \to 13663$	$S \to 1437$	$T \to 2940$	$U \to 10334$
$V \to 365$	$W \to 10789$	$X \to 8945$	$Y \to 11373$	$Z \to 5116$		

Thus, replacing $365; 0; 4845; 14930; 2608; 2608; 0$ by the corresponding letters we get:

$$VANILLA.$$

3. What happens if the RSA system is set up using p and q where p is prime but q is not? Does encryption work (is $E_k(x)1 - 1$)? Can all encrypted messages be uniquely decrypted? Illustrate your points with an example where p and q are two digit numbers.

   Solution: If the RSA system is set up using p and q where p is prime but q is not, then factoring $n = pq$ is potentially a lot easier so the security is effected negatively. Furthermore, $E_k(x)$ is not one-to-one and some encrypted messages cannot be uniquely decrypted.

   For example, suppose $p = 11$ and $q = 3 \cdot 2^2 = 12$. Then $n = 132$, $\phi(n) = 10 \cdot 4 = 40$, and a valid choice of a and b are 3 and 27 since $3 \cdot 27 = 81 \equiv 1 \pmod{40}$. If we encrypt the messages 94 and 28, then $94^3 \equiv 40 \pmod{132}$ and $28^3 \equiv 40 \pmod{132}$. Thus, $E_k(x)$ is not one-to-one. Furthermore, if 104 and 130 are encrypted messages then decrypting we get $104^{27} \equiv 40 \pmod{132}$ and $130^{27} \equiv 40 \pmod{132}$. Thus, some encrypted messages cannot be uniquely decrypted.

4. Find a (hopefully small) composite odd integer n, and an integer a, $1 \leq a \leq n - 1$ for which the Miller-Rabin algorithm answers "*prime*". Demonstrate this by stepping through the algorithm, assuming a is generated in step 2.

   Solution: Suppose $n = 9 = 3^3$. Clearly, n is an odd composite number. Starting with step (1) of the Miller-Rabin Algorithm and writing $n = 9$ in the form $n - 1 = 2^k m$, where m is odd, we have $9 - 1 = 8 = 2^3 \cdot 1$, which implies $m = 1$. Thus, moving on to step (2): we choose a random number a such that $1 \leq a \leq n - 1 = 8$. If $a = 8$ is chosen, then computing step (3) we get $8^1 \equiv 8 \not\equiv 1 \pmod{9}$, so we move on to step (4). In step (4) we get $8^1 \equiv 8 \equiv -1 \pmod{9}$, thus we answer "prime and stop". Hence, the Miller-Rabin algorithm answers "*prime*" for the composite odd integer 9 with $a = 8$. This is a bit trivial since $n - 1$ cause the algorithm to answer prime for any n so consider $n = 221 = 13 \times 17$ and $a = 174$. The algorithm answer prime on step (4) $i = 1$. (Numbers found on wiki)

5. Gary's Poor Security (GPS) public key cryptosystem has $E_k(x) = ax \pmod{17575}$, where a the receiver's public key, and $\gcd(a; n) = 1$. The plaintext space and ciphertext space are both $\mathbb{Z}_n$, and each element of $\mathbb{Z}_n$ represents three alphabetic characters as in the following examples:

$$DOG \to 3 \times 26^2 + 14 \times 26 + 6 = 2398$$

$$CAT \to 2 \times 26^2 + 0 \times 26 + 19 = 1731.$$

The following message has been encrypted using Gary's public key, 1411.

$$7017; 17342; 5595; 16298; 12285$$

Explain how to break the system and decrypt the message. Do it. Show your work.

Solution: This is pretty much an Affine cipher with $b = 0$ and an alphabet of size 26^3. So, the system can easily be broken by computing the inverse of the public key a, then multiplying each encrypted number by a^{-1} and representing the result in base 26. It is easy to change a number to a number in base 26 and it is computationally feasible to compute a^{-1} since in order to construct the system is would be computed with the same information we have.

Thus, computing a^{-1} for $a = 1411$ we get $a^{-1} = 16591$. Multiplying $7017; 17342; 5595; 16298; 12285$ each by $a^{-1} = 16591$ and reducing modulo 17575 we get:

$$2247; 797; 13070; 8743; 3160.$$

In base 26 these numbers are:

$$(3, 8, 11)_{26}; (1, 4, 17)_{26}; (19, 8, 18)_{26}; (12, 24, 7)_{26}; (4, 17, 14)_{26}.$$

Decoding using the correspondence A, B, ..., Z as 0, 1, ..., 25 we get:

$$DILBERT\ IS\ MY\ HERO.$$

3.6 Assignment 4

1. Factor 262063 and 9420457 using the $p-1$ method. In each case, how big does B have to be to be successful?

   Solution: Using the implementation below of the $p-1$ method in Java we get as output:

   "262063 factors into: 521 and 503 where $B = 13$ was used."

   "9420457 factors into: 2351 and 4007 where $B = 47$ was used.",

   which by my implementations design implies that $B = 13$ and $B = 47$ where the smallest possible values for B.

```
public class PollardPMinusOne {
    public static void main (String [] args){
    FactoringUsingPMinusOne(262063);
    FactoringUsingPMinusOne(9420457);
    }
    public static void FactoringUsingPMinusOne (int n){
        boolean nNotFactored = true;
        int i=1;
        double j=1;
        while(nNotFactored) {
            j=gcd(TwoToPowerBFactorialMod(i, n)-1, n);
            if(j!=1){
                System.out.println(n + " factors into: " + (int)j + " and " + (int)(n/j) + " where B=" + i + " was used.");
                nNotFactored = false;
            }
            i++;
        }
    }
    public static double TwoToPowerBFactorialMod (int b, int n){
        double TwoToPowBFact = 2;
        for(int i=1;i<=b;i++){
            TwoToPowBFact = bigPowerMod(TwoToPowBFact, i, n)%n;
        }
        return TwoToPowBFact;
    }
    public static double bigPowerMod (double x, int y, int n) {
        double powerOfXToY = x;
        for(int i=1; i<y;i++){
            powerOfXToY = (powerOfXToY*x)%n;
        }
        return powerOfXToY;
    }
    public static double gcd(double n, double a) {
        if (a == 0){
            return n;
        } else{
        return gcd(a, n % a);
        }
    }
```

2. Suppose $p = 199$, $q = 211$, and $B = 1357$ in a Rabin cryptosystem.

   (a) Determine the four square roots of 1 modulo n, where $n = pq$.

   Solution: We know the ± 1 are both square roots of 1 modulo p and modulo q. Thus, solving for x using the Chinese remainder theorem under these constraints $\pm 1 \equiv x \pmod{p}$ and $\pm 1 \equiv x \pmod{p}$ we get the four square roots of 1 modulo $n = pq = 41989$ are:

   $$1 \quad 6964 \quad 35025 \quad 41988$$

(b) Compute $E_k(32767)$.
Solution: $E_k(32767) \equiv 32767(32767 + 1357) \equiv 16027 \pmod{41989}$. So,

$$E_k(32767) = 16027.$$

(c) Determine the four possible decryptions of this ciphertext y.
Solution:

$$\begin{aligned} D_k(16027) &= \sqrt{4^{-1}(1357)^2 + 16027} - 2^{-1}(1357) \\ &\equiv \sqrt{31492(1357)^2 + 16027} - 20995(1357) \pmod{41989} \\ &\equiv \sqrt{4013} - 21673 \pmod{41989} \end{aligned}$$

Using the fact that p and q are both congruent to 3 modulo 4 and the chinese remainder theorem we get that the four roots of 4013 are: 1479 12451 29538 40510. Therefore, the four possible decryptions of the ciphertext $y = 16027$ are:

$$21795 \quad 32767 \quad 7865 \quad 18837.$$

3. The following message was encrypted using the Rabin cryptosystem with $n = 19177 = 127 \times 151$ and $B = 5679$:

$$2251,\ 8836,\ 7291,\ 6035$$

The elements of $\mathbb{Z}_n$ correspond to triples of alphabetic characters interpreted as numbers in base 26. Decrypt the message. Explain how you decided the four possible plaintexts for each ciphertext symbol.

Solution: Using the same decryption process and root finding technique as in question 2 we get:

$$2251 \to \{8129,\ 9941,\ 3557,\ 5369\}$$

$$8836 \to \{7888,\ 11952,\ 1546,\ 5610\}$$

$$7291 \to \{10359,\ 13407,\ 091,\ 3139\}$$

$$6035 \to \{8022, 10308, 3190, 5476\}$$

Thus, the possible decryptions are:

$$2251 \to \{MAR,\ OSJ,\ FGV,\ HYN\}$$

$$8836 \to \{LRK,\ RRS,\ CHM,\ IHU\}$$

$$7291 \to \{PIL,\ TVR,\ ADN,\ EQT\}$$

$$6035 \to \{LWO, PGM, ESS, ICQ\}.$$

Since english, or some language with syntax, was likely sent, we can eliminate possible triples by looking at the likelyhood those letter appear together as a triple in a word or at the beginning of one word and the end of another. Thus, we see that $MARCHMADNESS$ was likely sent, which we can break up into $MARCH\ MADNESS$.

4. (a) Show that 3 is a primitive root of 17, but 2 is not.

Solution: First, computing the first 16 powers of 3 modulo 17 we get:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$3^n \pmod{17}$	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1

By inspection we can see that all 16 power of 3 modulo 17 are unique. So, 3 generates $\mathbb{Z}_{17}^*$. Thus, 3 is a primitive root of 17.

Now, computing the first 16 powers of 3 modulo 17 we get:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$3^n \pmod{17}$	2	4	8	16	15	13	9	1	2	4	8	16	15	13	9	1

Thus, 2 is not a primitive root of 17 since there is no power of 2 that is congruent to 3 modulo 17.

(b) Make a table of logs (indices) for the nonzero elements $\mathbb{Z}_{17}$ with respect to 3.

Solution:

$x \in \mathbb{Z}_{17}^*$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$log_3(x)$	16	14	1	12	5	15	11	10	2	3	7	13	4	9	6	8

(c) Illustrate Shanks' algorithm by using it to compute $\log_3(12)$ in $\mathbb{Z}_{17}$.

Solution: Here $\beta = 12$, $\alpha = 3$, and $m = \lceil\sqrt{16}\rceil = 4$. Thus, computing the pairs $(j, \alpha^{mj} \pmod{\text{p}})$ for $0 \leq j \leq m-1$ and sorting them we get:

$$(0,1), \quad (3,4), \quad (1,13), \quad (2,16).$$

Similarly, computing the pairs $(i, \beta\alpha^{-i} \pmod{\text{p}})$ for $0 \leq i \leq m-1$ and sorting them we get:

$$(1,4), \quad (2,7), \quad (3,8), \quad (0,12).$$

Now, since $(3,4)$ and $(1,4)$ match in the second coordinate, $e = log_3(12) = 4 \cdot 3 + 1 = 13 \pmod{16}$, which agrees with the table from (b).

5. A cryptosystem has *perfect secrecy* if the conditional probability that the plaintext is x, given that the ciphertext y is observed, is equal to the (a priori) probability that the plaintext is x.

Let n be a positive integer. A *Latin square* of order n is an $n \times n$ array in which each row and each column is a permutation of $\{1, 2, \ldots, n\}$.

Let L be a Latin square of order n. Define a cryptosystem with message space and ciphertext space equal to $\{1, 2, \ldots, n\}$ as follows. For each i, $1 \leq i \leq n$, let $e_i(j) = L(i, j)$. Thus each row of L gives rise to an encryption rule. The key is the integer i. Both L and i are secret.

Prove that this cryptosystem has perfect secrecy.

Solution: First, the probability that the plaintext is x is $P(x) = \frac{1}{n}$. Now, the conditional probability the plaintext is x occurs, given that the ciphertext y is observed

$$P(x \mid y) = \frac{P(x \cap y)}{P(y)} = \frac{\frac{1}{n} \cdot \frac{1}{n}}{\frac{1}{n}} = \frac{1}{n}$$

since the probability that y is observed occurs with equal probability for each element of $\{1, 2, \ldots, n\}$ (i.e., $P(y) = \frac{1}{n}$); and the probability $P(x \cap y)$ is the probability row i is chosen (i.e., $\frac{1}{n}$) times the probability that the plaintext is x (i.e., $\frac{1}{n}$) because the row i is independent form the plaintext being encrypted. Hence, $P(x) = P(x \mid y)$. Therefore, the cryptosystem has perfect secrecy.

Bibliography

[CAMVAN 91] P.J. Cameron. J.H. Van Lint. "Designs, Graphs, Codes, and their Links" Cambridge University Press. Cambridge. 1991.

[Katz 08] J. Katz. Y. Lindell. "Introduction to Modern Cryptography". Chapman and Hall. Taylor and Francis Group. 2008.

[MacG 17] G. MacGillivray."Class Notes". Math 413: Applied Algebra. UVIC. Spring 2017.

[Hail 08] J. P. Hailman. "Coding and Redundancy". Harvard University Press. Massachusetts. 2008.