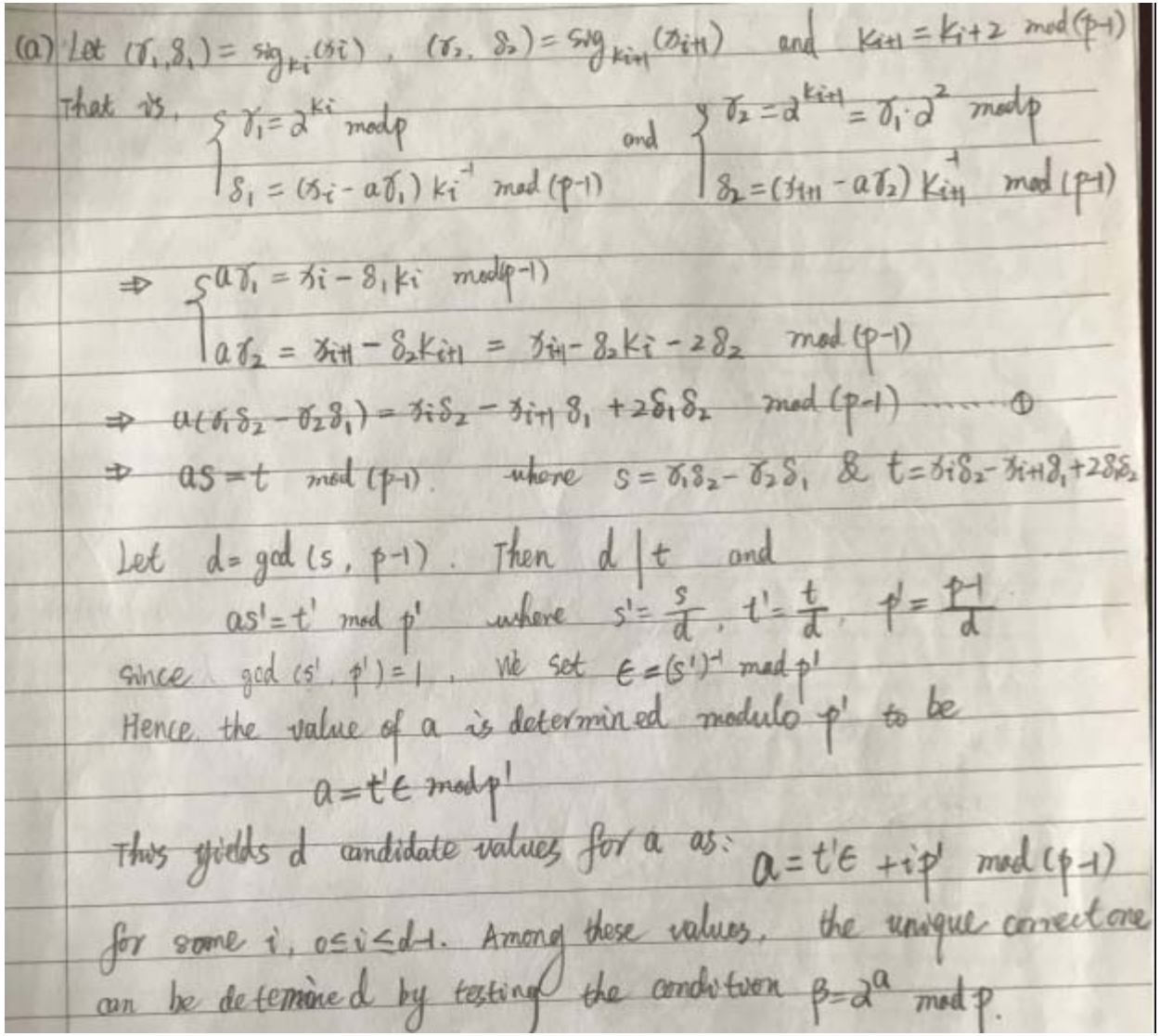
Homework 7

8.3



(a) Let $(\gamma_1, \delta_1) = sig_{k_i}(\gamma_i)$, $(\gamma_2, \delta_2) = sig_{k_{i+1}}(\gamma_{i+1})$ and $k_{i+1} = k_i + 2 \bmod (p-1)$

That is, $\begin{cases} \gamma_1 = \alpha^{k_i} \bmod p \\ \delta_1 = (\gamma_i - a\gamma_1) k_i^{-1} \bmod (p-1) \end{cases}$ and $\begin{cases} \gamma_2 = \alpha^{k_{i+1}} = \gamma_1 \cdot \alpha^2 \bmod p \\ \delta_2 = (\gamma_{i+1} - a\gamma_2) k_{i+1}^{-1} \bmod (p-1) \end{cases}$

$\Rightarrow \begin{cases} a\gamma_1 = \gamma_i - \delta_1 k_i \bmod (p-1) \\ a\gamma_2 = \gamma_{i+1} - \delta_2 k_{i+1} = \gamma_{i+1} - \delta_2 k_i - 2\delta_2 \bmod (p-1) \end{cases}$

$\Rightarrow a(\gamma_i \delta_2 - \gamma_2 \delta_1) = \gamma_i \delta_2 - \gamma_{i+1} \delta_1 + 2\delta_1 \delta_2 \bmod (p-1)$ ..... ①

$\Rightarrow as = t \bmod (p-1)$ where $s = \gamma_1 \delta_2 - \gamma_2 \delta_1$ & $t = \gamma_i \delta_2 - \gamma_{i+1} \delta_1 + 2\delta_1 \delta_2$

Let $d = \gcd(s, p-1)$. Then $d \mid t$ and

$as' = t' \bmod p'$ where $s' = \frac{s}{d}$, $t' = \frac{t}{d}$, $p' = \frac{p-1}{d}$

Since $\gcd(s', p') = 1$, we set $\epsilon = (s')^{-1} \bmod p'$

Hence the value of a is determined modulo p' to be

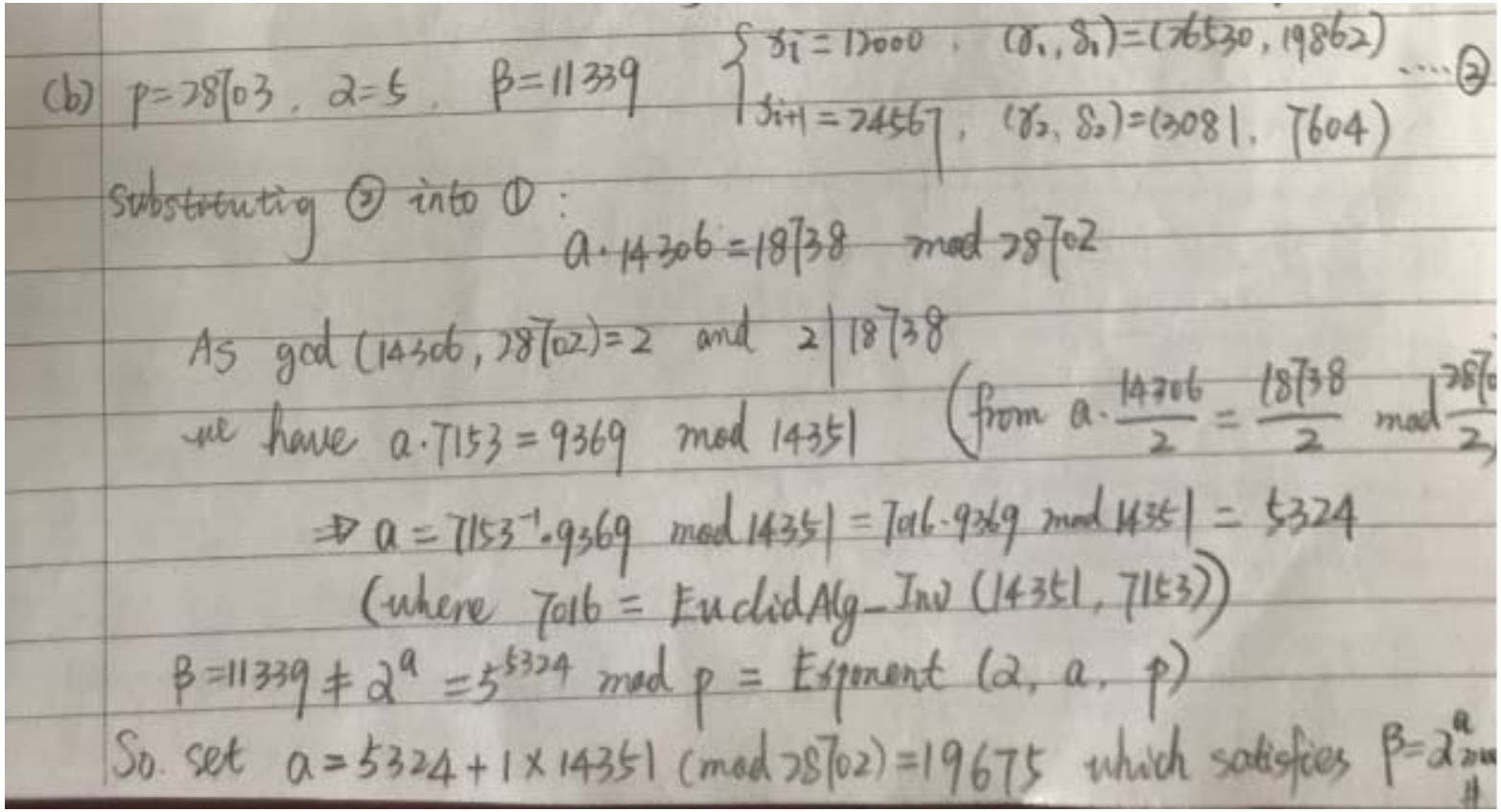
$a = t'\epsilon \bmod p'$

This yields d candidate values for a as: $a = t'\epsilon + ip' \bmod (p-1)$

for some i, $0 \le i \le d-1$. Among these values, the unique correct one can be determined by testing the condition $\beta = \alpha^a \bmod p$.

Homework 7

8.3



(b) $p=28703$, $\alpha=5$, $\beta=11339$ $\begin{cases} s_i=13000, & (\gamma_1,\delta_1)=(26530,19862) \\ s_{i+1}=24567, & (\gamma_2,\delta_2)=(3081,7604) \end{cases}$ ····②

Substituting ② into ①:

$$a\cdot 14306 = 18738 \mod 28702$$

As $\gcd(14306, 28702)=2$ and $2 \mid 18738$

we have $a\cdot 7153 = 9369 \mod 14351$ (from $a\cdot\frac{14306}{2}=\frac{18738}{2} \mod \frac{28702}{2}$)

$\Rightarrow a = 7153^{-1}\cdot 9369 \mod 14351 = 7016\cdot 9369 \mod 14351 = 5324$

(where $7016 = \text{EuclidAlg\_Inv}(14351, 7153)$)

$\beta = 11339 \neq 2^a = 5^{5324} \mod p = \text{Exponent}(2, a, p)$

So set $a = 5324 + 1\times 14351 \pmod{28702} = 19675$ which satisfies $\beta = 2^a \mod p$

Homework 6

7.1

- **Understand the idea of Shank's Algorithm**
- **Implement Shank's Algorithm with Matlab codes named "Shanks.m"**

$$\log_{106} 12375 = 22392 \pmod{24691}$$

$$\log_{6} 248388 = 232836 \pmod{458009}$$

Homework 6

7.1

- U…gorithm
- I…

"…

```
%Algorithm 7.1 of Shank's Algorithm which computes a = log(alpha)(beta)
function [a_exp] = Shanks(n,alpha,beta);

m = ceil(sqrt(n));

%am = alpha^m;
am = Exponent(alpha, m, n);%computing am = alpha^m (mod n) by using Algorithm 6.5

a(1) = 1;
for j = 2:1:m
    a(j) = mod(a(j-1) * am, n);
end

[List1, ind_List1] = sort(a);

temp_alpha(1) = 1;
inv_alpha(1) = 1;
b(1) = beta;
for i = 2:1:m
    temp_alpha(i) = mod(temp_alpha(i-1) * alpha, n);

    inv_alpha(i) = EuclidAlg_Inv(n,temp_alpha(i));%By using Algorithm 6.3

    b(i) = mod(beta * inv_alpha(i), n);
end

[List2, ind_List2] = sort(b);

flag = 0;
for i=1:1:m
    cur_a = List1(i);
    for j=1:1:m
        if cur_a == List2(j)
            flag = 1;
            a_exp = mod(m * (ind_List1(i) - 1) + (ind_List2(j) - 1), n);
            break;
        end
    end
    if flag == 1
        break;
    end
end
```

```
% Algorithm 6.3 which computes the inverse of the element b modulo a
function [inv] = EuclidAlg_Inv(mod_valu, elemt_valu);

% mod_valu = 9987;
% elemt_valu = 3125;

a = mod_valu;
b = elemt_valu;

t0 = 0;
t = 1;

q = floor(a/b);
r = a - q * b;

while r > 0
    temp = mod((t0 - q * t), mod_valu);
    t0 = t;
    t = temp;
    a = b;
    b = r;
    q = floor(a/b);
    r = a - q * b;
end

if b ~= 1
    b
else
    inv = t;
end
```

Homework 6

7.9 Decrypt the ElGamal ciphertext presented in Table 7.4. The parameters of the system are $p = 31847$, $\alpha = 5$, $a = 7899$ and $\beta = 18074$. Each element of $\mathbb{Z}_n$ represents three alphabetic characters as in Exercise ~~6.12~~ 6.13.

Answer: basic idea: For the ElGamal Cryptosystem

ElGamal Encryption & Decryption

$Enc(x) = (y_1, y_2)$: $y_1 = \alpha^k \bmod p$ and $y_2 = x\beta^k \bmod p$

$x = Dec(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p$

Represent x in 26-ary as $x=(bcd)_{26}$ where $x = b*26^2 + c*26 + d$ and b, c, d are in Z_{26}, then the plaintext can be obtained.

System setup: ***$p=31847$, $\alpha=5$, $\beta=18074$, $a=7899$***

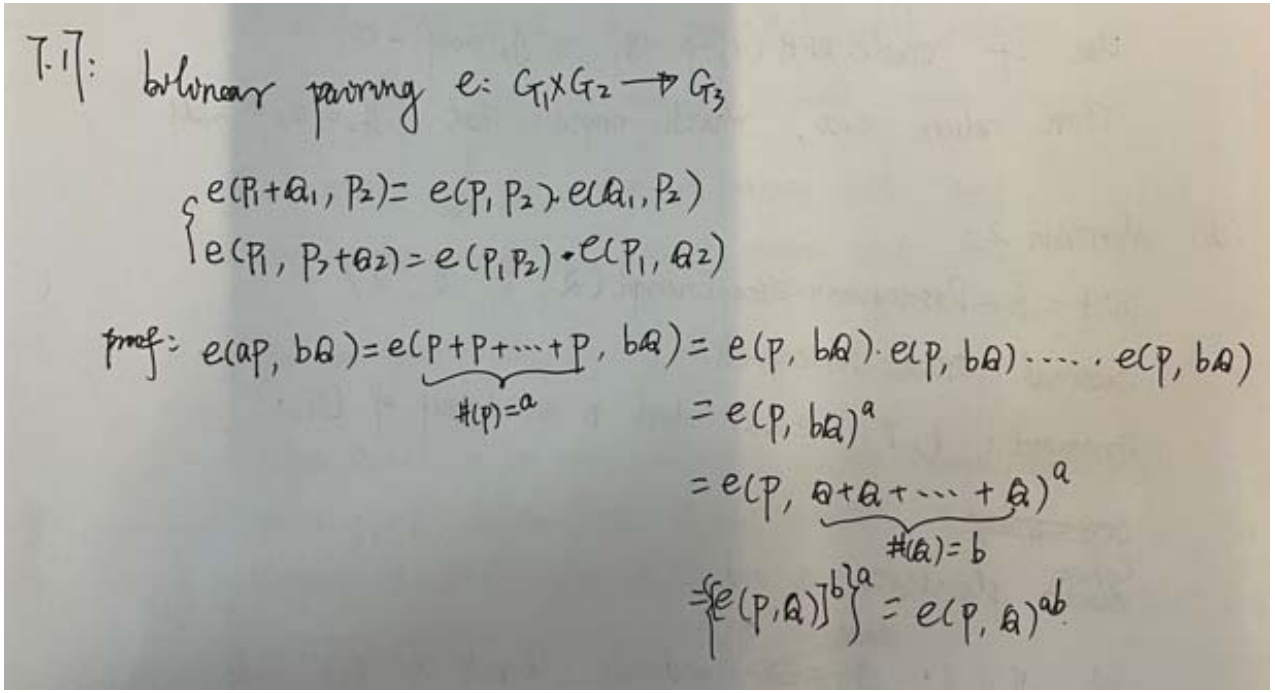
For example: $(y_1,y_2)=(3781, 14409)$

$x = y_2(y_1^a)^{-1} \bmod p = 14409 * (3781^{7899})^{-1} \bmod 31847$

then $x=(x_1 x_2 x_3)$ satisfying $x= x_1 * 26^2 + x_2 *26+x_3$ where x_i is in$\{0,1,\ldots, 25\} \leftrightarrow \{A,B,\ldots, Z\}$

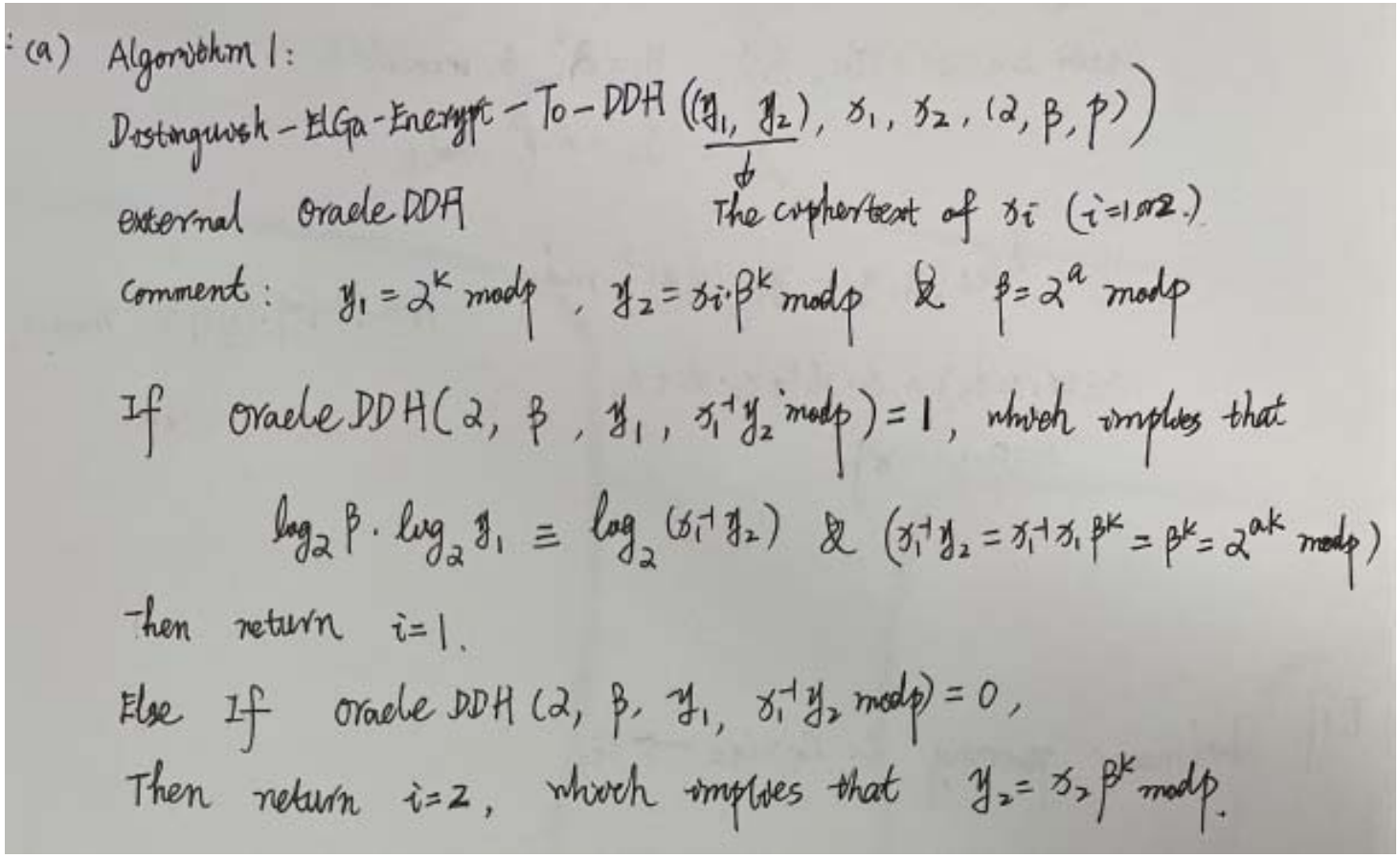
Homework 6

7.17



Homework 6

7.23



(a) Algorithm 1:

Distinguish-ElGa-Encrypt-To-DDH $((y_1, y_2), \gamma_1, \gamma_2, (2, \beta, p))$

external oracle DDH — (y_1, y_2): The ciphertext of γ_i $(i=1 \text{ or } 2)$.

Comment: $y_1 = 2^k \bmod p$, $y_2 = \gamma_i \cdot \beta^k \bmod p$ & $\beta = 2^a \bmod p$

If oracle DDH$(2, \beta, y_1, \gamma_1^{-1} y_2 \bmod p) = 1$, which implies that

$$\log_2 \beta \cdot \log_2 y_1 \equiv \log_2 (\gamma_1^{-1} y_2) \quad \& \quad (\gamma_1^{-1} y_2 = \gamma_1^{-1} \gamma_1 \beta^k = \beta^k = 2^{ak} \bmod p)$$

then return $i=1$.

Else If oracle DDH$(2, \beta, y_1, \gamma_1^{-1} y_2 \bmod p) = 0$,

Then return $i=2$, which implies that $y_2 = \gamma_2 \beta^k \bmod p$.

Homework 6

7.23

(b) Algorithm 2:

DDH − To − Distinguish-ElGa-Encrypt $(\alpha, \beta, \gamma, \delta)$

external Oracle Distinguish

Comment: $\beta, \gamma, \delta \in \langle \alpha \rangle$ which is a subgroup of $(G, \cdot)$

Given plaintext x_1 and x_2.

Set $y_1 = \gamma$, $y_2 = x_i \delta \bmod p$ for randomly choose of $i = 1, 2$

Set $B \triangleq$ Oracle Distinguish $((y_1, y_2), x_1, x_2, (\alpha, \beta, p))$

If $B=1$, which implies that $y_2 = x_1 \cdot \beta^k \bmod p$ & $y_1 = \alpha^k \bmod p$

Then $\delta = x_1^{-1} y_2 = x_1^{-1} x_1 \beta^k = \beta^k = \alpha^{ak}$

$\Leftrightarrow \log_\alpha \delta = \log_\alpha \beta \cdot \log_\alpha y_1 = \log_\alpha \beta \cdot \log_\alpha \gamma$

Then return 1.

If $B=2$, which implies that $\begin{cases} y_1 = \alpha^k \bmod p \\ y_2 = x_2 \cdot \beta^k \bmod p \end{cases}$

Then $\delta = x_2^{-1} y_2 = x_2^{-1} \cdot x_2 \beta^k = \alpha^{ak} \Leftrightarrow \log_\alpha \delta = \log_\alpha \beta \cdot \log_\alpha \gamma$

Then return 1

Homework 5

6.3 Implement the Extended Euclidean Algorithm (Algorithm 6.2) as the following tables

(a) Set

$$r_i = \begin{cases} |0| & , i=0 \\ |7| & , i=1 \\ r_{i-1} q_{i-1} + r_{i-2} & , i \geq 0 \quad (0 \leq r_{i+2} < r_{i+1}) \end{cases}$$

$$s_i = \begin{cases} 1 & , i=0 \\ 0 & , i=1 \\ s_{i-2} - q_{i-2} \cdot s_{i-1} & , i \geq 2 \end{cases} \quad \& \quad t_i = \begin{cases} 0 & , i=0 \\ 1 & , i=1 \\ t_{i-2} - q_{i-2} t_{i-1} & , i \geq 2 \end{cases}$$

If there exists a integer m such that $r_m = 1$, then $t_m = |7|^{-1} \bmod |0|$.

then

i	r_i	q_i	s_i	t_i
0	$\|0\|$		1	0
1	$\|7\|$	5	0	1
2	16	1	$1-5\cdot 0=1$	$0-5\cdot 1=-5$
3	1	16	$0-1\cdot 1=-1$	$1-1\cdot(-5)=6$

So: when $m=3$: $|7|^{-1} \bmod |0| = 6$.

Homework 5

6.3

⑥

i	r_i	q_i	s_i	t_i
0	1234		1	0
1	357	3	0	1
2	163	2	$1-3\cdot0=1$	$0-3\cdot1=-3$
3	31	5	$0-2\cdot1=-2$	$1-2\cdot(-3)=7$
4	8	3	$1-5\cdot(-2)=11$	$-3-5\cdot7=-38$
5	7	1	$-2-3\cdot11=-35$	$7-3\cdot(-38)=121$
6	1	7	$11-1\cdot(-35)=46$	$-38-1\cdot121=-159$

$\therefore 357^{-1} \equiv -159 \equiv 1075 \pmod{1234}$

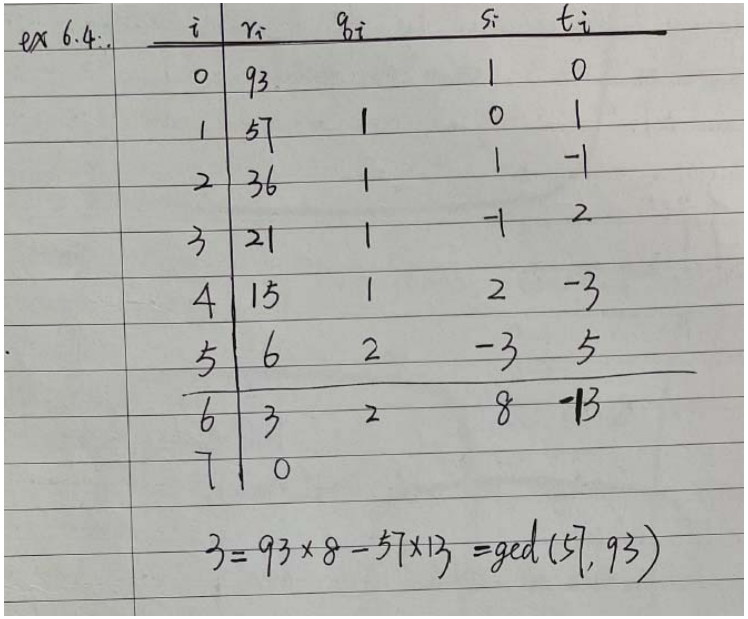
⑦

i	r_i	q_i	s_i	t_i
0	9987		1	0
1	3175	3	0	1
2	612	5	$1-3\cdot0=1$	$0-3\cdot1=-3$
3	65	9	$0-5\cdot1=-5$	$1-5\cdot(-3)=16$
4	27	2	$1-9\cdot(-5)=46$	$-3-9\cdot16=-147$
5	11	2	$-5-2\cdot46=-97$	$16-2\cdot(-147)=310$
6	5	2	$46-2\cdot(-97)=240$	$-147-2\cdot310=-767$
7	1	5	$-97-2\cdot240=-577$	$310-2\cdot(-767)=1844$

$\therefore 3175^{-1} \equiv 1844 \pmod{9987}$

Homework 5

6.4

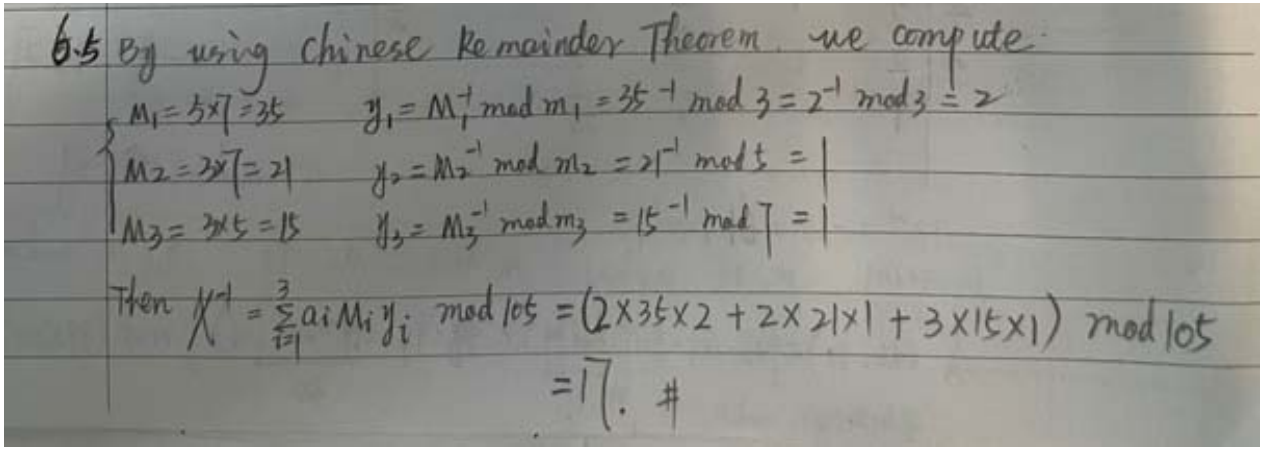


ex 6.4:

i	r_i	q_i	s_i	t_i
0	93		1	0
1	57	1	0	1
2	36	1	1	-1
3	21	1	-1	2
4	15	1	2	-3
5	6	2	-3	5
6	3	2	8	-13
7	0			

$3 = 93 \times 8 - 57 \times 13 = \gcd(57, 93)$

6.5



6.5 By using Chinese Remainder Theorem, we compute:

$M_1 = 5 \times 7 = 35 \quad y_1 = M_1^{-1} \bmod m_1 = 35^{-1} \bmod 3 = 2^{-1} \bmod 3 = 2$

$M_2 = 3 \times 7 = 21 \quad y_2 = M_2^{-1} \bmod m_2 = 21^{-1} \bmod 5 = 1$

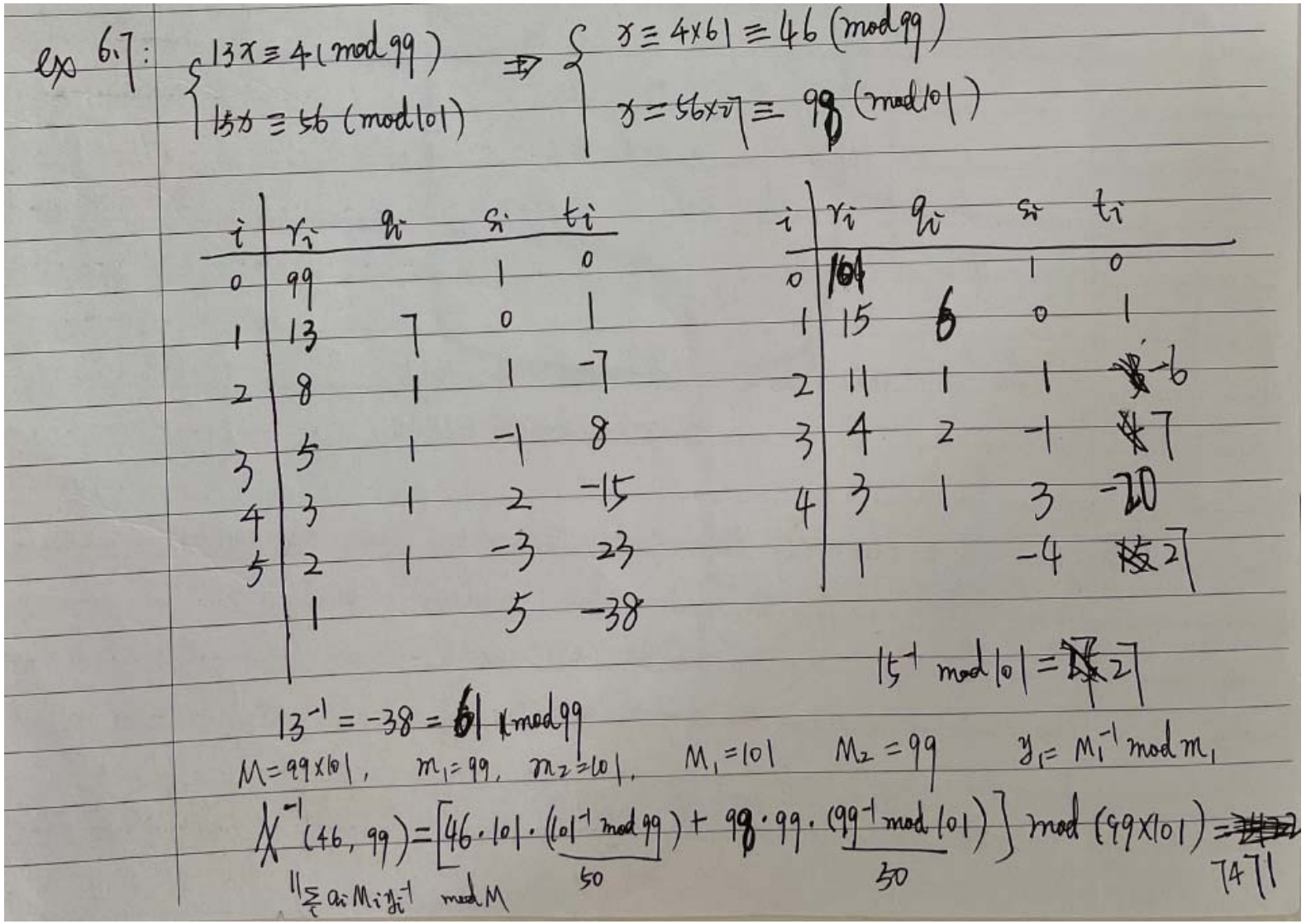
$M_3 = 3 \times 5 = 15 \quad y_3 = M_3^{-1} \bmod m_3 = 15^{-1} \bmod 7 = 1$

Then $X^{-1} = \sum_{i=1}^{3} a_i M_i y_i \bmod 105 = (2 \times 35 \times 2 + 2 \times 21 \times 1 + 3 \times 15 \times 1) \bmod 105$

$= 17.$ #

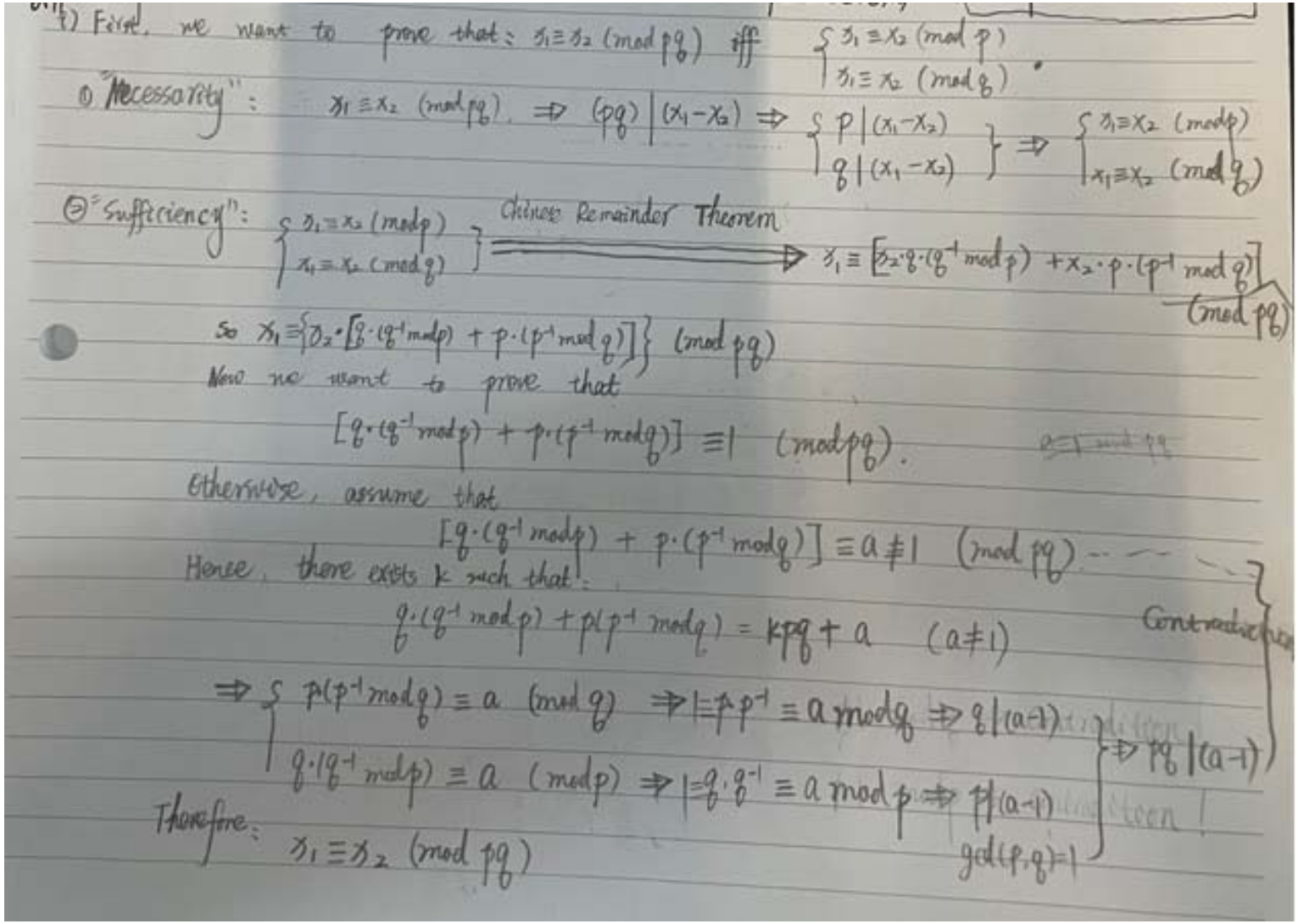
Homework 5

6.7



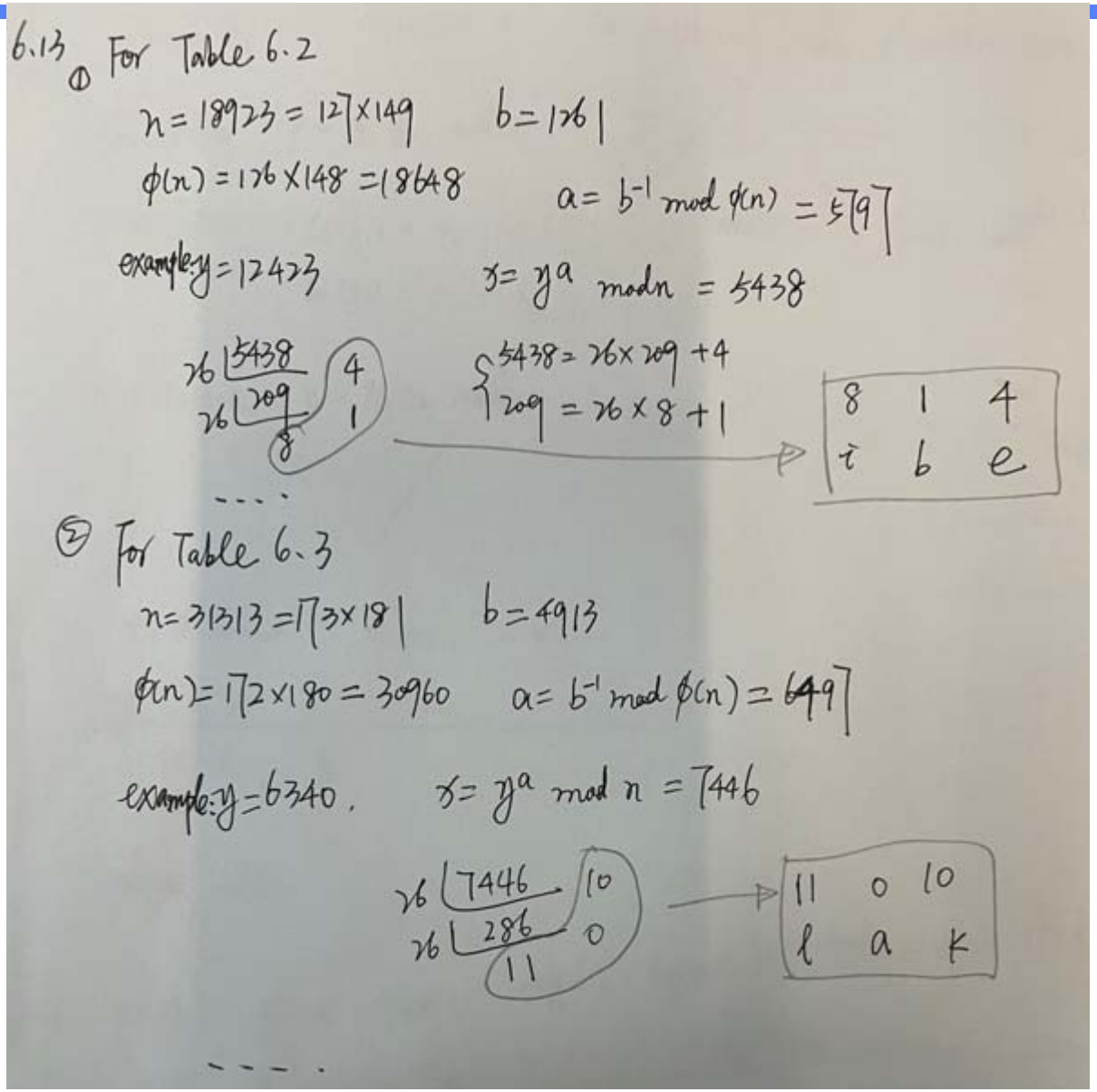
Homework 5

6.11 Proof:



Homework 5

6.13



6.13 ① For Table 6.2

$n = 18923 = 127 \times 149$ $\quad b = 1761$

$\phi(n) = 176 \times 148 = 18648$ $\quad a = b^{-1} \bmod \phi(n) = 5719$

example: $y = 12423$ $\quad x = y^a \bmod n = 5438$

$176 \mid 5438$ (4), $176 \mid 209$ (1), 8

$5438 = 176 \times 209 + 4$

$209 = 176 \times 8 + 1$

8	1	4
i	b	e

...

② For Table 6.3

$n = 31313 = 173 \times 181$ $\quad b = 4913$

$\phi(n) = 172 \times 180 = 30960$ $\quad a = b^{-1} \bmod \phi(n) = 6449$

example: $y = 6340$. $\quad x = y^a \bmod n = 7446$

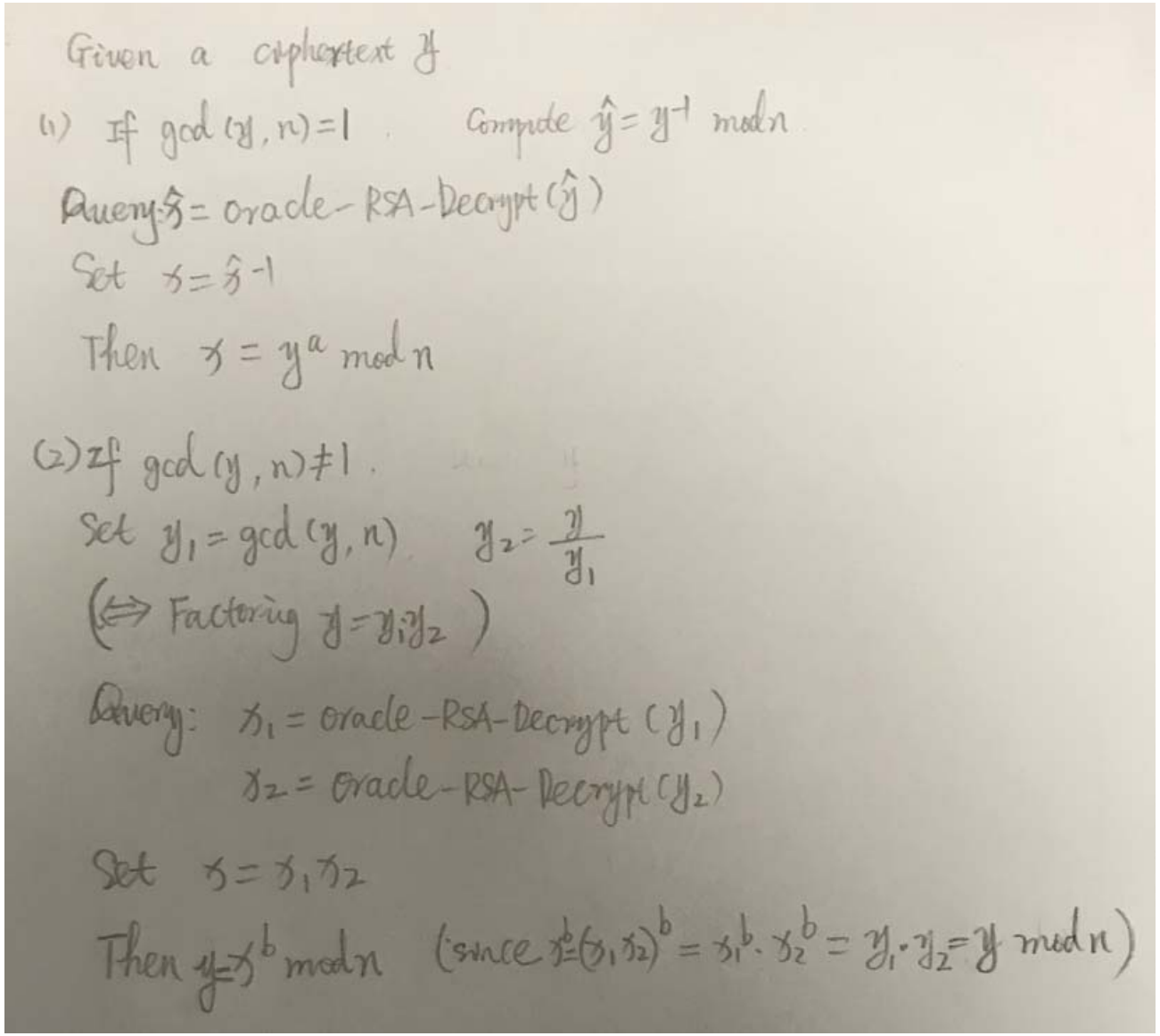
$176 \mid 7446$ (10), $176 \mid 286$ (0), 11

11	0	10
l	a	k

....

Homework 5

6.15



Homework 5

```
%Algorithm 6.5 which computes z = x^c (mod n)
function [z] = Exponent(x, c, n);

bi_c = dec2bin(c);
len_bi_c = length(bi_c);

z = 1;
for i = 1:1:len_bi_c
    z = mod(z * z, n);
    if bi_c(i) == dec2bin(1)
        z = mod(z * x, n);
    end
end
```

6.16 a) By Look-up Table

b) n = 18721, b = 25 = $(11001)_2$; use the square-and-multiply algorithm (Algorithm 6.5) to compute x^b mod n for all x in Z_{26} to generate the table as follows:

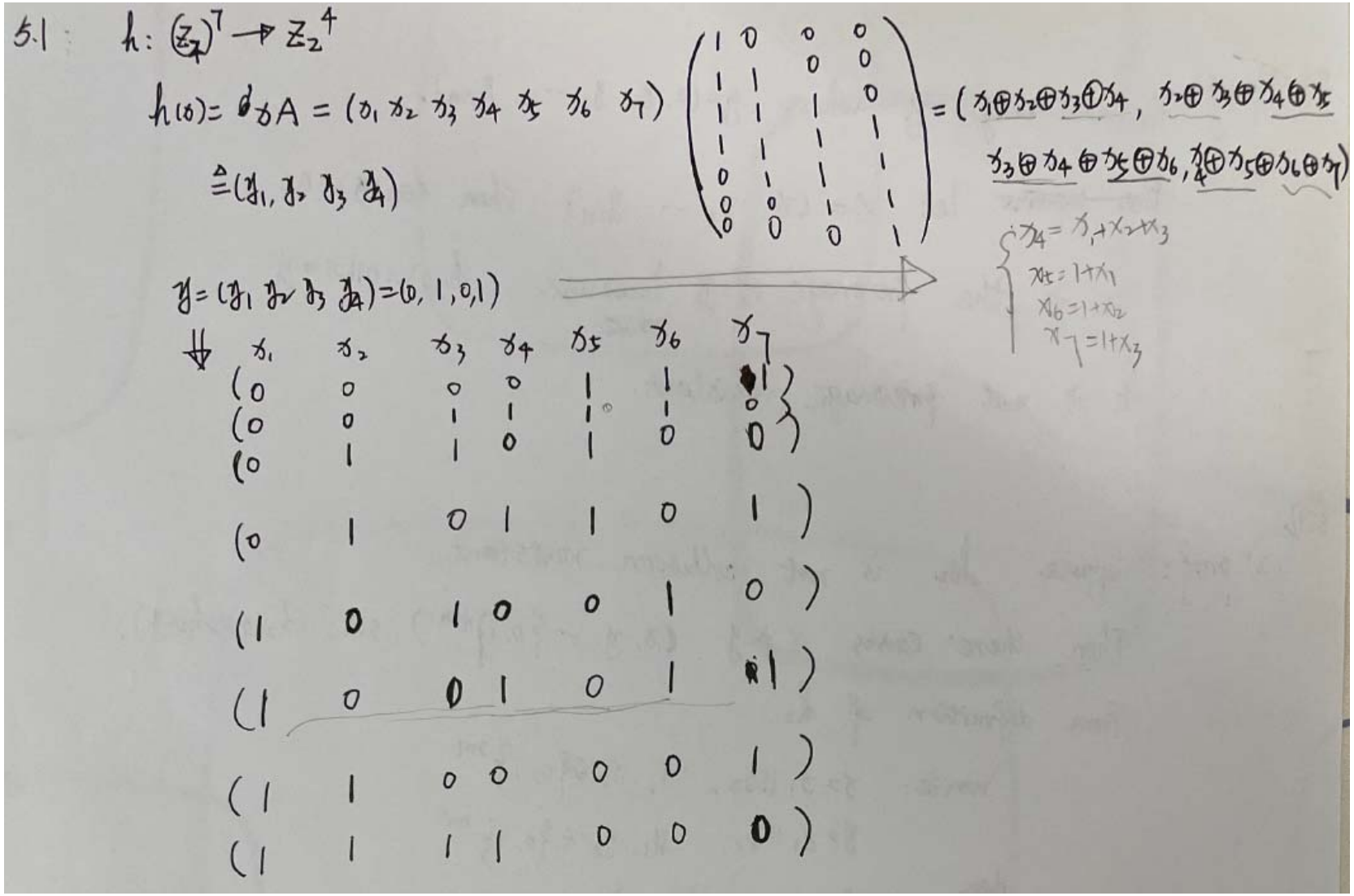
x	0	1	2	3	4	5	6	7	8	9
y	0	1	6400	18718	17173	1759	18242	12359	14930	9

x	10	11	12	13	14	15	16	17	18
y	6279	2608	4644	4845	1375	13444	16	13663	1437

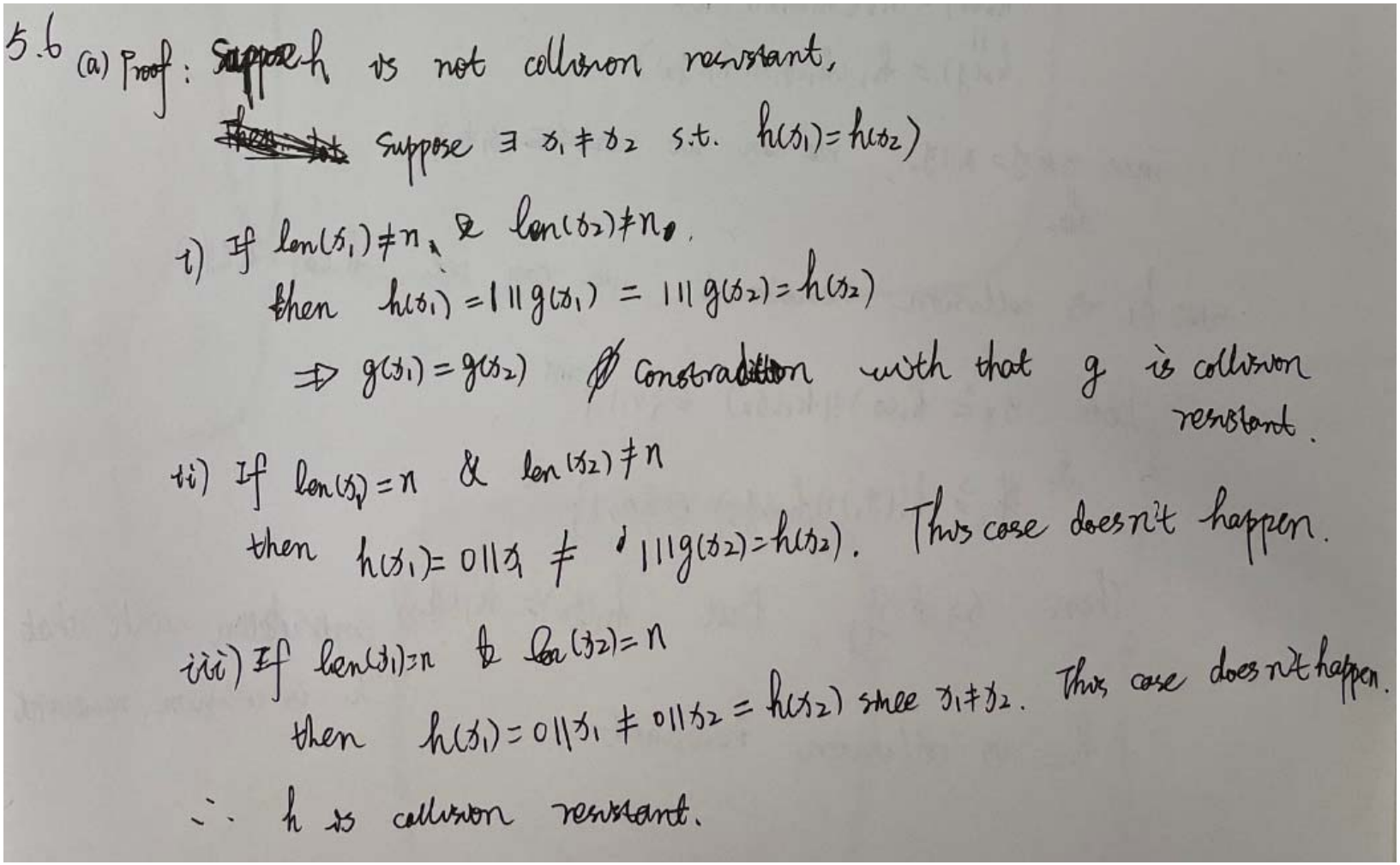
x	19	20	21	22	23	24	25
y	2940	10334	365	10789	8945	11373	5116

365, 0, 4845, 14930, 2608, 2608, 0 ⇒ VANILLA

Homework 4


5.1
$h: (\mathbb{Z}_2)^7 \to \mathbb{Z}_2^4$
$h(0) = 0A = (x_1\ x_2\ x_3\ x_4\ x_5\ x_6\ x_7)$
$\triangleq (y_1, y_2, y_3, y_4)$
$= (x_1 \oplus x_2 \oplus x_3 \oplus x_4,\ x_2 \oplus x_3 \oplus x_4 \oplus x_5,$
$x_3 \oplus x_4 \oplus x_5 \oplus x_6,\ x_4 \oplus x_5 \oplus x_6 \oplus x_7)$
$y = (y_1\ y_2\ y_3\ y_4) = (0, 1, 0, 1)$
$x_4 = y_1 + x_2 + x_3$
$x_5 = 1 + x_1$
$x_6 = 1 + x_2$
$x_7 = 1 + x_3$

Homework 4



5.6 (a) Proof: ~~Suppose~~ h is not collision resistant,

~~Then, not~~ Suppose $\exists\ x_1 \neq x_2$ s.t. $h(x_1) = h(x_2)$

i) If $len(x_1) \neq n$, & $len(x_2) \neq n$.

then $h(x_1) = 1 \| g(x_1) = 1 \| g(x_2) = h(x_2)$

$\Rightarrow g(x_1) = g(x_2)$ ⊘ Contradiction with that g is collision resistant.

ii) If $len(x_1) = n$ & $len(x_2) \neq n$

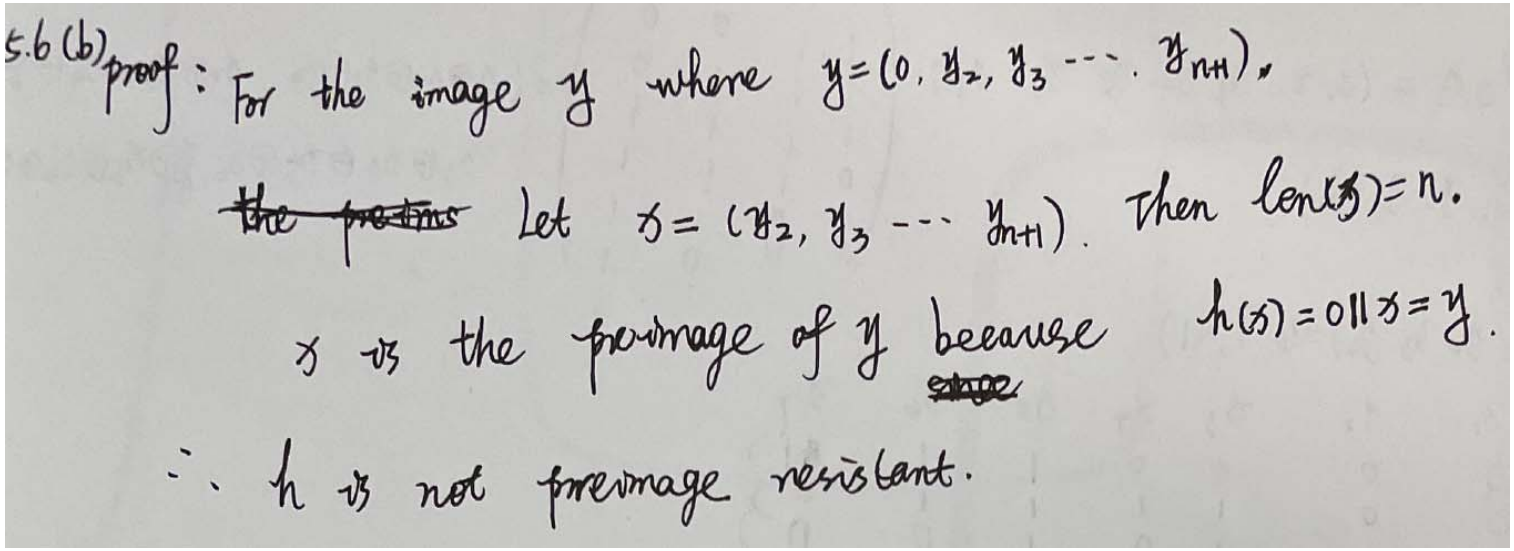
then $h(x_1) = 0 \| x_1 \neq 1 \| g(x_2) = h(x_2)$. This case doesn't happen.

iii) If $len(x_1) = n$ & $len(x_2) = n$

then $h(x_1) = 0 \| x_1 \neq 0 \| x_2 = h(x_2)$ since $x_1 \neq x_2$. This case doesn't happen.

∴ h is collision resistant.

Homework 4

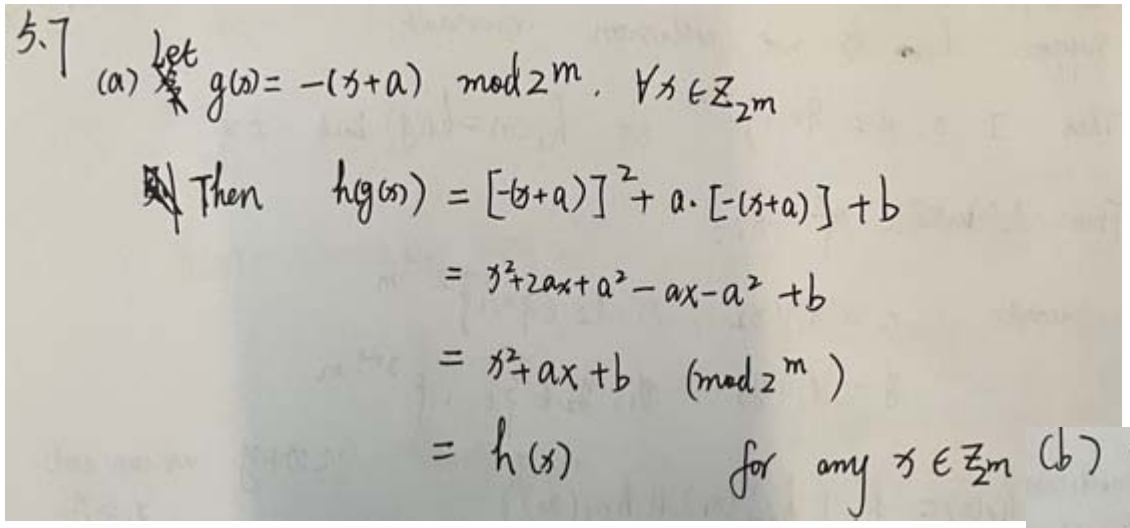

5.6 (b) proof: For the image y where $y=(0, y_2, y_3 \cdots y_{n+1})$,

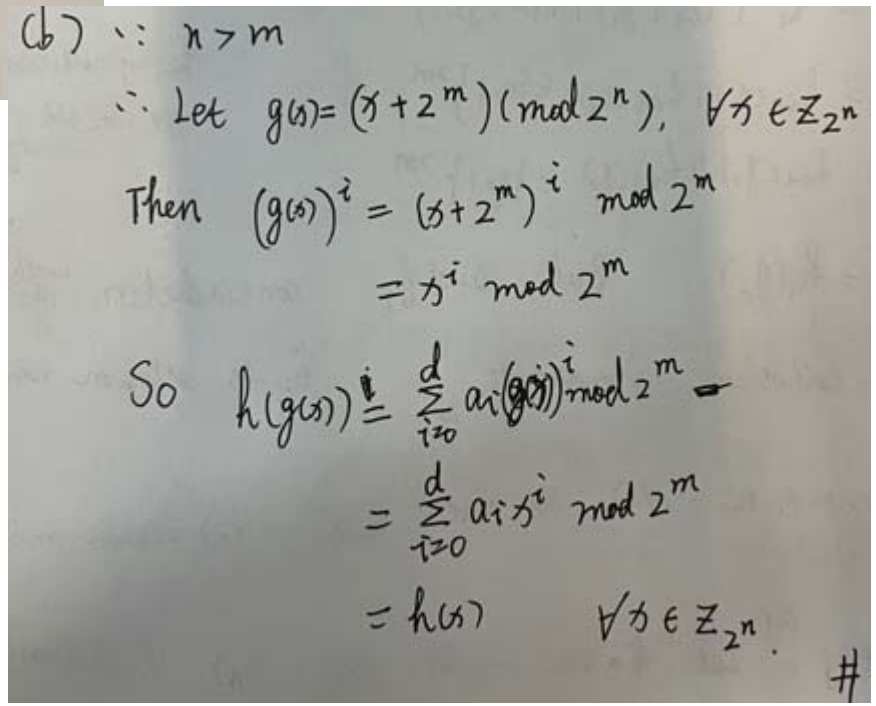
~~the pretims~~ Let $x=(y_2, y_3 \cdots y_{n+1})$. Then $len(x)=n$.

x is the preimage of y because $h(x)=0||x=y$.

$\therefore$ h is not preimage resistant.

Homework 4


5.7
(a) Let $g(x) = -(x+a) \bmod 2^m$, $\forall x \in Z_{2^m}$
Then $h(g(x)) = [-(x+a)]^2 + a\cdot[-(x+a)] + b$
$= x^2 + 2ax + a^2 - ax - a^2 + b$
$= x^2 + ax + b \pmod{2^m}$
$= h(x)$ for any $x \in Z_{2^m}$


(b) $\because n > m$
$\therefore$ Let $g(x) = (x + 2^m) \pmod{2^n}$, $\forall x \in Z_{2^n}$
Then $(g(x))^i = (x+2^m)^i \bmod 2^m$
$= x^i \bmod 2^m$
So $h(g(x)) \equiv \sum_{i=0}^{d} a_i (g(x))^i \bmod 2^m$
$= \sum_{i=0}^{d} a_i x^i \bmod 2^m$
$= h(x)$ $\forall x \in Z_{2^n}$.
#

Homework 4

5.8

Proof: Let $x_1 = x' \| x''$ where $x' \neq x'' \in \{0,1\}^m$. Then $x_1 \in \{0,1\}^{2m}$

Let $\hat{x}' = x' \oplus 0^{m-1}1$ $\hat{x}'' = x'' \oplus 0^{m-1}0$ and $x_2 = \hat{x}' \| \hat{x}'' \in \{0,1\}^{2m}$

~~and $x_2 = \hat{x}' \| \hat{x}''$~~.

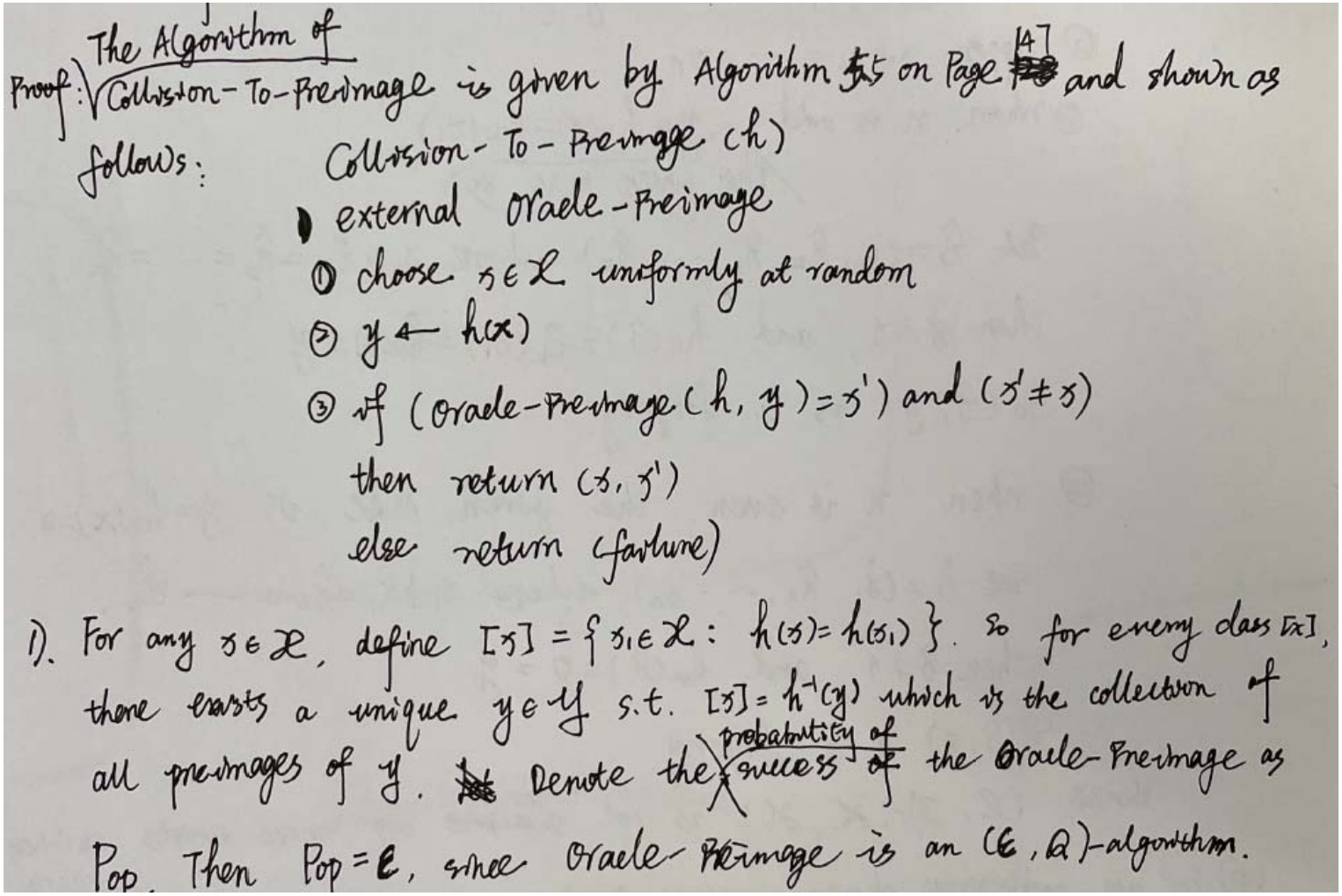
Then $x_2 \neq x_1$, but $x' \oplus x'' = \hat{x}' \oplus \hat{x}''$

Hence $h(x_2) = f(\hat{x}' \oplus \hat{x}'') = f(x' \oplus x'') = h(x_1)$. since f is a preimage resistant bijection.

Therefore h is not second preimage resistant.

Homework 4

5.11



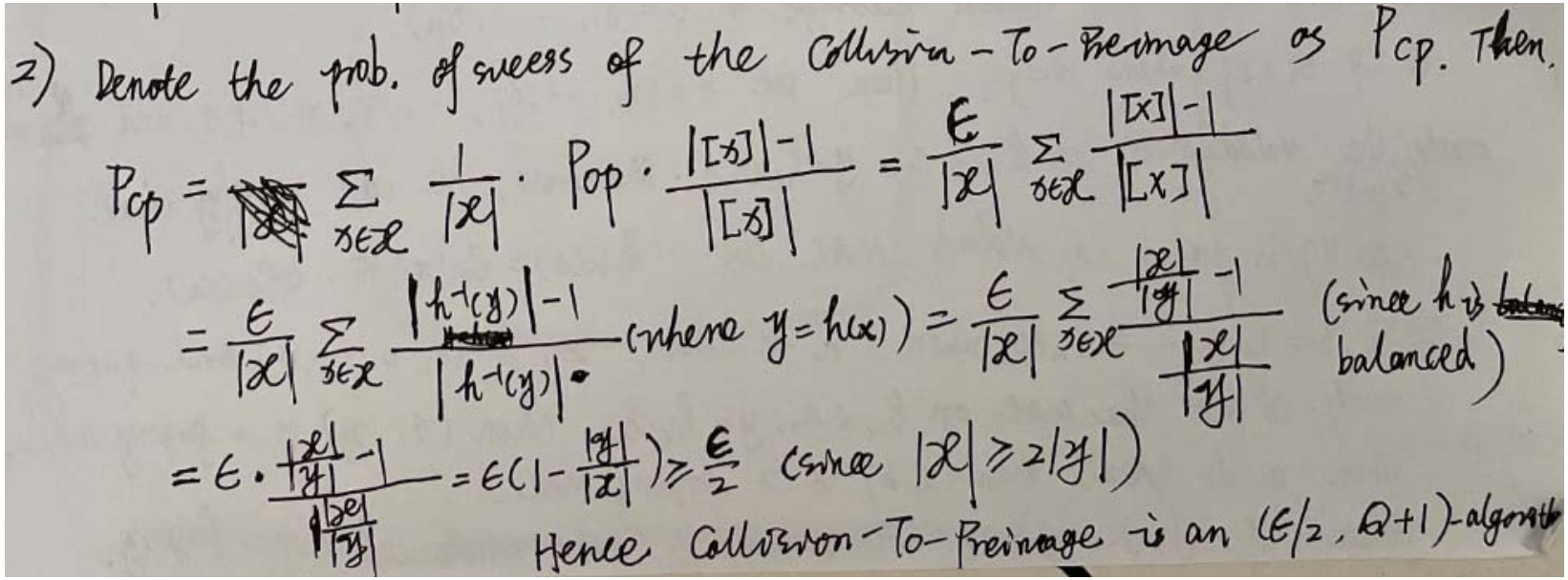
Proof: The Algorithm of Collision-To-Preimage is given by Algorithm 4.5 on Page [4] and shown as follows:

```
Collision-To-Preimage(h)
external Oracle-Preimage
① choose x ∈ X uniformly at random
② y ← h(x)
③ if (Oracle-Preimage(h, y) = x') and (x' ≠ x)
    then return (x, x')
    else return (failure)
```

1). For any $x \in \mathcal{X}$, define $[x] = \{x_1 \in \mathcal{X} : h(x) = h(x_1)\}$. So for every class $[x]$, there exists a unique $y \in Y$ s.t. $[x] = h^{-1}(y)$ which is the collection of all preimages of y. Denote the ~~probability of success of~~ the Oracle-Preimage as Pop. Then Pop = ϵ, since Oracle-Preimage is an (ϵ, Q)-algorithm.

Homework 4

5.11



Homework 4

5.12

(a) proof: Suppose h_2 is not collision resistant.

Then there exists $x \neq y$ ($x, y \in \{0,1\}^{2m}$), s.t. $h_2(x) = h_2(y)$.

From definition of h_2:

write: $x = x_1 \| x_2$, $x_1, x_2 \in \{0,1\}^m$

$y = y_1 \| y_2$, $y_1, y_2 \in \{0,1\}^m$

then $h_2(x) = h_1(h_1(x_1) \| h_1(x_2))$

$h_2(y) = h_1(h_1(y_1) \| h_1(y_2))$

since $x \neq y = x_1 \| x_2$, we can set $x_1 \neq y_1$

since h_1 is collision resistant, we can set $h_1(x_1) \neq h_1(y_1)$

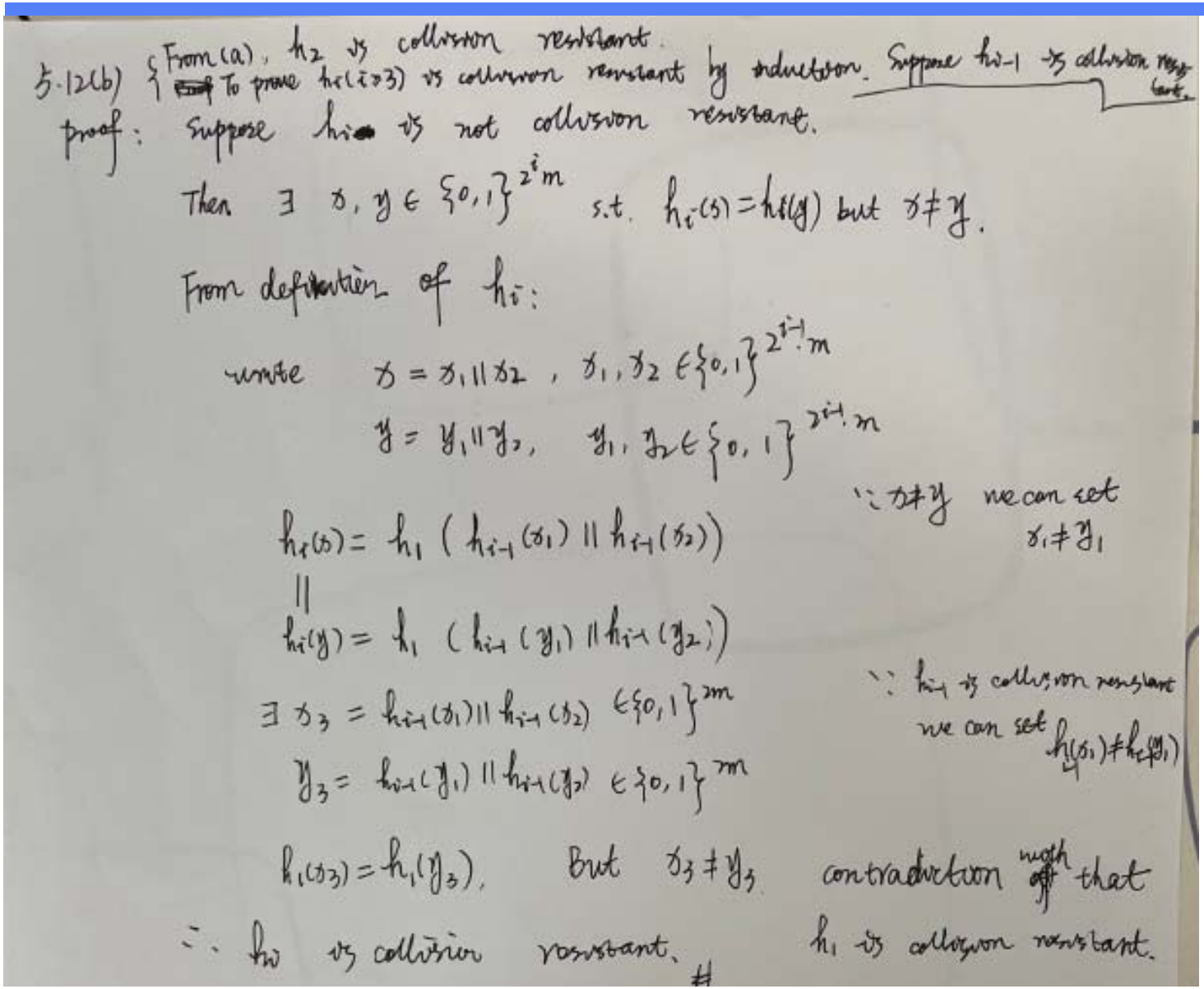
Let $z_3 \triangleq h_1(x_1) \| h_1(x_2) \in \{0,1\}^{2m}$

& $z_3' \triangleq h_1(y_1) \| h_1(y_2) \in \{0,1\}^{2m}$

Then $z_3 \neq z_3'$. But $h_1(z_3) = h_1(z_3')$ contradiction with that h_1 is collision resistant

$\therefore$ h_2 is collision resistant

Homework 4



5.12(b) {From (a), h_2 is collision resistant.
To prove $h_i (i \geq 3)$ is collision resistant by induction. Suppose h_{i-1} is collision resistant.

proof: Suppose h_i is not collision resistant.

Then $\exists\ x, y \in \{0,1\}^{2^i m}$ s.t. $h_i(x) = h_i(y)$ but $x \neq y$.

From definition of h_i:

write $x = x_1 \| x_2$, $x_1, x_2 \in \{0,1\}^{2^{i-1} \cdot m}$

$y = y_1 \| y_2$, $y_1, y_2 \in \{0,1\}^{2^{i-1} \cdot m}$

$\because x \neq y$ we can set $x_1 \neq y_1$

$h_i(x) = h_1(h_{i-1}(x_1) \| h_{i-1}(x_2))$

$\|$

$h_i(y) = h_1(h_{i-1}(y_1) \| h_{i-1}(y_2))$

$\because h_{i-1}$ is collision resistant we can set $h_{i-1}(x_1) \neq h_{i-1}(y_1)$

$\exists\ z_3 = h_{i-1}(x_1) \| h_{i-1}(x_2) \in \{0,1\}^{2m}$

$y_3 = h_{i-1}(y_1) \| h_{i-1}(y_2) \in \{0,1\}^{2m}$

$h_1(z_3) = h_1(y_3)$. But $z_3 \neq y_3$ contradiction with that h_1 is collision resistant.

$\therefore h_i$ is collision resistant. #

Homework 4

5.15 Suppose that $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is a cryptosystem with $\mathcal{P} = \mathcal{C} = \{0,1\}^m$. Let $n \geq 2$ be a fixed integer, and define a hash family $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, where $\mathcal{X} = (\{0,1\}^m)^n$ and $\mathcal{Y} = \{0,1\}^m$, as follows:

$$h_K(x_1, \ldots, x_n) = e_K(x_1) \oplus \cdots \oplus e_K(x_n).$$

Suppose that $(x_1, \ldots, x_n)$ is an arbitrary message. Show how an adversary can then determine $h_K(x_1, \ldots, x_n)$ ~~$= e_K(x_1)$~~ by using at most one oracle query. (This is called a *selective forgery*, because a specific message is given to the adversary and the adversary is then required to find the tag for the given message.)

5.15 Proof: For Given the message $x(x_1, x_2, \cdots, x_n)$, its MAC $h_K(x_1, \cdots x_n)$ ~~$= e_K(x_1)$~~ can be obtained as follows:

① If $\exists\, i, j$ s.t. $x_i \neq x_j$ ($i \neq j$), Let $\hat{x} = (x_1, \cdots x_j, x_i, \cdots x_n)$ & Query its MAC $y = h_K(\hat{x})$ ~~$= e_K(x_1)$~~. Then the MAC of x is $y = h_K(x)$

② If $x_1 = x_2 = \cdots = x_n$, & n is even Then the MAC of x is 0.

③ If $x_1 = x_2 = \cdots = x_n$ & n is odd $n \geq 3$. ~~Then~~ Let $\hat{x} = (x_1, 0, \cdots 0)$ & Query its MAC $y = h_K(\hat{x}) = e_K(x_1)$. Then the MAC of x is $y = \oplus\, h_K(x) = e_K(x_1)$.

Homework 3

4.3 Suppose y = DES(x, K) and y' = DES(c(x), c(K)), where c(·) denotes the bitwise complement of its argument. Prove that y'= c(y).

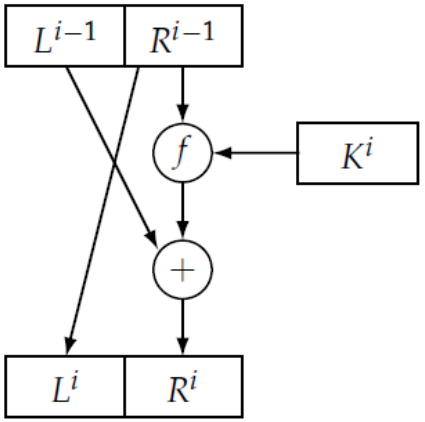
Answer:

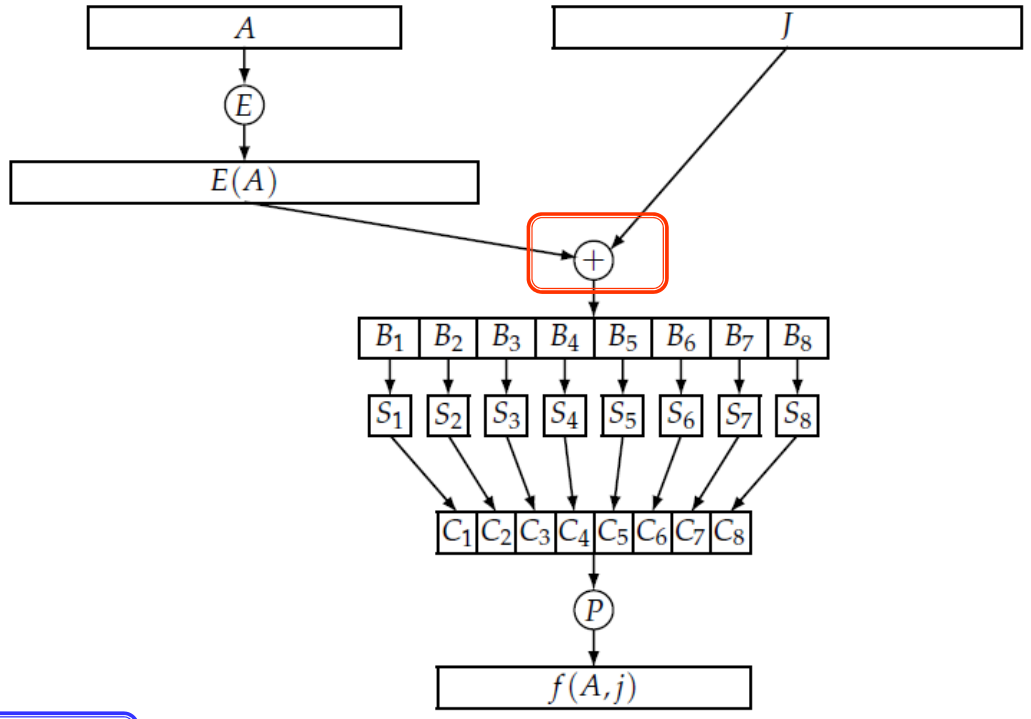
$$\mathbf{IP}(x) = L^0R^0$$

$$g(L^{i-1}, R^{i-1}, K^i) = (L^i, R^i)$$

$$L^i = R^{i-1}$$

$$R^i = L^{i-1} \oplus f(R^{i-1}, K^i)$$





$$y = \mathbf{IP}^{-1}(R^{16}L^{16})$$

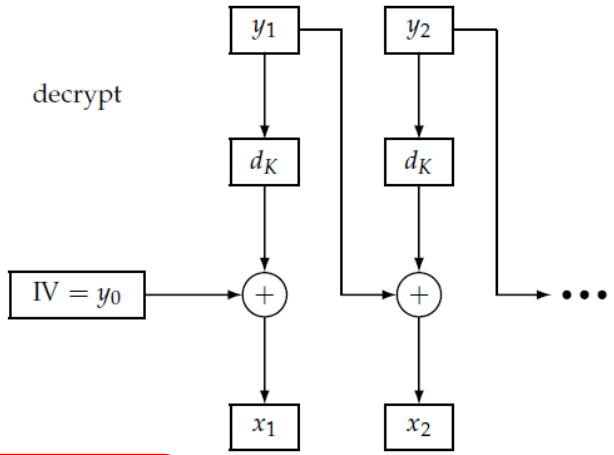
Homework 3

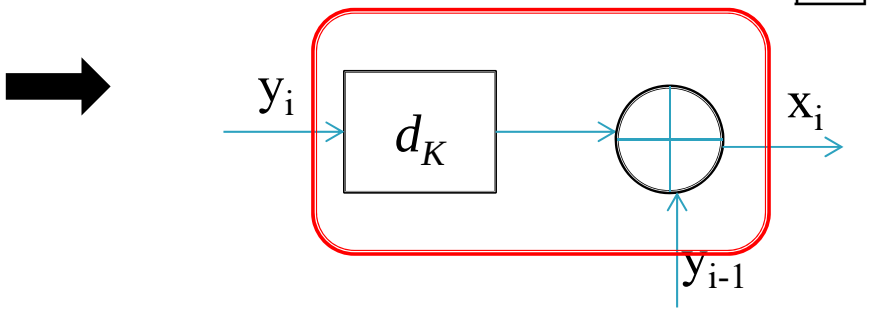
4.6 Prove that decryption in CBC mode can be parallelized efficiently. More precisely, suppose we have n ciphertext blocks and n processors. Show that it is possible to decrypt all n ciphertext blocks in constant time.

Answer: For CBC mode

$$y_0 = \text{IV}$$

$$x_i = y_{i-1} \oplus d_K(y_i)$$





Homework 3

4.8 Suppose that $X = (x_1, \ldots, x_n)$ and $X' = (x'_1, \ldots, x'_n)$ are two sequences of n plaintext blocks. Define

$$\textbf{same}(X, X') = \max\{j : x_i = x'_i \text{ for all } i \leq j\}.$$

Suppose X and X' are encrypted in CBC or CFB mode using the same key and the same IV. Show that it is easy for an adversary to compute $\textbf{same}(X, X')$.

Answers: For CFB mode,

$y_0 = \text{IV}$	$y_0 = \text{IV}$
$z_i = e_K(y_{i-1})$	$z_i = e_K(y_{i-1})$
$y_i = x_i \oplus z_i$	$x_i = y_i \oplus z_i$

If the same key and the same IV are used, then it is easy to compute the following value from the ciphertexts Y and Y' as

same(X, X') = same(Y, Y')

Homework 3

4.9 Suppose that $X = (x_1, \ldots, x_n)$ and $X' = (x'_1, \ldots, x'_n)$ are two sequences of n plaintext blocks. Suppose X and X' are encrypted in OFB mode using the same key and the same IV. Show that it is easy for an adversary to compute $X \oplus X'$.

Answers: For OFB mode,

$$z_0 = \text{IV}$$

$$z_i = e_K(z_{i-1})$$

$$y_i = x_i \oplus z_i$$

If the same key and the same IV are used, then the same sequence of z_i is obtained. Hence it is easy to compute $X \oplus X'$ since

$$y_i \oplus y'_i = x_i \oplus z_i \oplus x'_i \oplus z_i = x_i \oplus x'_i$$